

# DEPLOYED WAZUH EDR AND SYSMON FOR COMPLETE CYBER DEFENSE AT INTERNEE.PK

Project by:

Ahmad Sameer

#1 Implement Endpoint Security & Monitoring

[ahmadsameer0990@gmail.com](mailto:ahmadsameer0990@gmail.com)

LinkedIn: [Ahmad Sameer](#) | GitHub: [Ahmadx90](#)

# Table of Contents

1. Objective.....	4
Lab Setup & IP Details .....	5
2. Step 1: Installing Wazuh Manager on Ubuntu 20.04 (EDR Setup) .....	5
1.1 Update and Install Dependencies.....	5
1.2 Download and Install Wazuh Manager .....	6
1.3 Verify Wazuh Services and Dashboard.....	7
Step 2: Install Wazuh Agent on Windows 10.....	11
2.1 Download Wazuh Agent.....	11
2.2 Configure Agent to Connect to Ubuntu Wazuh Manager .....	11
2.3 Start Agent and Verify Connection .....	13
Step 3: Install and Configure Sysmon on Windows 10.....	14
3.1 Download Sysmon .....	14
3.2 Download Sysmon Configuration File .....	15
3.3 Install Sysmon with Configuration .....	15
3.4 Verify Sysmon Event Logging .....	16
Step 4: Implement File Integrity Monitoring (FIM) in Wazuh .....	17
4.1 Purpose of File Integrity Monitoring .....	17
4.2 Configure FIM on Wazuh Manager .....	17
4.4 Verify FIM Events on Wazuh Dashboard .....	18
Event Table Explanation.....	21
Step 5: Configure Security Alerts and Automated Responses in Wazuh .....	22
5.1 Objective of Alerts and Automation.....	22
5.2 Review Default Wazuh Rules.....	23
5.3 Configure Email or Slack Alerts (Optional).....	24

5.4 Enable Active Response .....	24
5.5 Validate Alerts .....	26
Step 6: Integrate Threat Intelligence with MalwareBazaar in Wazuh.....	26
6.1 Objective of Threat Intelligence Integration .....	26
6.2 Download MalwareBazaar Threat Feed .....	26
6.3 Configure Wazuh to Use Threat Feed .....	26
6.4 Create a Detection Rule for Malware Hashes .....	28
6.5 Test Threat Intelligence Detection.....	28
7. Conclusion .....	30

## 1. Objective

I was assigned to implement an advanced endpoint security and monitoring solution designed to protect critical systems from evolving cyber threats. The objective of this project was to deploy a robust, real-time security framework using Wazuh as an Endpoint Detection and Response (EDR) platform, complemented by Sysmon for detailed event logging. The goal was to strengthen endpoint defenses against malware, unauthorized access, insider threats, and system integrity violations.

To achieve this, I deployed Wazuh Manager on an Ubuntu 20.04 server to provide centralized visibility and threat detection. A Windows 10 endpoint was integrated using the Wazuh Agent, enabling continuous monitoring of file changes, process activities, and network connections. Sysmon was installed and configured to enhance logging capabilities, ensuring granular visibility into endpoint behavior.

Additionally, I configured File Integrity Monitoring (FIM) to detect unauthorized file modifications, created custom alert rules for suspicious activities, and enabled automated responses to react to security events in real time. To further enhance the solution, I integrated MalwareBazaar threat intelligence feeds into Wazuh, enabling proactive detection of known malware indicators.

The overarching objective was to build a comprehensive security ecosystem capable of early threat detection, automated incident response, and improved organizational resilience against cyberattacks.

## Lab Setup & IP Details

### Lab Topology and IP Configuration

In this lab, I implement Endpoint Security & Monitoring using **Wazuh** as the EDR and **Sysmon** for advanced endpoint logging.

The environment consists of the following systems:

Device	OS	IP Address	Role
Ubuntu	Ubuntu 20.04	<a href="http://192.168.137.153">192.168.137.153</a>	Wazuh Manager / EDR
Windows 10	Windows 10 Pro	<a href="http://192.168.137.148">192.168.137.148</a>	Endpoint (Wazuh Agent)

## 2. Step 1: Installing Wazuh Manager on Ubuntu 20.04 (EDR Setup)

### 1.1 Update and Install Dependencies

Running the following commands to update the Ubuntu 20.04 system and install required dependencies:

```
sudo apt update && sudo apt upgrade -y  
sudo apt install curl apt-transport-https unzip -y
```



```
spider@raven:~$ sudo apt install curl apt-transport-https unzip -y
[sudo] password for spider:
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-25ubuntu1.2).
unzip set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  apt-transport-https curl
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 163 kB of archives.
After this operation, 575 kB of additional disk space will be used.
Get:1 http://pk.archive.ubuntu.com/ubuntu focal-updates/universe amd64 apt-transport-https all 2.0.10 [1,708 B]
Get:2 http://security.ubuntu.com/ubuntu focal-security/main amd64 curl amd64 7.68.0-1ubuntu2.25 [162 kB]
Fetched 163 kB in 3s (51.0 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 185550 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.0.10_all.deb ...
Unpacking apt-transport-https (2.0.10) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.68.0-1ubuntu2.25_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.25) ...
Setting up apt-transport-https (2.0.10) ...
Setting up curl (7.68.0-1ubuntu2.25) ...
Processing triggers for man-db (2.9.1-1) ...
```

Figure 1 Install Dependencies

## 1.2 Download and Install Wazuh Manager

I downloaded the official Wazuh installer script and ran it with the `-a` option to deploy the all-in-one stack:

```
curl -s0 https://packages.wazuh.com/4.8/wazuh-install.sh
sudo bash wazuh-install.sh -a
```

This installed:

- *Wazuh Manager*
- *Elasticsearch*
- *Kibana*
- *Filebeat*

```

Setting up curl (7.68.0-1ubuntu2.25) ...
Processing triggers for man-db (2.9.1-1) ...
spider@raven:~$ curl -sO https://packages.wazuh.com/4.8/wazuh-install.sh
spider@raven:~$ sudo bash wazuh-install.sh -a
06/08/2025 07:12:20 INFO: Starting Wazuh installation assistant. Wazuh version: 4.8.2
06/08/2025 07:12:20 INFO: Verbose logging redirected to /var/log/wazuh-install.log
06/08/2025 07:12:23 INFO: Verifying that your system meets the recommended minimum hardware requirements.
06/08/2025 07:12:32 INFO: --- Dependencies ---
06/08/2025 07:12:32 INFO: Installing gawk.
06/08/2025 07:12:39 INFO: Wazuh web interface port will be 443.
06/08/2025 07:12:49 INFO: Wazuh repository added.
06/08/2025 07:12:49 INFO: --- Configuration files ---
06/08/2025 07:12:49 INFO: Generating configuration files.
06/08/2025 07:12:50 INFO: Generating the root certificate.
06/08/2025 07:12:50 INFO: Generating Admin certificates.
06/08/2025 07:12:50 INFO: Generating Wazuh indexer certificates.
06/08/2025 07:12:50 INFO: Generating Filebeat certificates.
06/08/2025 07:12:50 INFO: Generating Wazuh dashboard certificates.
06/08/2025 07:12:50 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
06/08/2025 07:12:51 INFO: --- Wazuh indexer ---
06/08/2025 07:12:51 INFO: Starting Wazuh indexer installation.
06/08/2025 07:17:41 INFO: Wazuh indexer installation finished.
06/08/2025 07:17:42 INFO: Wazuh indexer post-install configuration finished.
06/08/2025 07:17:42 INFO: Starting service wazuh-indexer.
06/08/2025 07:18:09 INFO: wazuh-indexer service started.
06/08/2025 07:18:09 INFO: Initializing Wazuh indexer cluster security settings.
06/08/2025 07:18:19 INFO: Wazuh indexer cluster security configuration initialized.
06/08/2025 07:18:19 INFO: Wazuh indexer cluster initialized.
06/08/2025 07:18:19 INFO: --- Wazuh server ---
06/08/2025 07:18:19 INFO: Starting the Wazuh manager installation.

```

Figure 2 Installation of ELK Stack

```

06/08/2025 07:22:48 INFO: --- Wazuh dashboard ---
06/08/2025 07:22:48 INFO: Starting Wazuh dashboard installation.
06/08/2025 07:24:48 INFO: Wazuh dashboard installation finished.
06/08/2025 07:24:48 INFO: Wazuh dashboard post-install configuration finished.
06/08/2025 07:24:48 INFO: Starting service wazuh-dashboard.
06/08/2025 07:24:49 INFO: wazuh-dashboard service started.
06/08/2025 07:24:52 INFO: Updating the internal users.
06/08/2025 07:25:00 INFO: A backup of the internal users has been saved in the /etc/wazuh-indexer/internalusers-backup folder.
06/08/2025 07:25:56 INFO: Initializing Wazuh dashboard web application.
06/08/2025 07:25:57 INFO: Wazuh dashboard web application initialized.
06/08/2025 07:25:57 INFO: --- Summary ---
06/08/2025 07:25:57 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
User: admin
Password: 2k1LL23er3rlY1Hul4*zCF*lg*Ihv3Es
06/08/2025 07:25:57 INFO: --- Dependencies ---
06/08/2025 07:25:57 INFO: Removing gawk.

