

## Sysmon Integration with Wazuh

### Sysmon:

Sysmon (System Monitor) is a Windows service developed by Microsoft that continuously monitors internal system activity and generates high-detail security logs.

Default Windows logs tell you *that* something happened.

Sysmon logs explain *what happened, how it happened, who initiated it, and what followed.*

Sysmon captures low-level system behavior that attackers commonly abuse, such as process execution, parent-child relationships, network connections, file creation, and registry changes details that standard Windows logging often misses.

Because of this depth, Sysmon provides visibility before, during, and after an attack, rather than showing only a single command or isolated event.

### Why Wazuh with Sysmon:

Sysmon alone is a powerful log generator, but it cannot analyze or alert on events by itself. Wazuh provides the missing capabilities: log collection, processing, correlation, and alerting.

### Wazuh Agent

Collects logs from Sysmon

Sends logs securely to the Wazuh Manager.

Can apply local file integrity checks and basic analysis.

### Wazuh Manager

Receives logs from agents.

Decodes and normalizes Sysmon events.

Correlates events across endpoints to identify suspicious activity.

Generates alerts and visualizes attack timelines in the Wazuh Dashboard for SOC analysts.

## **Environment Overview:**

**Wazuh Agent:** Windows Server 2022 (monitors Sysmon logs)

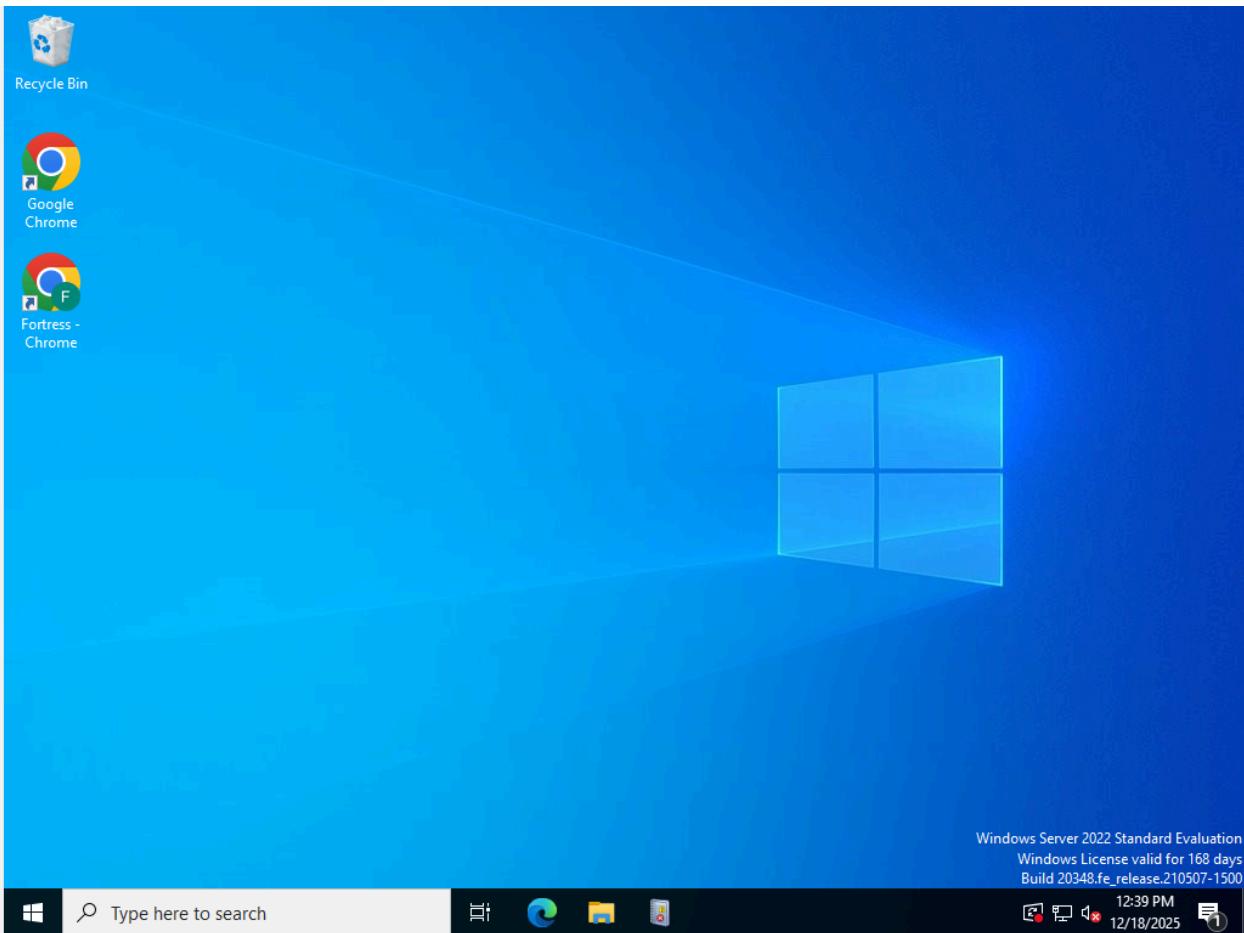
**Wazuh Manager + Indexer + Dashboard:** Ubuntu Desktop (central server, receives logs, visualizes alerts)

Agent collects Sysmon logs and sends them securely to the Manager.  
Manager decodes, correlates, and displays events in the Dashboard.