```

Fig 2.1 Credentials of Wazuh Web Interface

### 1.3 Verify Wazuh Services and Dashboard

To confirm that all Wazuh components were running properly, I ran:

```

bash
sudo systemctl status wazuh-manager

```

```
>Password:: command not found
spider@raven:~$ sudo systemctl status wazuh-manager
[sudo] password for spider:
● wazuh-manager.service - Wazuh manager
  Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2025-08-06 07:25:38 PKT; 2min 50s ago
    Tasks: 153 (limit: 4534)
   Memory: 659.0M
  CGroup: /system.slice/wazuh-manager.service
          ├─56025 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
          ├─56026 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
          ├─56029 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
          ├─56032 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
          ├─56073 /var/ossec/bin/wazuh-authd
          ├─56086 /var/ossec/bin/wazuh-db
          ├─56112 /var/ossec/bin/wazuh-execd
          ├─56126 /var/ossec/bin/wazuh-analysisd
          ├─56169 /var/ossec/bin/wazuh-syscheckd
          ├─56189 /var/ossec/bin/wazuh-remoted
          ├─56224 /var/ossec/bin/wazuh-logcollector
          └─56243 /var/ossec/bin/wazuh-monitord
              ├─56264 /var/ossec/bin/wazuh-modulesd
```

Figure 3 Checking status Wazuh Manager

- Wazuh Indexer installed and running
- Wazuh Dashboard installed and running

Then, I opened the dashboard on my browser:

```
https://192.168.137.153:443
```

Default login:

- *Username: admin*
- *Password: \*\*\**

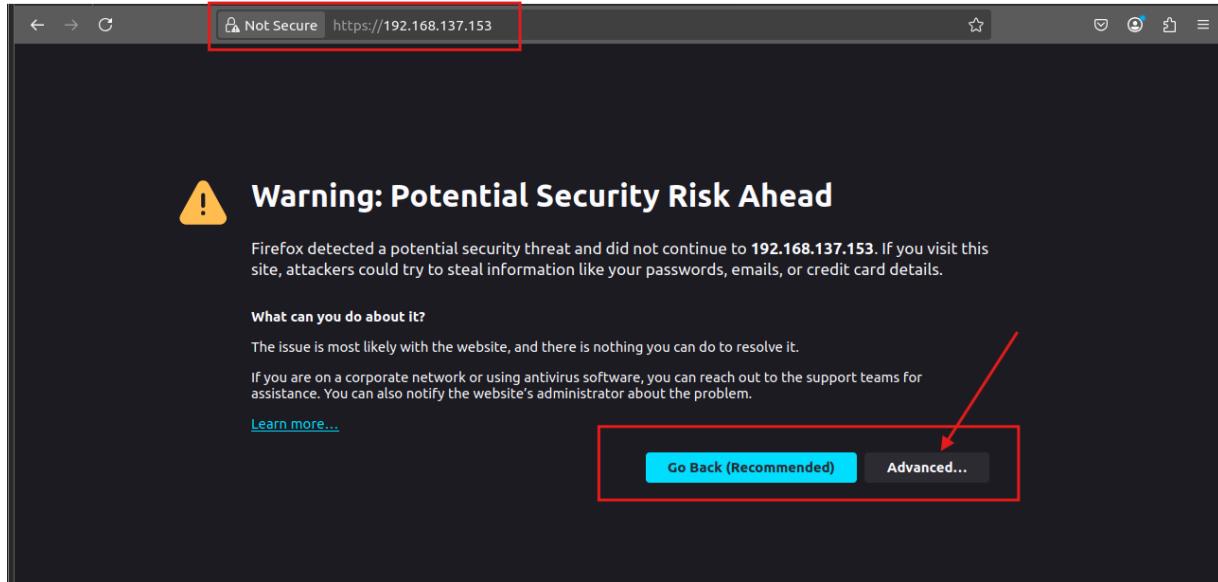


Figure 4 Launching Wazuh by clicking on Advanced

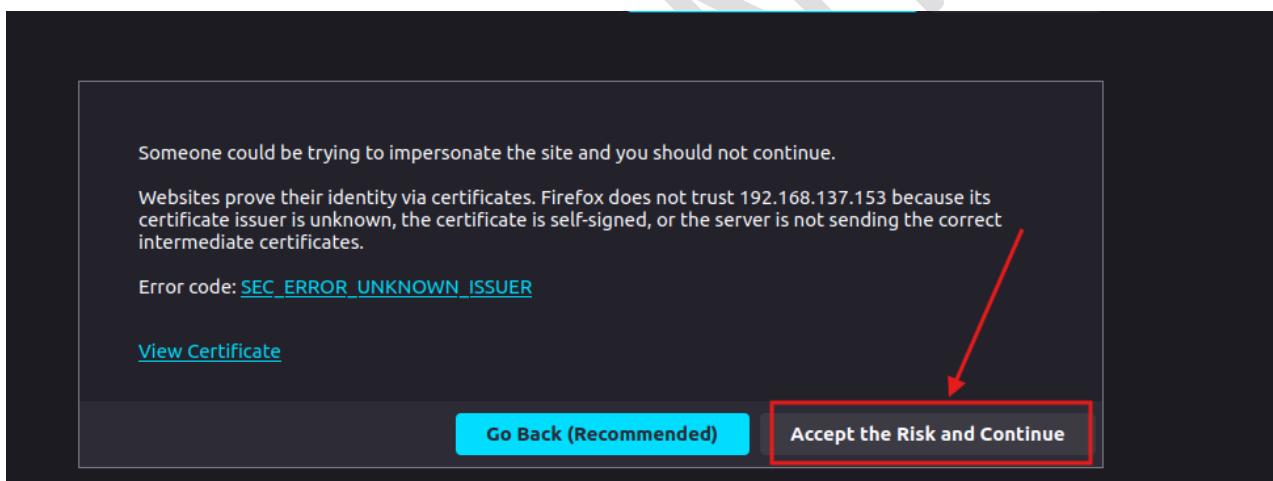


Figure 5 Click on Accept the Risk to Continue

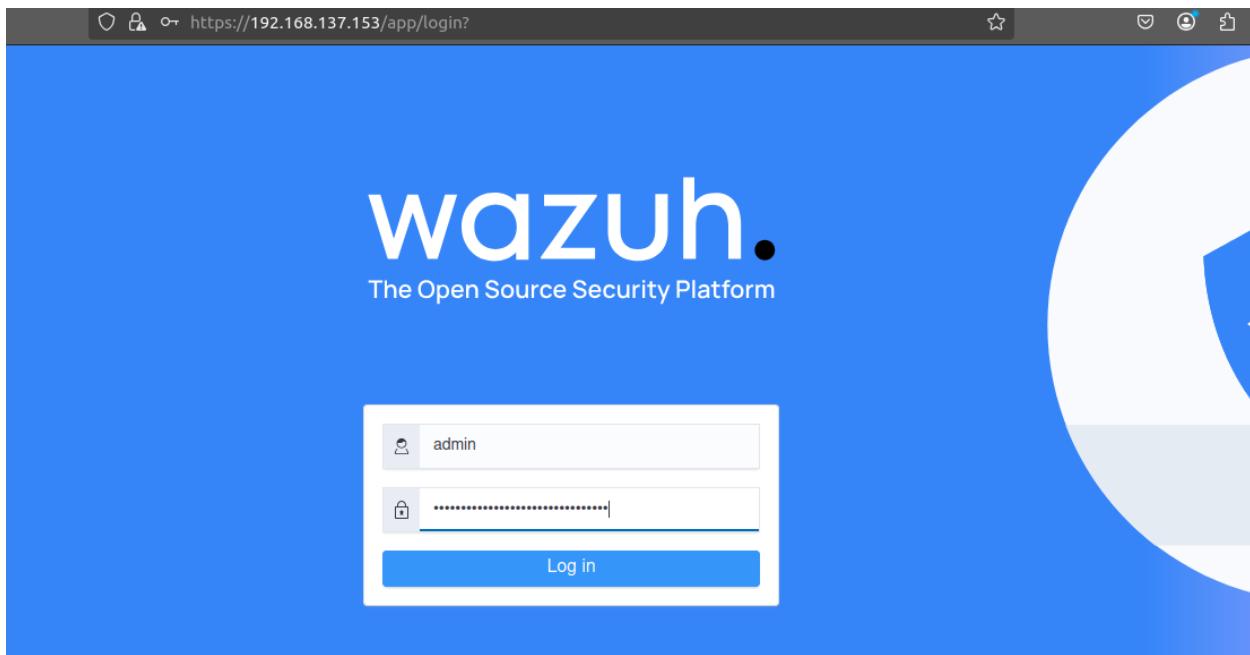


Figure 6 Login to Wazuh using default credentials which I get during Installation of Manager

A screenshot of the Wazuh interface dashboard. The URL in the address bar is https://192.168.137.153/app/wz-home#/overview?\_g=(filters:(),refreshInterval:(pause:0,value:0),time:(from:now-24h,to:now))&amp;\_a=(columns:[]). The dashboard includes sections for "AGENTS SUMMARY" (No results), "LAST 24 HOURS ALERTS" (0 Critical, 0 High, 120 Medium, 129 Low severity), "ENDPOINT SECURITY" (Configuration Assessment, Malware Detection, File Integrity Monitoring), and "THREAT INTELLIGENCE" (Threat Hunting, MITRE ATT&amp;CK, Vulnerability Detection, VirusTotal).