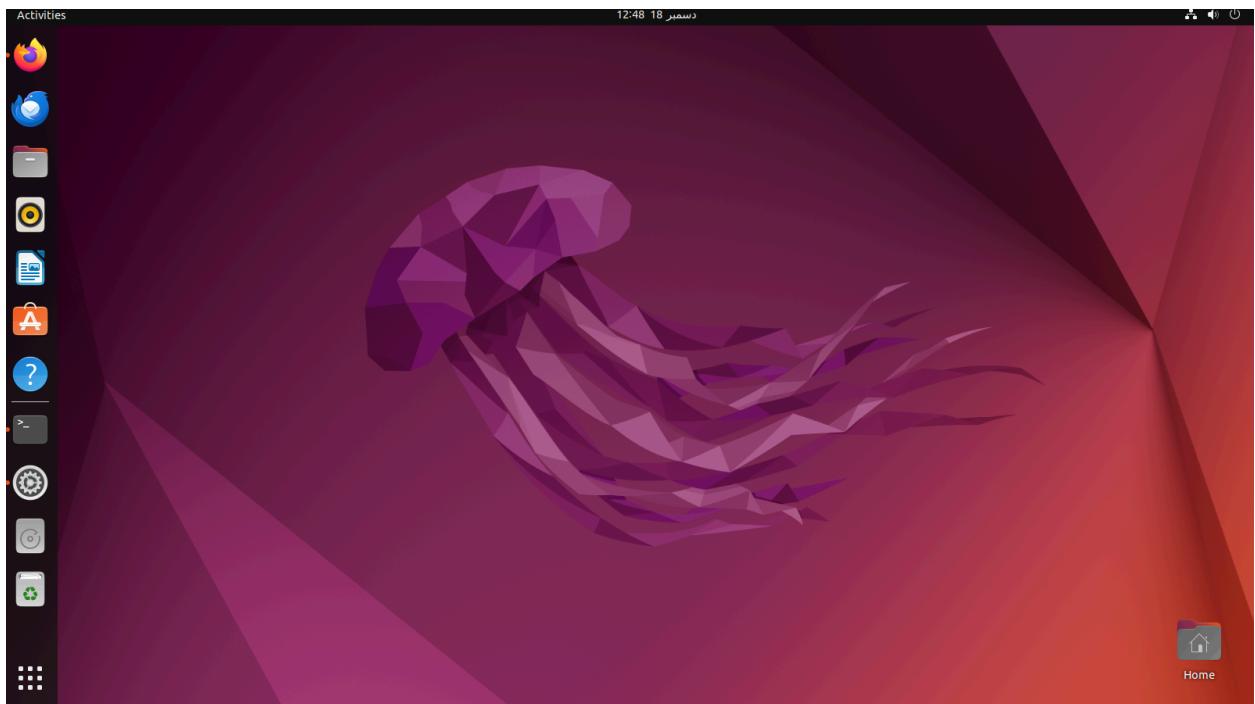
## **Step by Step Methodology:**

The first step is to ensure that the Wazuh Agent can communicate with the Wazuh Manager.

This is our Window Server where our Agent is installed.



And this is our Ubuntu Desktop based where our Wazuh Manager with other components is installed.



Use the ping command to verify network communication between the machines:

First Ping from Ubuntu Desktop to Windows Server is successful.

```
wazuh@fypserver:~$ ping 10.10.140.31
PING 10.10.140.31 (10.10.140.31) 56(84) bytes of data.
64 bytes from 10.10.140.31: icmp_seq=1 ttl=128 time=0.508 ms
64 bytes from 10.10.140.31: icmp_seq=2 ttl=128 time=0.631 ms
64 bytes from 10.10.140.31: icmp_seq=3 ttl=128 time=0.962 ms
^C
--- 10.10.140.31 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2072ms
rtt min/avg/max/mdev = 0.508/0.700/0.962/0.191 ms
wazuh@fypserver:~$
```

Ping from Windows Server to Ubuntu Desktop is successful.

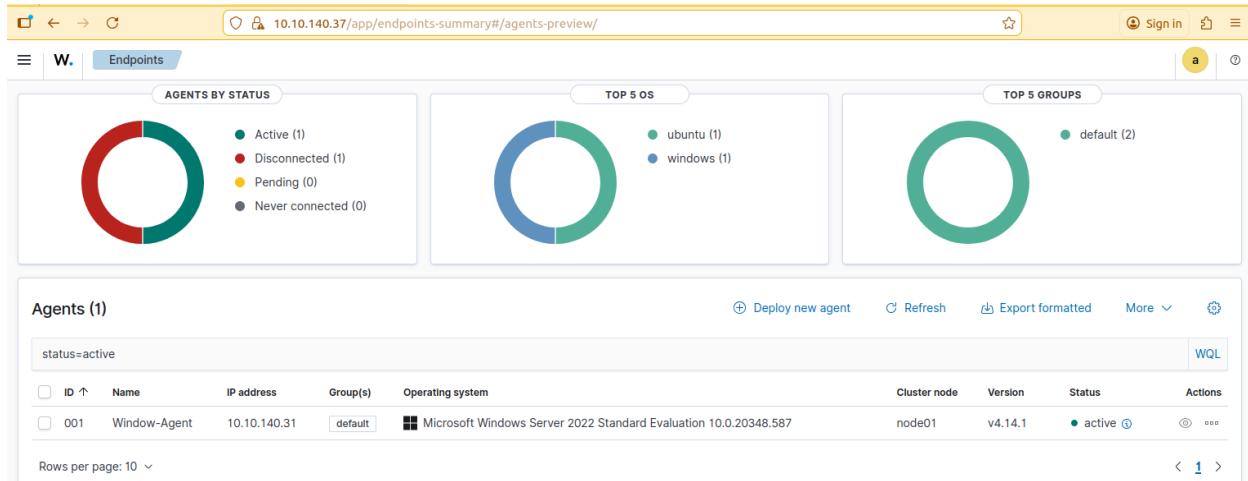
```
Administrator: Windows PowerShell
PS C:\Users\Administrator> ping 10.10.140.37

Pinging 10.10.140.37 with 32 bytes of data:
Reply from 10.10.140.37: bytes=32 time<1ms TTL=64

Ping statistics for 10.10.140.37:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
PS C:\Users\Administrator> ^C
```

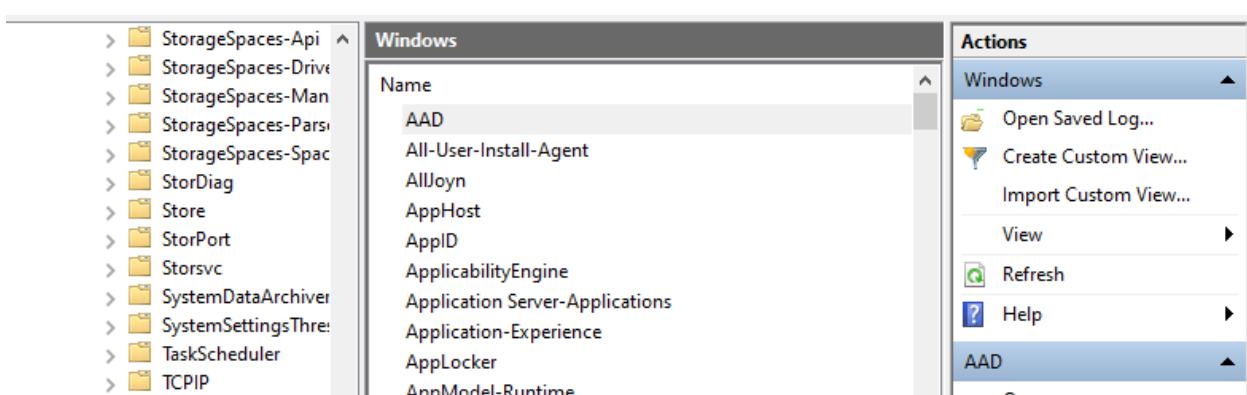
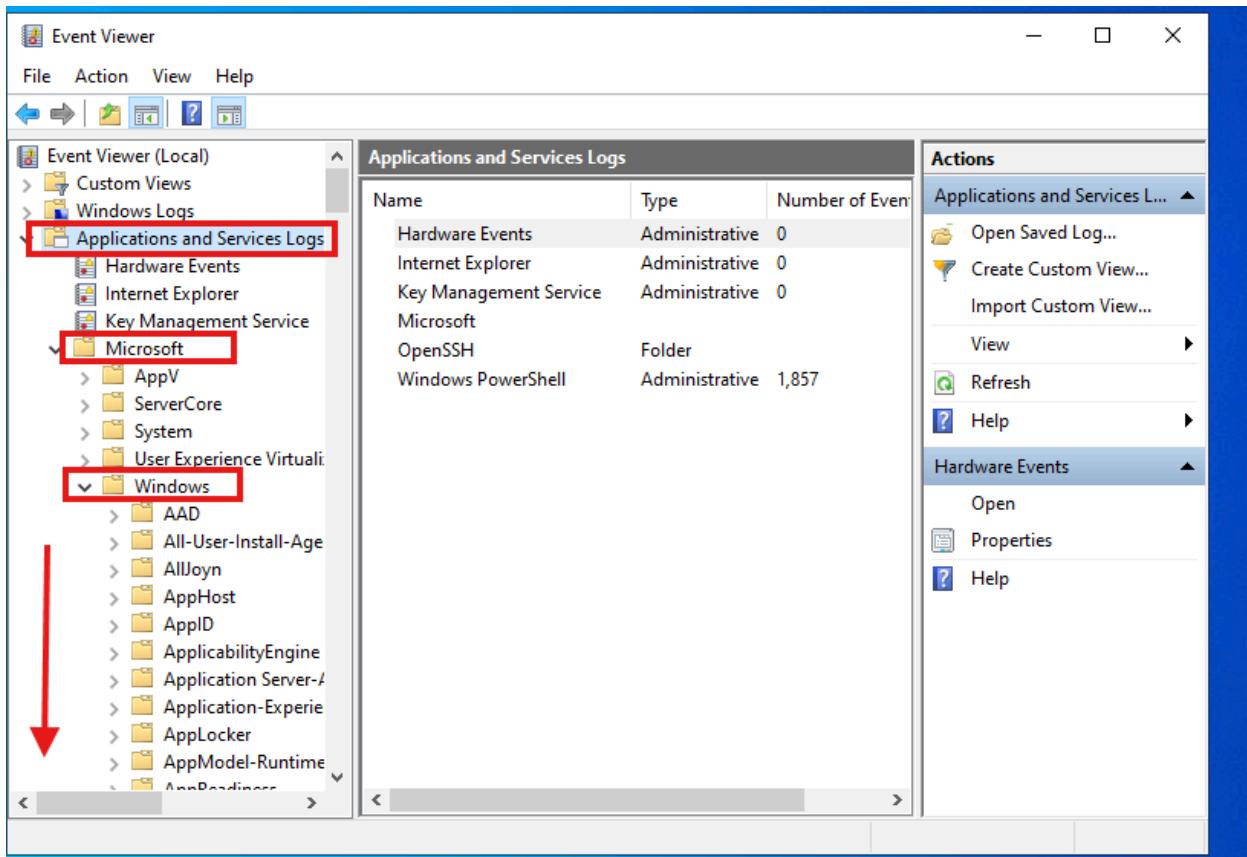
Both machines can communicate, confirming that the network connection between the Agent and Manager is working correctly.