Figure 7 Wazuh Interface

## Step 2: Install Wazuh Agent on Windows 10

### 2.1 Download Wazuh Agent

I opened my Windows 10 machine (192.168.137.148) and downloaded the latest Windows Wazuh agent from the official site:

```
Windows PowerShell
PS C:\Users\R A V E N> Invoke-WebRequest -Uri "https://packages.wazuh.com/4.x/windows/wazuh-agent-4.12.0-1.msi" -OutFile "$env:USERPROFILE\Downloads\wazuh-agent-4.12.0-1.msi"
>>
PS C:\Users\R A V E N> Start-Process "C:\Windows\TEMP\wazuh-agent-4.12.0-1.msi" -ArgumentList "/q" -Wait
```

Figure 8 Installing Agent on Window 10 to use it as Endpoint System

After downloading, I double-clicked the **.msi** file to start the installation wizard.

### 2.2 Configure Agent to Connect to Ubuntu Wazuh Manager

During the installation wizard:

1. I selected **Next** until the **Server Address** prompt appeared.
2. I entered my **Wazuh Manager IP**: 192.168.137.153
3. I kept the **default port 1514** for communication.
4. I completed the installation with default options.

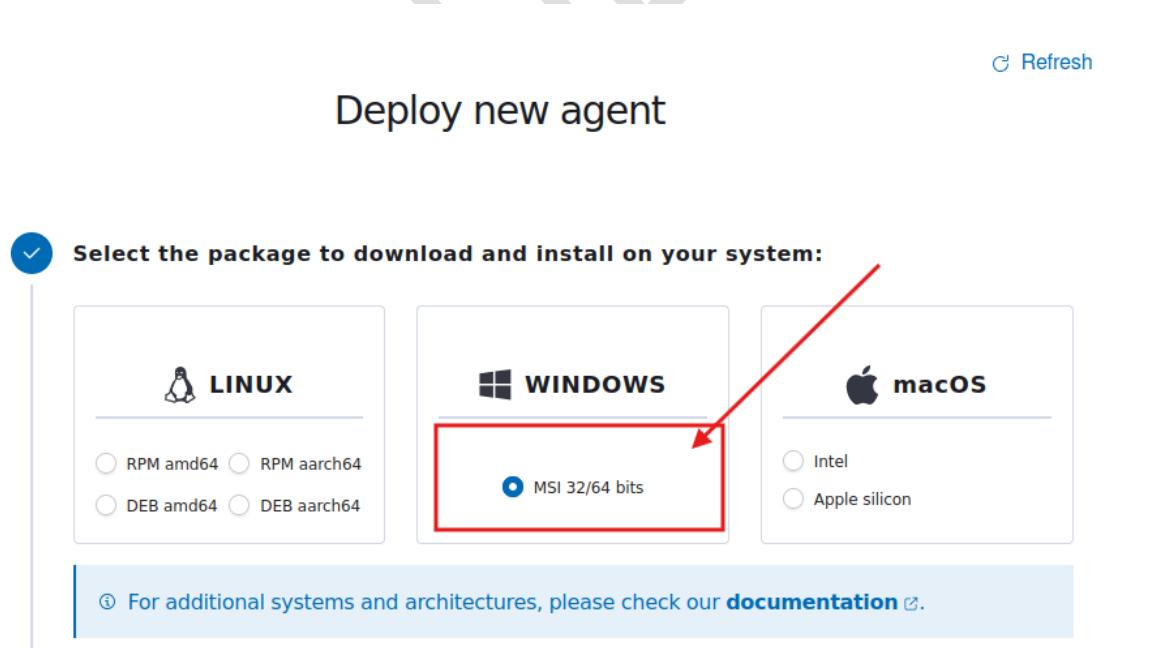


Figure 9 Deploying MSI 32/64 bits window Agent to connect Manager and Agent

**Server address:**

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

**Assign a server address** ?

Remember server address

Figure 10 Putting Server address which is Host(Ubuntu where Wazuh manager is running)

**Optional settings:**

By default, the deployment uses the hostname as the agent name. Optionally, you can use a different agent name in the field below.

**Assign an agent name:** ?

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

**Select one or more existing groups:** ?

Default

Figure 11 Setting name of Agent "Window 10"

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

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.8.2-1.msi -OutFile $(env.tmp)\wazuh-agent; msiexec.exe /i $(env.tmp)\wazuh-agent /q WAZUH_MANAGER='192.168.137.153' WAZUH_AGENT_NAME='Window10'
```

Figure 12 Getting Built in command from Wazuh manager to connect Window Agent



## 2.3 Start Agent and Verify Connection

After installation, I started the Wazuh agent service:

"`net start wazuhsvc`"



Figure 13 Copying the command and pasting in Shell of Window to Start the Wazuh service