And in the Dashboard for Agent status is Active:



### Step1:

Opened the Event Viewer. In the left panel, go to Application and Services Logs > Microsoft > Windows. Scroll down the list and check. And see that Sysmon is not present by default.



Sysmon is not installed here and so downloaded it from the following official Microsoft source:

<https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon>

[learn.microsoft.com/en-us/sysinternals/downloads/syomon](https://learn.microsoft.com/en-us/sysinternals/downloads/syomon)

Find by title

By Mark Russinovich and Thomas Garnier  
Published: July 23, 2024

 [Download Sysmon](#) (4.6 MB)

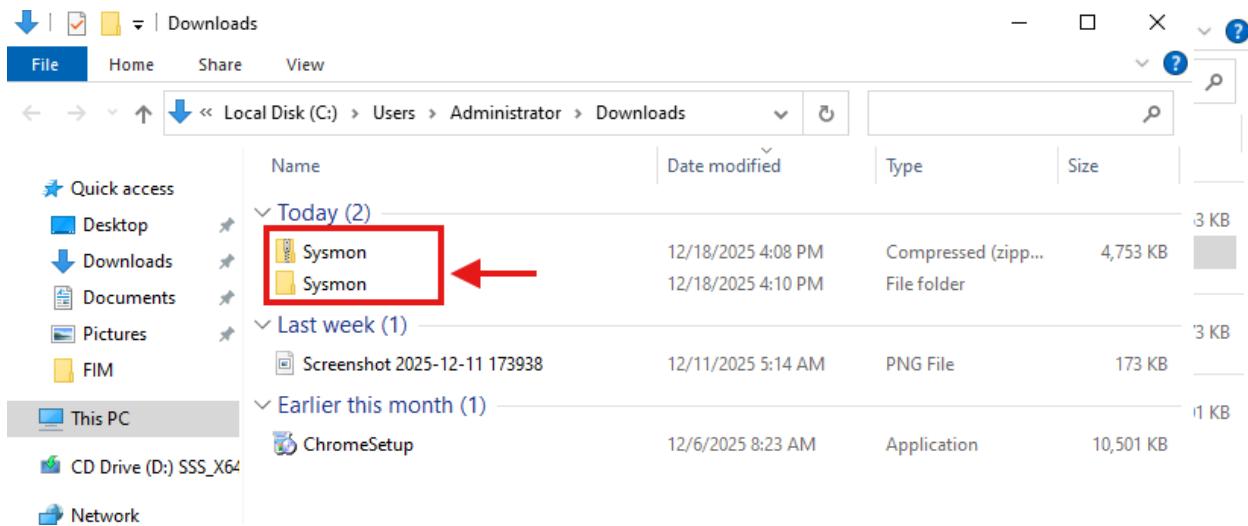
[Download Sysmon for Linux \(GitHub\)](#)

## Introduction

*System Monitor (Sysmon)* is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time. By collecting the events it generates using [Windows Event Collection](#) or [SIEM](#) agents and subsequently analyzing them, you can identify malicious or anomalous activity and understand how intruders and malware operate on your network. The service runs as a [protected process](#), thus disallowing a wide range of user mode interactions.

Note that *Sysmon* does not provide analysis of the events it generates, nor does it attempt to hide itself from attackers.

After downloading the Sysmon, extract the folder.



After downloading Sysmon, we also need to download its configuration file from GitHub. The configuration file helps Sysmon know what events to monitor and log.

We can get the official Sysmon configuration file from the following link:

<https://github.com/SwiftOnSecurity/sysmon-config>

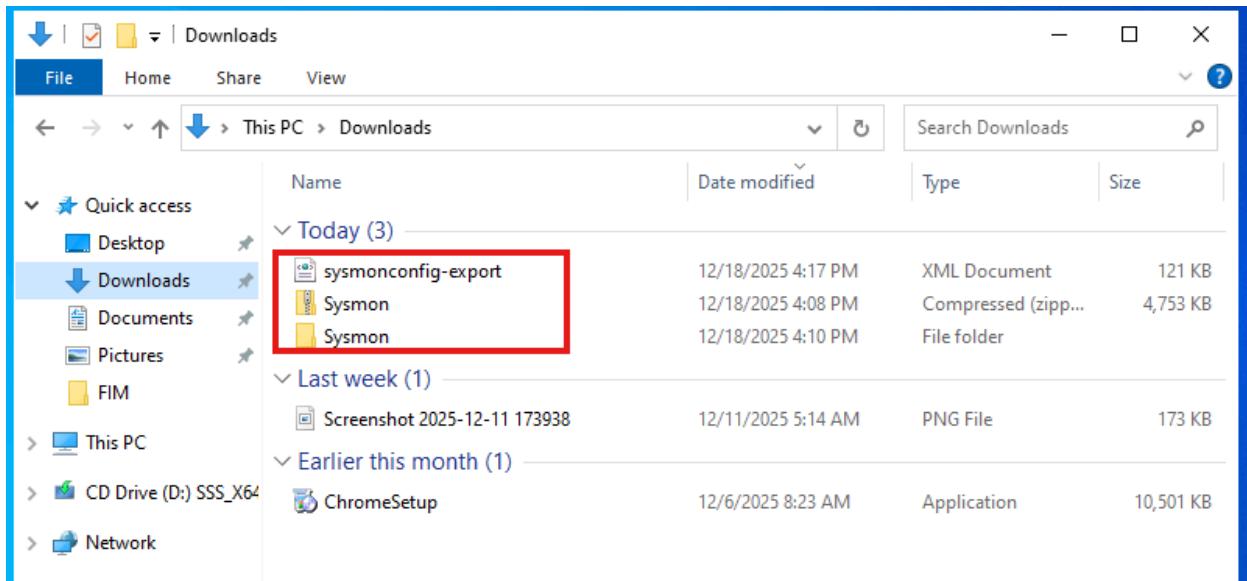
Now, download the “sysmonconfig-export.xml” file.

The screenshot shows a GitHub repository page for 'SwiftOnSecurity / sysmon-config'. The URL 'github.com/SwiftOnSecurity/sysmon-config' is highlighted in red in the address bar. The repository name 'SwiftOnSecurity / sysmon-config' is highlighted in red in the header. The file 'sysmonconfig-export.xml' is highlighted in red in the list of files.

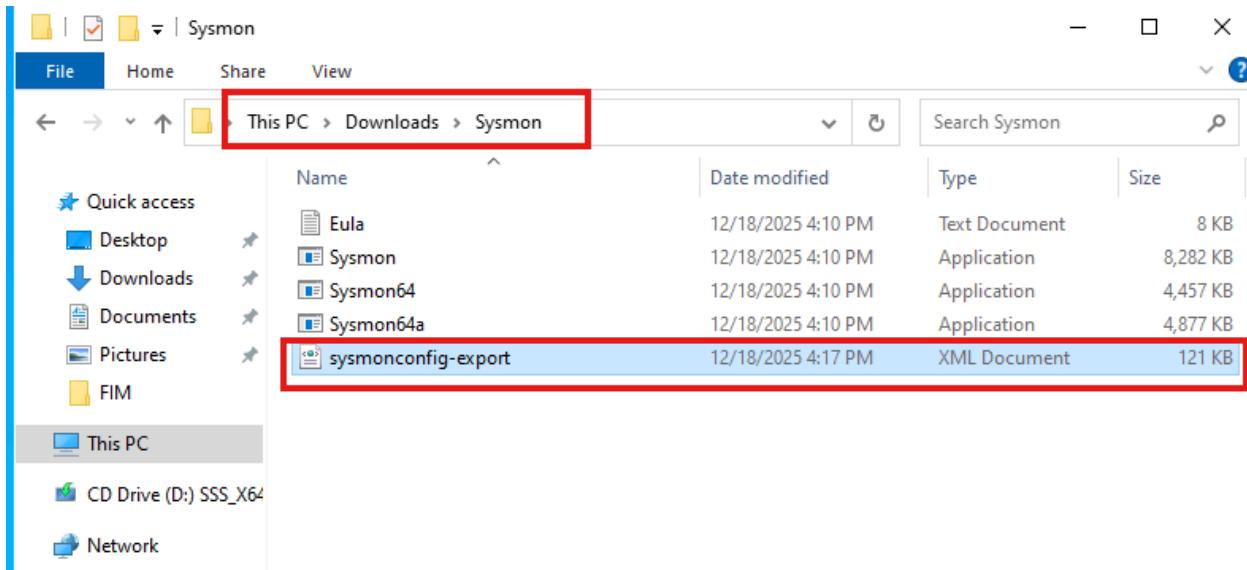
The screenshot shows the GitHub file page for 'sysmonconfig-export.xml' within the 'SwiftOnSecurity / sysmon-config' repository. The repository name 'SwiftOnSecurity / sysmon-config' is highlighted in red in the header. The file path 'sysmon-config / sysmonconfig-export.xml' is highlighted in red in the header. The file content is shown below, with the 'Raw' button highlighted in red.

```
1  <!--
2  sysmon-config | A Sysmon configuration focused on default high-quality event tracing and easy customization by the community
3  Source version: 74 | Date: 2021-07-08
4  Source project: https://github.com/SwiftOnSecurity/sysmon-config
5  Source license: Creative Commons Attribution 4.0 | You may privatize, fork, edit, teach, publish, or deploy for commercial us
6
7  Fork version: <N/A>
```