```

Administrator: Windows PowerShell
PS C:\Windows\system32> Invoke-WebRequest -Uri "https://packages.wazuh.com/4.x/windows/wazuh-agent-4.8.2-1.msi" -OutFile "$env:USERPROFILE\Downloads\wazuh-agent.msi"
>>
PS C:\Windows\system32> msieexec.exe /i '$env:USERPROFILE\Downloads\wazuh-agent.msi' /q WAZUH_MANAGER="192.168.137.153" WAZUH_AGENT_NAME="Windows10"
>>
PS C:\Windows\system32> net start WazuhSvc
>>
The Wazuh service is starting.
The Wazuh service was started successfully.
PS C:\Windows\system32>

```

Figure 14 Installation and starting services of Wazuh in Window 10 Administrator PowerShell

Or, I could start it through **Services.msc** by locating **Wazuh Agent** and clicking **Start**.

To verify the agent connection to the manager:

1. I opened the Wazuh Dashboard on my browser:

"<http://192.168.137.153:443>"

2. I navigated to **Wazuh → Agents** and saw the **Windows 10 machine listed as "*active*"**.

The screenshot shows the Wazuh Manager's 'Endpoints' section. At the top, there are three tabs: 'Status', 'Details', and 'Evolution'. The 'Status' tab is selected, displaying a summary of agent status: Active (1), Disconnected (0), Pending (0), and Never connected (0). Below this, it shows 'Active' (1), 'Disconnected' (0), 'Pending' (0), 'Never connected' (0), and 'Agents coverage' at 100.00%. It also lists the 'Last enrolled agent' as 'Window10' and the 'Most active agent' as 'Window10'. The 'Evolution' tab shows a chart for the last 24 hours. Below the summary, a table titled 'Agents (1)' lists one agent: '001 Window10' with IP '192.168.137.162' and operating system 'Microsoft Windows 10 Pro N 10.0.19045.6093'. The 'Operating system' row is highlighted with a red box. The table includes columns for ID, Name, IP address, Group(s), Operating system, Cluster node, Version, Status, and Actions.

Figure 15 Actie Wazuh Agent and Analyzing the system details

**Step 2 is complete.** My Windows 10 endpoint is now securely connected to the Wazuh Manager.

## Step 3: Install and Configure Sysmon on Windows 10

### 3.1 Download Sysmon

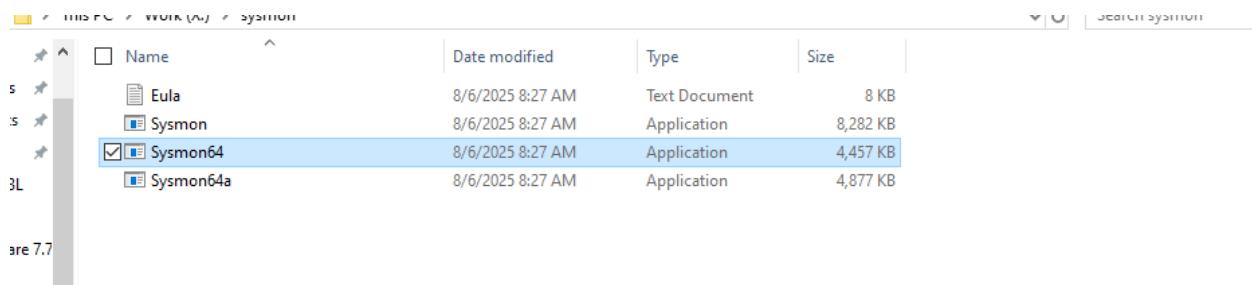
I started by downloading **Sysmon** from the official Microsoft Sysinternals page:

[“<https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon>”](https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon)

The screenshot shows the Microsoft Learn page for Sysmon v15.15. The URL is learn.microsoft.com/en-us/sysinternals/downloads/sysmon. The page title is 'Sysmon v15.15'. It was published on July 23, 2024, by Mark Russinovich and Thomas Garnier. There are download links for 'Download Sysmon (4.6 MB)' and 'Download Sysmon for Linux (GitHub)'. The left sidebar has a navigation menu with sections like Home, Downloads, File and Disk Utilities, Networking Utilities, Process Utilities, Security Utilities, Autologon, LogonSessions, NewSID, PsLoggedOn, PsLogList, RootkitRevealer, and Sysmon. The 'Sysmon' link is currently selected. The right sidebar contains an 'In this article' section with links to Introduction, Overview of Sysmon Capabilities, Screenshots, Usage, Examples, Events, Configuration files, Configuration Entries, and Event filtering entries.

Figure 16 Installing Sysmon V15.15 in Window to monitor Logs and Activity

I extracted the ZIP file to a folder.



Name	Date modified	Type	Size
Eula	8/6/2025 8:27 AM	Text Document	8 KB
Sysmon	8/6/2025 8:27 AM	Application	8,282 KB
<input checked="" type="checkbox"/> Sysmon64	8/6/2025 8:27 AM	Application	4,457 KB
Sysmon64a	8/6/2025 8:27 AM	Application	4,877 KB

Figure 17 Sysmon files Extracted

### 3.2 Download Sysmon Configuration File

To capture high-quality logs, I downloaded the **SwiftOnSecurity Sysmon configuration file**, which is widely used in enterprise environments:

[“https://raw.githubusercontent.com/SwiftOnSecurity/sysmon-config/master/sysmonconfig-export.xml”](https://raw.githubusercontent.com/SwiftOnSecurity/sysmon-config/master/sysmonconfig-export.xml)

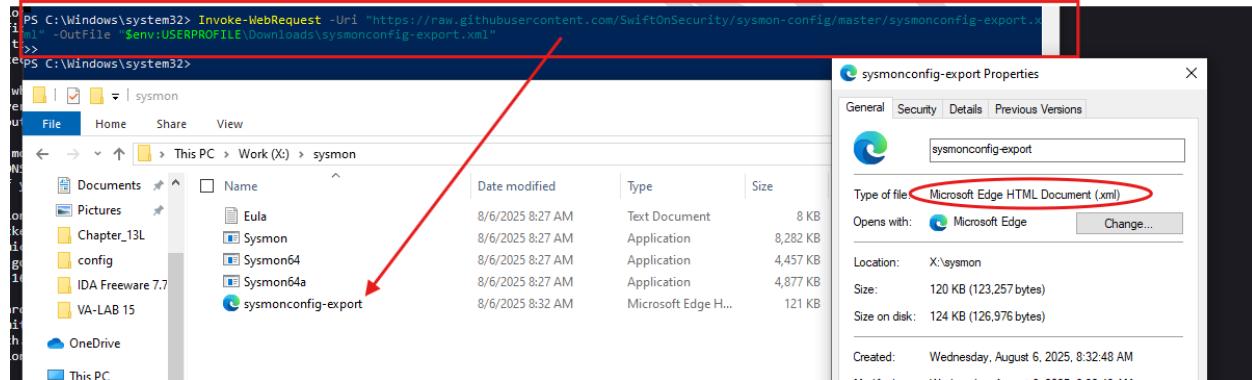


Figure 18 Installing Sysmon Configuration File.xml

I saved it in the same folder as Sysmon binaries for easy access.

### 3.3 Install Sysmon with Configuration

I opened **PowerShell as Administrator** in the Sysmon folder and installed Sysmon with the configuration file using:

```
powershell
Sysmon64.exe -accepteula -i sysmonconfig-export.xml
```

- **-accepteula**: Automatically accepts the license agreement.
- **-i**: Installs Sysmon as a service with the specified configuration.

After installation, I verified that Sysmon was running as a Windows service.

```

PS X:\sysmon> .\Sysmon64.exe -accepteula -i sysmonconfig-export.xml
>>

System Monitor v15.15 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.50
Sysmon schema version: 4.90
Configuration file validated.
Sysmon64 installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon64..
Sysmon64 started.
PS X:\sysmon>

```

Figure 19 Installing Sysmon .exe and sysconfig-export.xml. After Installation starting Sysmon

### 3.4 Verify Sysmon Event Logging

To confirm Sysmon was generating logs, I opened **Event Viewer**:

1. Pressed **Win + R**, typed **eventvwr.msc**, and hit Enter.
2. Navigated to:

**"Applications and Services Logs → Microsoft → Windows → Sysmon → Operational"**

3. I verified that Sysmon was logging events such as **Process Creation, Network Connections, and File Changes**.

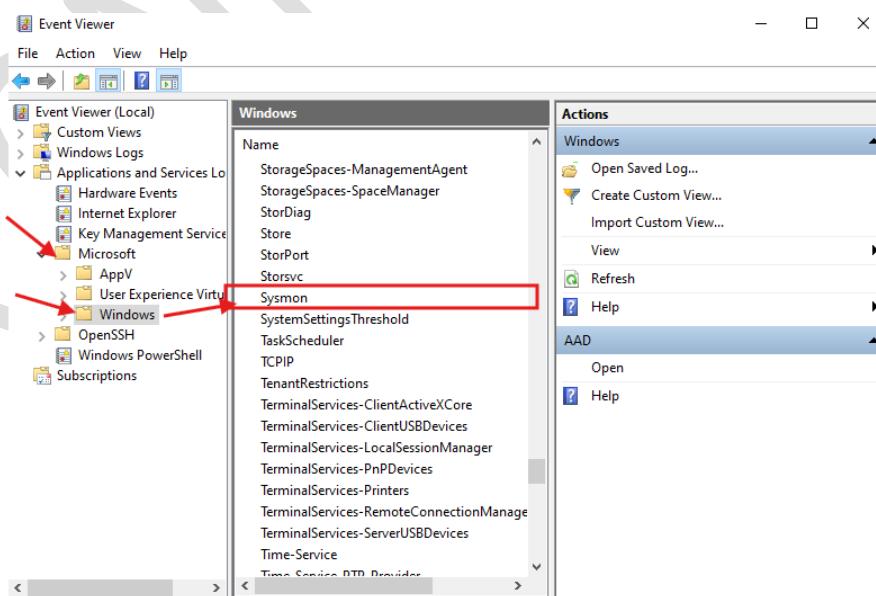


Figure 20 Verifying the service of Sysmon Logging



## Step 3 is complete.

I now have **Sysmon installed and actively monitoring process, network, and file activities** on my Windows 10 endpoint. These logs will enhance threat detection in Wazuh.

## Step 4: Implement File Integrity Monitoring (FIM) in Wazuh

### 4.1 Purpose of File Integrity Monitoring

I implemented **File Integrity Monitoring (FIM)** to detect **unauthorized modifications, creations, or deletions** of critical files and directories on my Windows 10 endpoint. This enhances security by alerting me to potential malware activity or insider threats.

- *Wazuh Dashboard → Security Events → File Integrity Monitoring section*

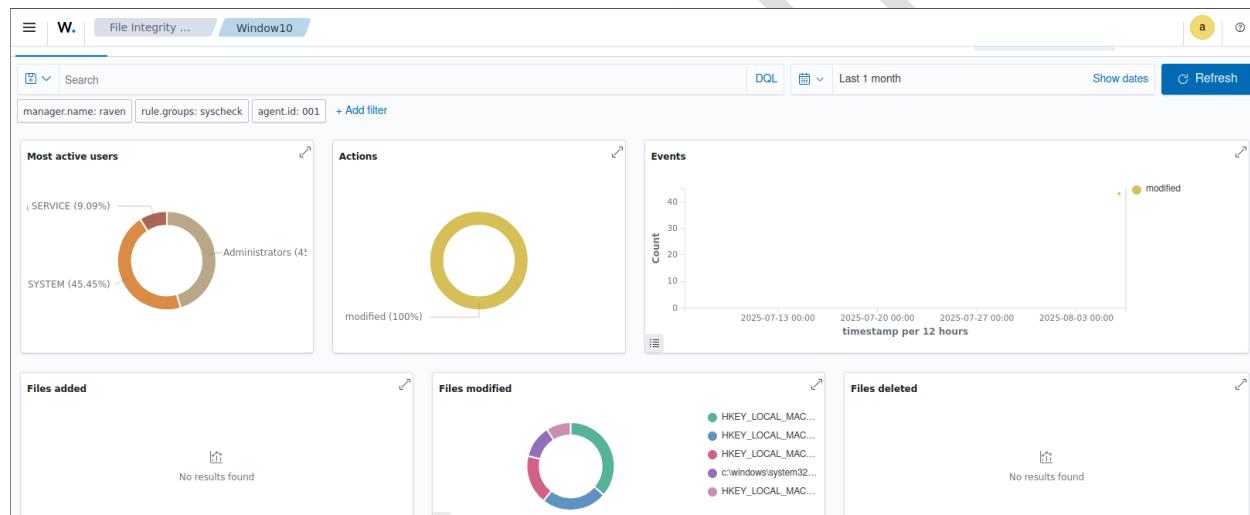


Figure 21 Checking security Events of Window 10. Most Active, Actions and file modifications

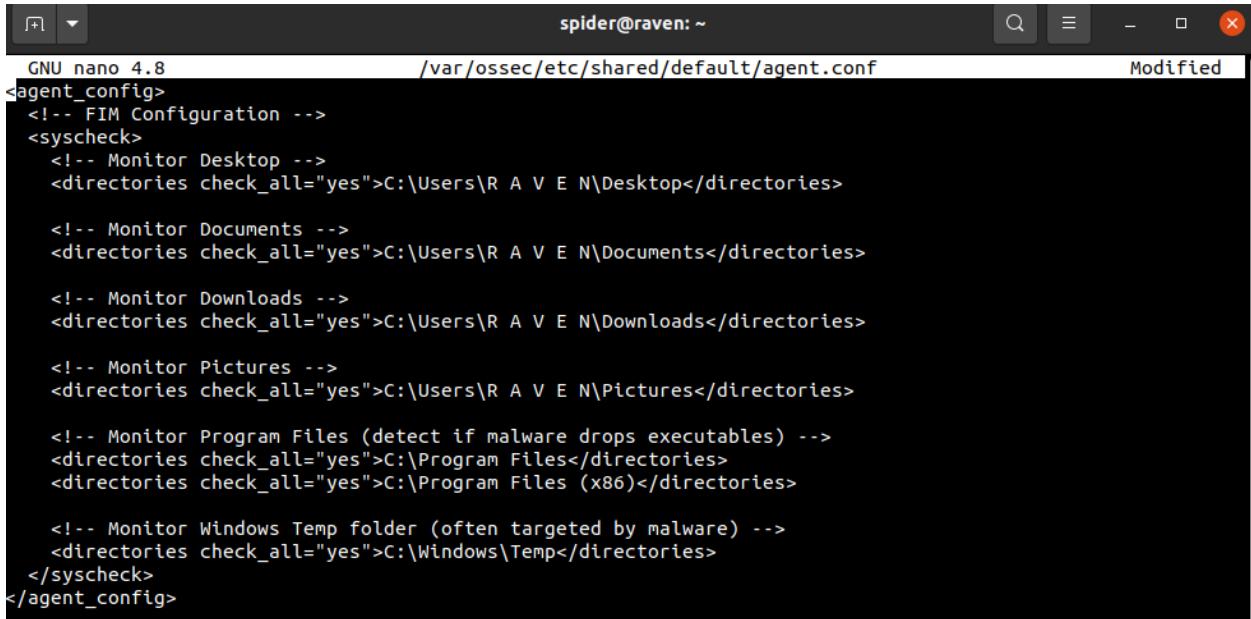
### 4.2 Configure FIM on Wazuh Manager

I configured the Wazuh Manager to monitor specific directories.

1. I logged in to my **Ubuntu 20.04 Wazuh Manager (192.168.137.153)**.
2. I edited the **ossec.conf** file using nano:

**“sudo nano /var/ossec/etc/ossec.conf”**

3. I located the **<syscheck>** section and added directories to monitor.  
Example configuration for **Windows system files**:
4. I saved the file and exited nano.