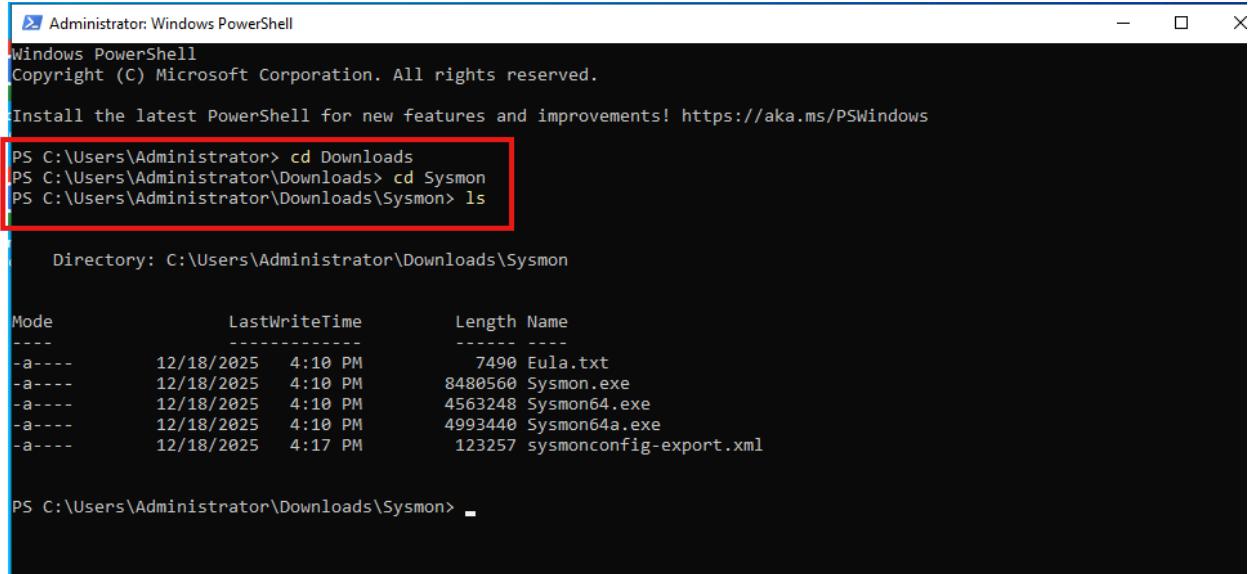
After this step, both Sysmon and the "sysmonconfig-export.xml" file will be downloaded successfully.



Now, paste the configuration file into the Sysmon folder.



Run PowerShell as Administrator in the Sysmon folder:



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Administrator> cd Downloads
PS C:\Users\Administrator\Downloads> cd Sysmon
PS C:\Users\Administrator\Downloads\Sysmon> ls

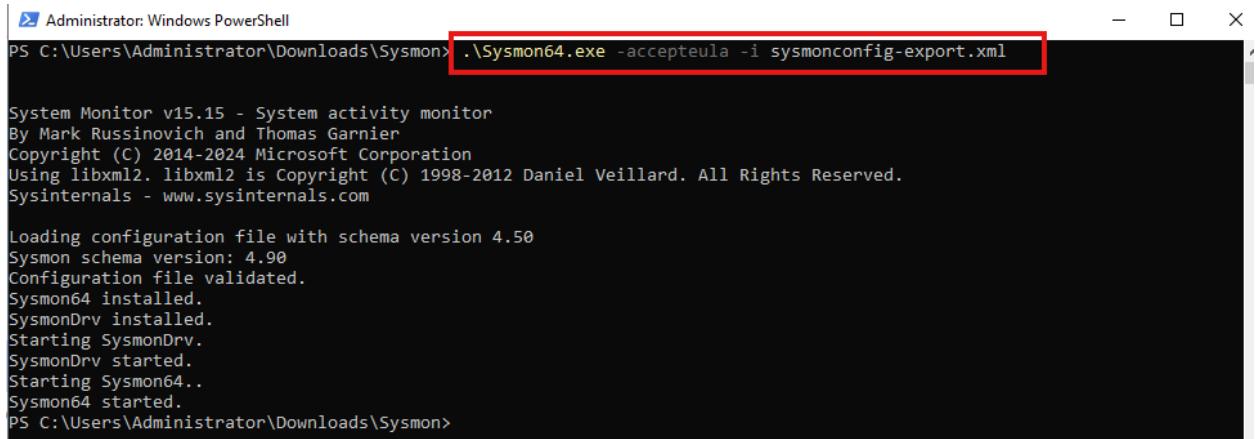
Directory: C:\Users\Administrator\Downloads\Sysmon

Mode                LastWriteTime         Length Name
----                -              -          -
-a----   12/18/2025  4:10 PM           7490 Eula.txt
-a----   12/18/2025  4:10 PM        8480560 Sysmon.exe
-a----   12/18/2025  4:10 PM        4563248 Sysmon64.exe
-a----   12/18/2025  4:10 PM        4993440 Sysmon64a.exe
-a----   12/18/2025  4:17 PM       123257 sysmonconfig-export.xml

PS C:\Users\Administrator\Downloads\Sysmon>
```

Now run the following command:

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



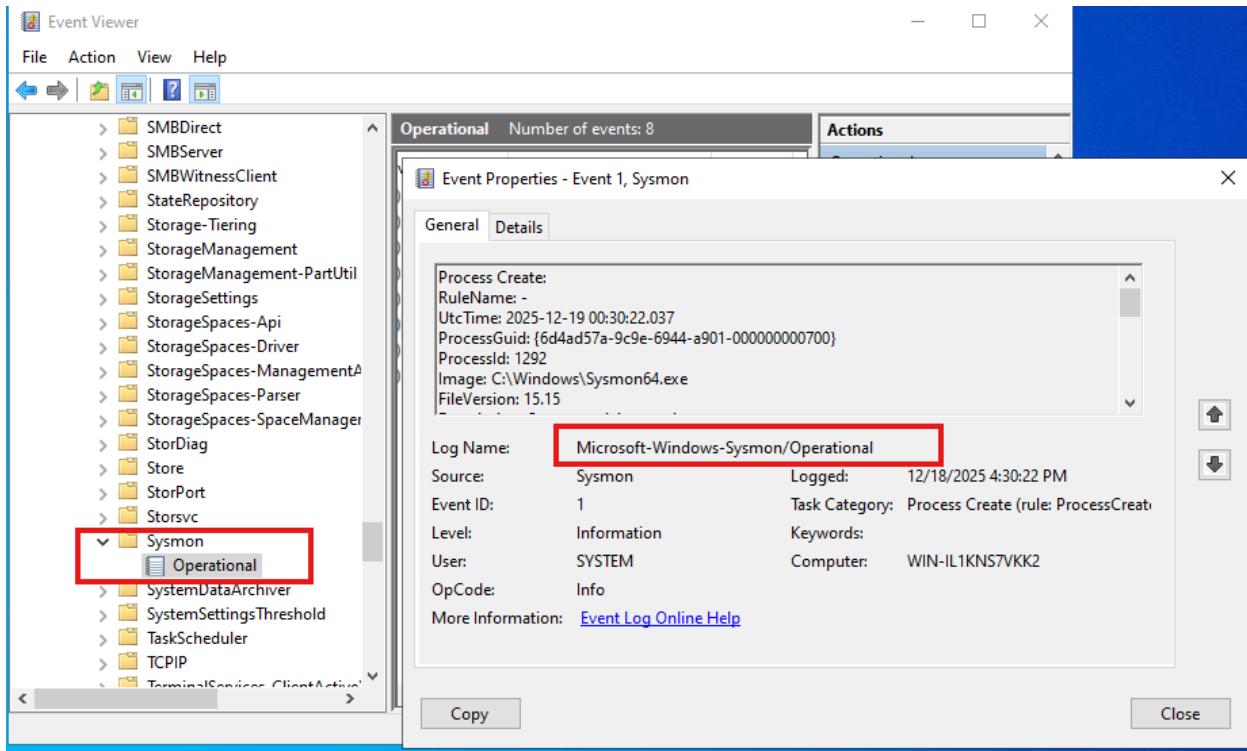
```
Administrator: Windows PowerShell
PS C:\Users\Administrator\Downloads\Sysmon> .\Sysmon64.exe -accepteula -i sysmonconfig-export.xml