```

GNU nano 4.8                               /var/ossec/etc/shared/default/agent.conf
agent_config>
<!-- FIM Configuration -->
<syscheck>
  <!-- Monitor Desktop -->
  <directories check_all="yes">C:\Users\R A V E N\Desktop</directories>

  <!-- Monitor Documents -->
  <directories check_all="yes">C:\Users\R A V E N\Documents</directories>

  <!-- Monitor Downloads -->
  <directories check_all="yes">C:\Users\R A V E N\Downloads</directories>

  <!-- Monitor Pictures -->
  <directories check_all="yes">C:\Users\R A V E N\Pictures</directories>

  <!-- Monitor Program Files (detect if malware drops executables) -->
  <directories check_all="yes">C:\Program Files</directories>
  <directories check_all="yes">C:\Program Files (x86)</directories>

  <!-- Monitor Windows Temp folder (often targeted by malware) -->
  <directories check_all="yes">C:\Windows\Temp</directories>
</syscheck>
</agent_config>

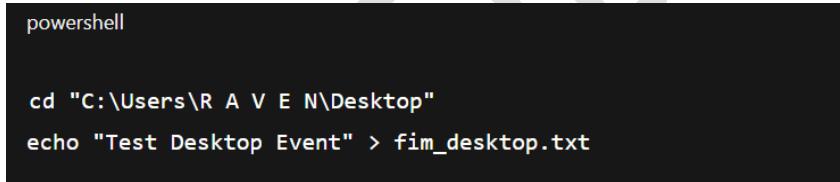
```

Figure 22 Adding Directories to monitor all activity

## 4.4 Verify FIM Events on Wazuh Dashboard

I tested the FIM feature by **creating a new file** in one of the monitored directories on my Windows 10 machine:

### 1) Desktop:



```

powershell

cd "C:\Users\R A V E N\Desktop"
echo "Test Desktop Event" > fim_desktop.txt

```

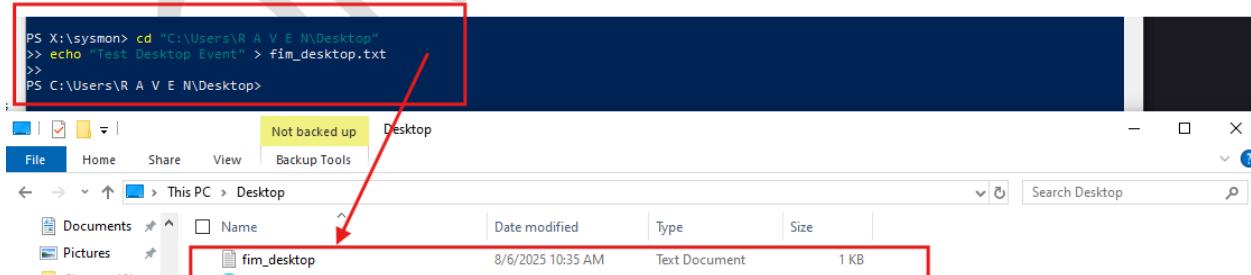


Figure 23 creating random file in window to insure and grab activity in Wazuh

## 2) Downloads:

```
powershell

cd "C:\Users\R A V E N\Downloads"
echo "Test Download Event" > fim_download.txt
```

Figure 24 Creating File in downloader folder

Temp folder:

```
powershell

cd C:\Windows\Temp
echo "Test Temp Event" > fim_temp.txt
```

config	nsz914A.tmp	8/6/2025 8:22 AM	File folder
IDA Freeware 7.7	vmware-SYSTEM	5/19/2025 2:54 AM	File folder
VA-LAB 15	WinSAT	8/6/2025 5:54 AM	File folder
	bb3a785178f443fd931098a5a9a306b.d...	8/6/2025 7:57 AM	SES File
	<input checked="" type="checkbox"/> fim_temp	8/6/2025 10:39 AM	Text Document
	FASAPIRebugLogfile	8/27/2025 10:15 PM	Text Document
	MpCmdRun	8/6/2025 8:18 AM	Text Document
	MpSigStub	8/6/2025 7:57 AM	Text Document
	nfa94BC.tmp	8/6/2025 8:22 AM	TMP File
	secexport	8/6/2025 10:30 AM	CFG File

Figure 25 creating fim\_temp file in temp directory of window to make sure the activity captured in Events

Then **modify and delete** to generate more events:

```
powershell

echo "Modified Event" >> fim_temp.txt
del fim_temp.txt
```

```
PS C:\Windows\Temp> echo "Modified Event" >> fim_temp.txt
>> del fim_temp.txt
>>
PS C:\Windows\Temp>
```

Figure 26 Modifying the file and after that deleting to monitor all Events of Agent Wazuh as an Endpoint Device

Then I logged in to the **Wazuh Dashboard** → **Security Events** → **File Integrity Monitoring** and verified that:

- **A new file creation event** appeared.
- The agent **Windows10** generated the alert successfully.

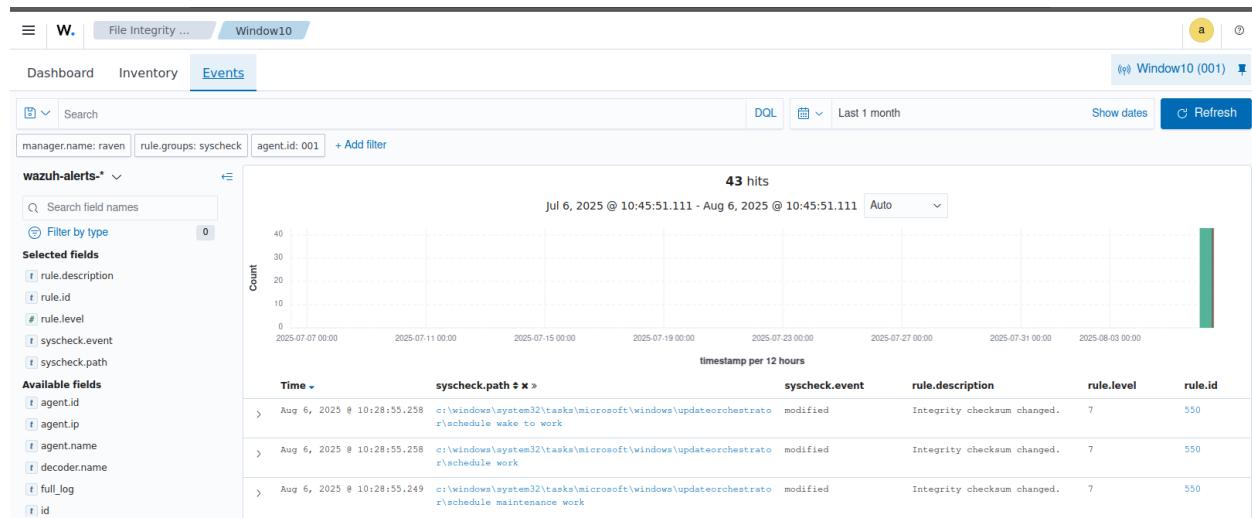


Figure 27 43 hits (shows what type of changes are made on Endpoint)

input.type	Time	path	event	description	level	ID
	> Aug 6, 2025 @ 10:28:51.580	c:\windows\system32\sleepstudy\usernotpresentsession.etl	modified	Integrity checksum changed.	7	550
	> Aug 6, 2025 @ 10:27:07.608	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits	modified	Registry Value Integrity Checksum Changed	5	750
	> Aug 6, 2025 @ 10:27:07.582	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits	modified	Registry Value Integrity Checksum Changed	5	750
	> Aug 6, 2025 @ 10:27:07.566	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits	modified	Registry Value Integrity Checksum Changed	5	750
	> Aug 6, 2025 @ 10:27:07.552	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits	modified	Registry Key Integrity Checks um Changed	5	594
	> Aug 6, 2025 @ 10:27:07.542	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits\RunTime	modified	Registry Value Integrity Checksum Changed	5	750
	> Aug 6, 2025 @ 10:27:07.542	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits\RunTime	modified	Registry Key Integrity Checks um Changed	5	594
	> Aug 6, 2025 @ 10:24:25.794	c:\windows\system32\logfiles\wmi\netcore.etl	modified	Integrity checksum changed.	7	550
	> Aug 6, 2025 @ 10:24:25.646	c:\windows\system32\logfiles\wmi\lwt.netlog.etl	modified	Integrity checksum changed.	7	550
	> Aug 6, 2025 @ 10:22:53.195	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits	modified	Registry Value Integrity Checksum Changed	5	750
	> Aug 6, 2025 @ 10:22:53.179	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits	modified	Registry Value Integrity Checksum Changed	5	750

Figure 27.1 Another further details of Figure 27.

### • Top Bar:

- Dashboard shows **File Integrity Monitoring → Events Tab**.
- Agent:** Window10 (001) (your Windows 10 endpoint is successfully connected).
- Time Filter:** Last 1 month.

### • Middle Graph:

- 43 hits** → This means **43 file integrity events** have been detected in the selected time range.
- Bar at the right side of the graph** → Indicates recent events, likely triggered today.



## Event Table Explanation

The lower section lists **each FIM event**. Columns mean:

- **Time:**
  - Timestamp when the file change was detected.
  - Example: Aug 6, 2025 @ 10:28:55.258 → Happened today at 10:28 AM.
- **syscheck.path:**
  - Full path of the file that changed.
  - Example:

**"c:\windows\system32\tasks\microsoft\windows\updateorchestrator\schedule wake to work"**

- This shows that a **Windows scheduled task** file was modified.
- **syscheck.event:**
  - Type of change detected:
    - **added** → New file created.
    - **modified** → File was changed.
    - **deleted** → File was removed.
  - In your snapshot: **modified** → File content or checksum changed.
- **rule.description:**
  - Wazuh's interpretation of the event.
  - Example: **Integrity checksum changed** → Means the file's hash changed (indicating modification).
- **rule.level:**
  - Severity of the event (1–15).
  - Here it's **7**, which is **medium priority**.
  - Changes in **C:\Windows\System32** are important to watch because malware often targets this directory.
- **rule.id (550):**
  - ID of the rule in Wazuh that triggered the alert.

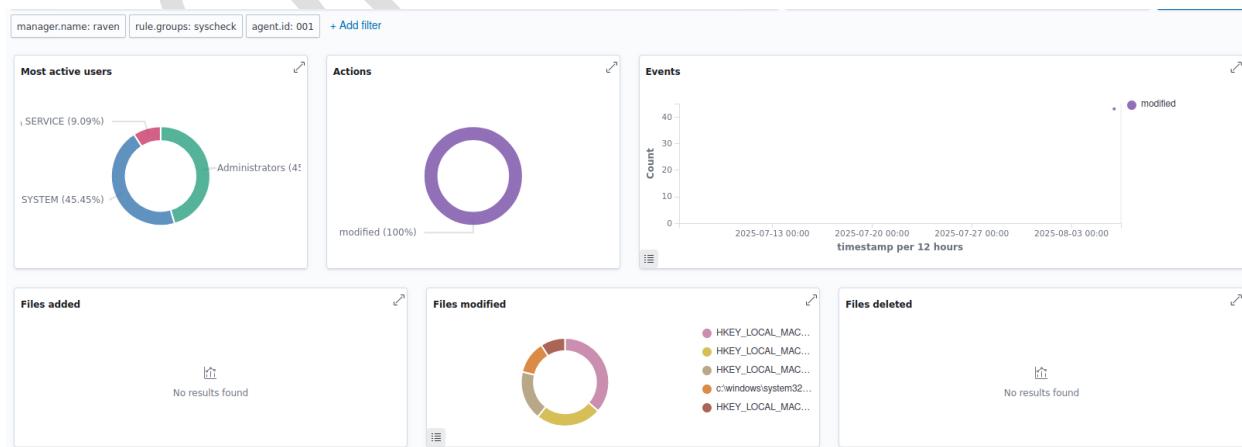


Figure 28 Captured Activity of End Point system

## Step 4 is complete.

I now have **real-time file integrity monitoring** active, and any unauthorized change will trigger an alert in Wazuh.

## Step 5: Configure Security Alerts and Automated Responses in Wazuh

### 5.1 Objective of Alerts and Automation

I configured **alerts and automated responses** in Wazuh to:

- Detect **suspicious behavior** like failed logins, file tampering, or malware indicators.
- Automatically **trigger notifications** or **run responses** when threats are detected.

This ensures **real-time visibility** and **quick reaction** to incidents.

ID	Description	Groups	Regulatory compliance	Level	File	Path
1	Generic template for all syslog rules.	syslog		0	0010-rules_config.xml	ruleset/rules
2	Generic template for all firewall rules.	firewall		0	0010-rules_config.xml	ruleset/rules
3	Generic template for all ids rules.	ids		0	0010-rules_config.xml	ruleset/rules
4	Generic template for all web rules.	web-log		0	0010-rules_config.xml	ruleset/rules
5	Generic template for all web proxy rules.	squid		0	0010-rules_config.xml	ruleset/rules
6	Generic template for all windows rules.	windows		0	0010-rules_config.xml	ruleset/rules
7	Generic template for all wazuh rules.	ossec		0	0010-rules_config.xml	ruleset/rules
200	Grouping of wazuh rules.	wazuh		0	0016-wazuh_rules.xml	ruleset/rules
201	Agent event queue rule	agent_flooding, wazuh		0	0016-wazuh_rules.xml	ruleset/rules
202	Agent event queue is <b>level</b> full.	agent_flooding, ...	PCI_DSS, GDPR	7	0016-wazuh_rules.xml	ruleset/rules

Figure 29 Rules to Alert which shows real-time visibility and quick reaction to incidents

Name	Program name	Order	File	Path
wazuh			0005-wazuh_decoders.xml	ruleset/decoders
agent-buffer		level	0005-wazuh_decoders.xml	ruleset/decoders
agent-upgrade		agent.id, agent.name, status	0005-wazuh_decoders.xml	ruleset/decoders
agent-upgrade		error	0005-wazuh_decoders.xml	ruleset/decoders
agent-upgrade		agent.cur_version	0005-wazuh_decoders.xml	ruleset/decoders
agent-upgrade		agent.new_version	0005-wazuh_decoders.xml	ruleset/decoders
agent-restart		module	0005-wazuh_decoders.xml	ruleset/decoders
firm-state			0005-wazuh_decoders.xml	ruleset/decoders
json			0006-json_decoders.xml	ruleset/decoders
wazuh-api			0007-wazuh-api_decoders.xml	ruleset/decoders

Figure 29.1 Decoders Alerts



## 5.2 Review Default Wazuh Rules

Wazuh comes with **pre-configured rules** that generate alerts for:

- **File Integrity Monitoring events (FIM)**
- **System log anomalies**
- **Unauthorized access attempts**
- **Malware or suspicious behavior detected via Sysmon logs**

To view these rules, I navigated to:

**“cd /var/ossec/rules**

**ls”**

- Default rules:

```
GNU nano 4.8                               /var/ossec/etc/rules/local_rules.xml
<!-- Local rules -->
<!-- Modify it at your will. -->
<!-- Copyright (C) 2015, Wazuh Inc. -->
<!-- Example -->
<group name="local,syslog,sshd,">

<!--
Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
-->
<rule id="100001" level="5">
<if_sid>5716</if_sid>
<srcip>1.1.1.1</srcip>
<description>sshd: authentication failed from IP 1.1.1.1.</description>
<group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
</rule>
</group>
```

Figure 30 Pre define defaults rules

FIM Rules:

```
root@raven: ~
GNU nano 4.8                               /var/ossec/etc/rules/local_rules.xml
<!-- Local custom rules -->
<group name="local,syscheck,">

<!-- Trigger when a file is added -->
<rule id="100100" level="7">
<if_group>syscheck</if_group>
<field name="syscheck.event">added</field>
<description>File ADDED: ${file} in monitored directory</description>
</rule>

<!-- Trigger when a file is deleted -->
<rule id="100101" level="7">
<if_group>syscheck</if_group>
<field name="syscheck.event">deleted</field>
<description>File DELETED: ${file} from monitored directory</description>
</rule>

<!-- Trigger when a file is modified -->
<rule id="100102" level="7">
<if_group>syscheck</if_group>
<field name="syscheck.event">modified</field>
<description>File MODIFIED: ${file} in monitored directory</description>
</rule>
</group>
```

Figure 31 Writing new FIM rules to ADD, Delete and Modify file



### 5.3 Configure Email or Slack Alerts (Optional)

To receive notifications, I can configure:

1. Email Alerts – by editing the Wazuh configuration:

**“sudo nano /var/ossec/etc/ossec.conf”**

I ensured **<global>** section includes my SMTP settings:

```
GNU nano 4.8                               /var/ossec/etc/ossec.conf
-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>

  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

  <!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->

```

Figure 32 SMTP Rules setting for Emails

### 5.4 Enable Active Response

I enabled **Active Responses** to automatically block malicious behavior, such as IP addresses performing brute force attacks.

**Example:** Block repeated SSH failures on Linux or RDP on Windows:

1. Edit active-response configuration:

**“sudo nano /var/ossec/etc/ossec.conf”**

- Add:



```

root@raven: ~
/var/ossec/etc/ossec.conf
Modified

-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>

  <active-response>
    <command>firewall-drop</command>
    <location>local</location>
    <level>10</level>
  </active-response>

  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

```

Figure 33 Updating Script by adding Firewall level 10 active response

### **Restart Wazuh Manager:**

**“sudo systemctl restart wazuh-manager”**

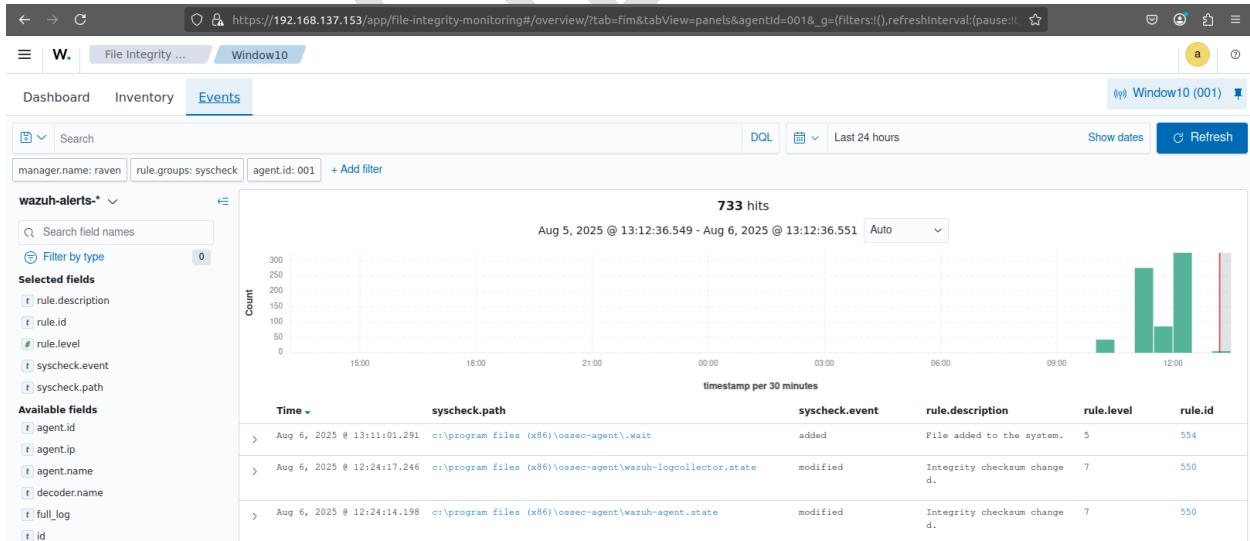


Figure 34 733 Hits occurs after changing the rules and creating files to Analyze Event.



## 5.5 Validate Alerts

To test my alerts, I performed a **failed login attempt** on Windows 10:

1. I deliberately entered the wrong password 3 times.
2. Wazuh dashboard showed a **security event alert** for multiple failed logins.
3. If active response was enabled, the IP could be **blocked automatically**.
4. **Step 5 is complete.**

I now have **real-time alerts and optional automated responses** configured to monitor and react to suspicious activity efficiently.

## Step 6: Integrate Threat Intelligence with MalwareBazaar in Wazuh

### 6.1 Objective of Threat Intelligence Integration

I integrated **MalwareBazaar threat intelligence feeds** into Wazuh to:

- Detect **known malware** using file hashes or indicators of compromise (IOCs).
- Enhance Wazuh alerts with **external intelligence**, improving incident response.

### 6.2 Download MalwareBazaar Threat Feed

I used **Ubuntu 20.04 Wazuh Manager (192.168.137.153)** to fetch the **daily hash feed** from MalwareBazaar:

```
root@raven:/var/ossec/etc/lists# sudo curl -O https://bazaar.abuse.ch/export/txt/sha256/recent
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload   Total Spent  Left Speed
100 64064  100 64064    0     0  64515      0 --:--:-- --:--:-- 64645
root@raven:/var/ossec/etc/lists# sudo mv recent index_malware_hashes.txt
root@raven:/var/ossec/etc/lists# head index_malware_hashes.txt
#####
# MalwareBazaar recent malware samples (SHA256 hashes)
# Last updated: 2025-08-06 08:11:08 UTC
#
# Terms Of Use: https://bazaar.abuse.ch/fan/#tos
# For questions please contact bazaar [at] abuse.ch
#
#
# sha256_hash
e37ea155cad7c8b1996fdf0e26b681cf11a732a31ed4b2f5a3c3d6b9ef974914
root@raven:/var/ossec/etc/lists#
```

Figure 35

- This creates a **list of malicious hashes** that Wazuh can use for detection.

### 6.3 Configure Wazuh to Use Threat Feed

I linked the downloaded feed with Wazuh's **CDB list**:



## 1. Edit ***ossec.conf***:

```
"sudo nano /var/ossec/etc/ossec.conf"
```

```

GNU nano 4.8
/var/ossec/etc/ossec.conf

<!--
Wazuh - Manager Configuration File
Clean and validated configuration for Ubuntu 20.04
-->

<ossec_config>
  <!-- Global settings -->
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>

  <!-- Alerts -->
  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

  <!-- Logging -->
  <logging>
    <log_format>%d %p %m</log_format>
  </logging>
</ossec_config>

```

Figure 36 *ossec.conf* which contains the list/script of Alerts, Logging SMTP etc

## 2. Add the new list under the ***<ruleset>*** section:

```

<><ruleset>
  <list>lists/index_malware_hashes.txt</list>
</ruleset>

  , commands
  <!-- Active Response Configuration -->
  <active-response>
    <disabled>no</disabled>
    <command>firewall-drop</command>
    <location>local</location>
    <level>10</level>
  </active-response>

  <!-- Ruleset for MalwareBazaar Threat Intel -->
  <ruleset>
    <decoder_dir>ruleset/decoders</decoder_dir>
    <rule_dir>ruleset/rules</rule_dir>
    <rule_exclude>0215-policy_rules.xml</rule_exclude>
    <rule_exclude>0216-nginx_rules.xml</rule_exclude>

    <!-- MalwareBazaar Threat Feed -->
    <list>lists/index_malware_hashes.txt</list>
  </ruleset>

</ossec_config>

```

Figure 37 Adding and setting new Rules of Malware Threat feed

## 6.4 Create a Detection Rule for Malware Hashes

To trigger alerts when a **file hash** matches MalwareBazaar IOCs, I created a **custom rule**:

- Create a new rule file:**

```
"sudo nano /var/ossec/etc/rules/local_rules.xml"
```

- Add the rule:**

```
<!-- Trigger when a file is modified -->
<rule id="100102" level="7">
  <if_group>syscheck</if_group>
  <field name="syscheck.event">modified</field>
  <description>File MODIFIED: ${file} in monitored directory</description>
</rule>

</group>

<!-- MalwareBazaar Threat Intelligence Rule -->
<group name="malwarebazaar,syscheck,">
  <rule id="100200" level="12">
    <if_sid>550</if_sid> <!-- Syscheck file integrity event -->
    <list field="hash_sha256" lookup="match">lists/index_malware_hashes.txt</list>
    <description>Malware file detected using MalwareBazaar Threat Intelligence</description>
    <mitre>
      <id>T1059</id> <!-- Example MITRE technique -->
    </mitre>
  </rule>
</group>
```

Figure 38 Adding Levels to check systems integrity's

- Save the file and restart Wazuh:**

```
"sudo systemctl restart wazuh-manager"
```

## 6.5 Test Threat Intelligence Detection

I verified the integration by creating a **dummy file** with a hash from MalwareBazaar's feed:

```
PS C:\Users\R A V E N\Desktop> New-Item -ItemType Directory -Path C:\Temp
>>

Directory: C:\

Mode                LastWriteTime     Length Name
----                -----        ----- 
d-----          8/6/2025 1:57 PM           Temp

PS C:\Users\R A V E N\Desktop> echo "malware_test" > C:\Temp\malwaretest.txt
>>
PS C:\Users\R A V E N\Desktop> Get-FileHash C:\Temp\malwaretest.txt -Algorithm SHA256
>>

Algorithm      Hash
-----      -----
SHA256        A6CFC960757A08649DAF4BB7DECCAF2BFC64BF0845D0C7BB05EAFBBCCC23C6B2
                                         Path
                                         -----
                                         C:\Temp\malwaretest.txt
```

Figure 39 Creating Temp MalwareBazaar file "malware\_test"

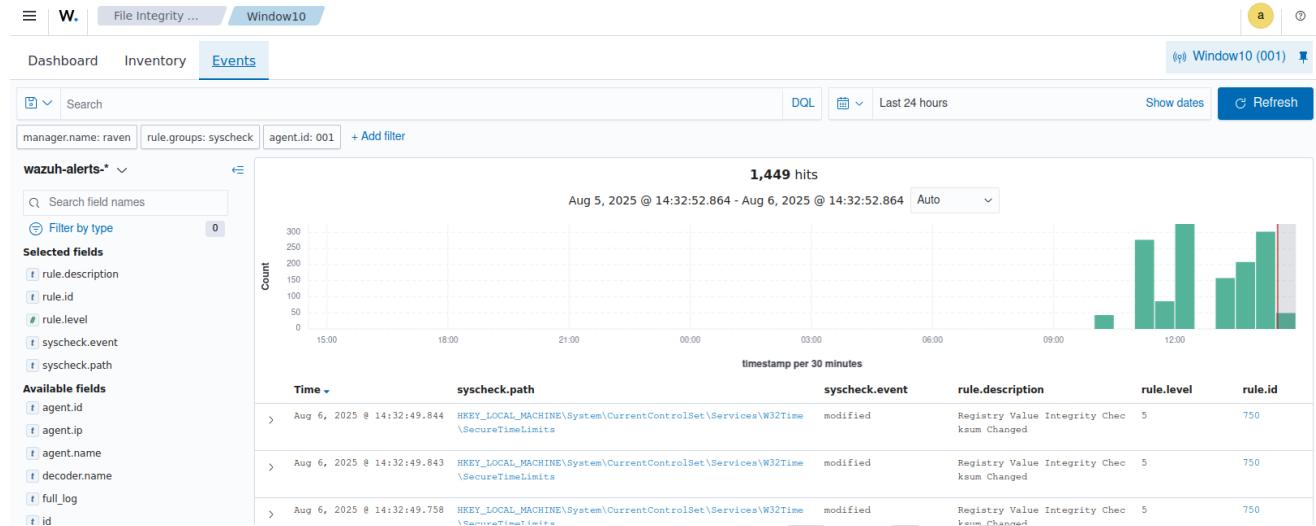


Figure 40 Analyzing the last stage of Hit Events after Performing ALL 1-6 required and assigned steps

HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits

- > Details
- Registry values

Date ↓	Value name	Value type	sha1
Aug 6, 2025 @ 14:32:47.000	SecureTimeLow	REG_QWORD	d55efa478d920ee5e4f7b6c2062d4cf6ec53b304
Aug 6, 2025 @ 14:32:47.000	SecureTimeHigh	REG_QWORD	3c8b34daaeea703bd1cbaa0162428927b8b29c2a
Aug 6, 2025 @ 14:32:47.000	SecureTimeEstimated	REG_QWORD	a46931d9b641588e4df95a0ad7027452ca25a3a1

Rows per page: 5 < 1 >

- Recent events 16 hits

Search DQL Last 24 hours Show dates Refresh

Figure 41.1 The following snapshot shows the detail information of Sha1 form the file which is created temp directory on Endpoint system.

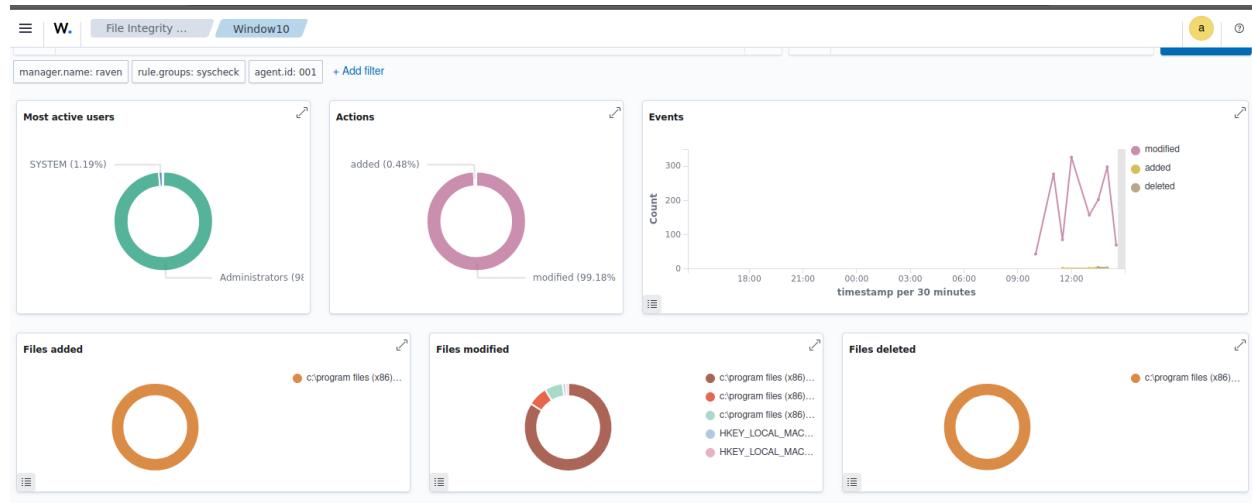


Figure 41.2 The Final Snapshot shows File Integrity's Most active, Actions, Events like Files added, Modified and Deleted.

## 7. Conclusion

The implementation of Wazuh as an EDR platform, coupled with Sysmon logging, successfully delivered a layered defense system that strengthens endpoint security and visibility. By deploying the Wazuh Manager on Ubuntu and integrating a Windows 10 endpoint, I achieved centralized monitoring and real-time alerting for critical security events. Sysmon logs enhanced detection capabilities by providing deep insights into process executions, network activities, and file changes, which significantly reduced blind spots in endpoint monitoring.

File Integrity Monitoring was configured to detect unauthorized file changes, and automated alerting mechanisms were introduced to respond swiftly to abnormal behaviors. Active response capabilities further ensured that specific threats, such as brute-force attempts, could be mitigated instantly without manual intervention. The integration of MalwareBazaar threat intelligence added an additional layer of proactive defense, allowing for rapid identification of known malware through hash-based detection.

This project demonstrates the importance of combining real-time monitoring, behavioral analysis, and external threat intelligence for a robust cybersecurity posture. By completing this implementation, I successfully created an operational framework that enables early detection, automated responses, and improved situational awareness—critical components for reducing risk and ensuring system resilience against modern cyber threats.