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 C:\Users\Administrator\Downloads\Sysmon>
```

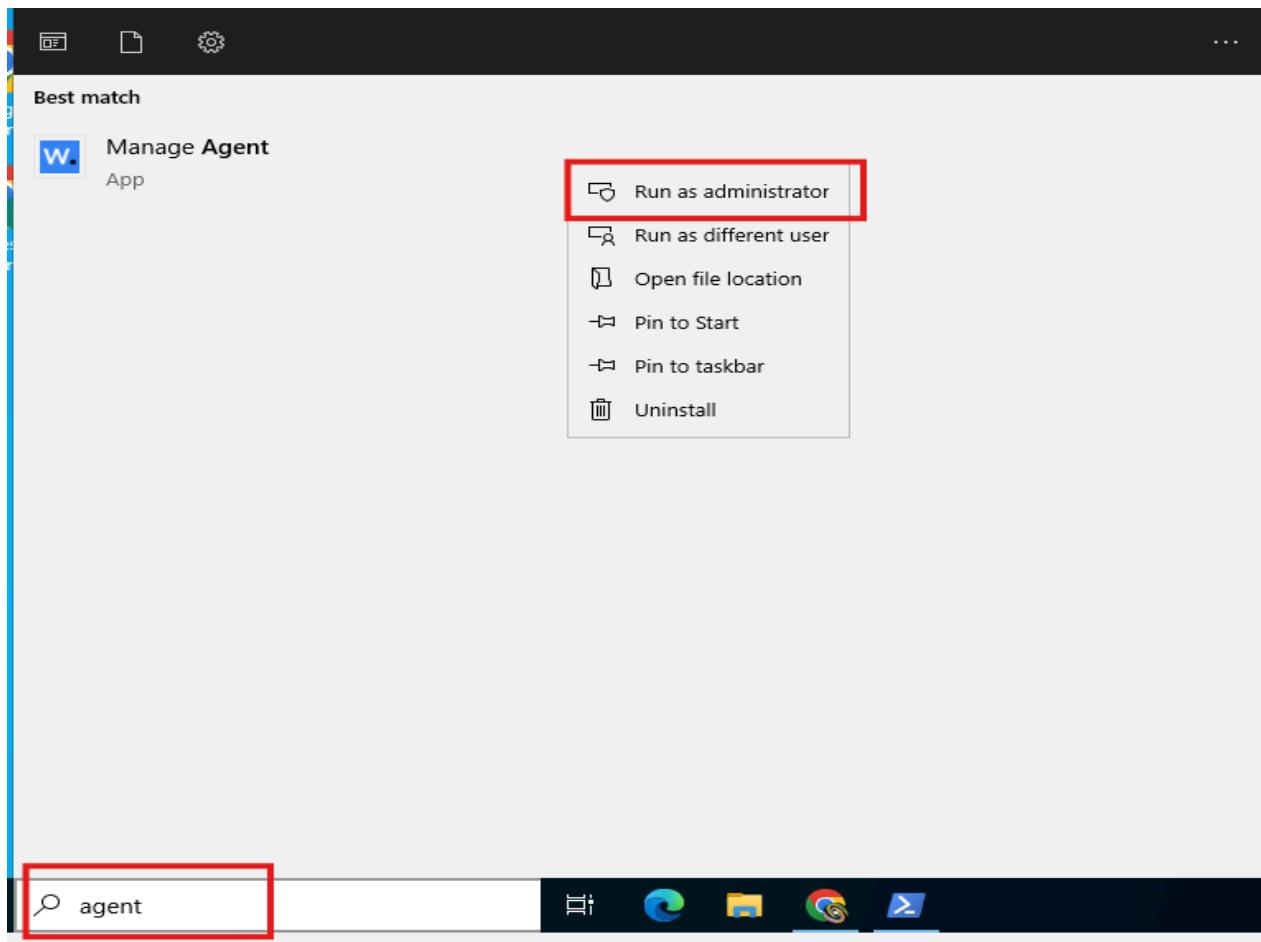
After the installation is complete, I need to verify it. Open Event Viewer and check if Sysmon logs are now visible under:

Application and Services Logs > Microsoft > Windows > Sysmon.

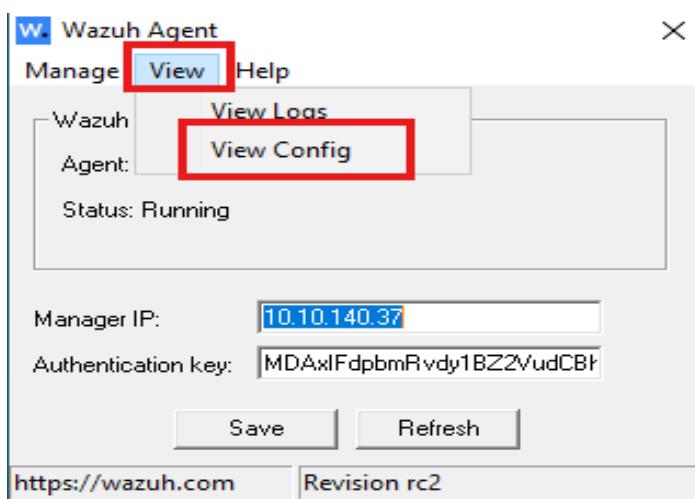


Now Sysmon is successfully installed and logs are appearing .

Now we need to forward Sysmon logs to Wazuh Manager. For this, open the **Wazuh agent** on your Windows Server. Search Agent in menu bar and "Run as administrator".



Next, open the "ossec.conf" file. We can do this by clicking on "View Config" in the Wazuh agent.



In the "ossec.conf" file, search for the "localfile" section.

After that, we have to specify the location of the Sysmon logs.

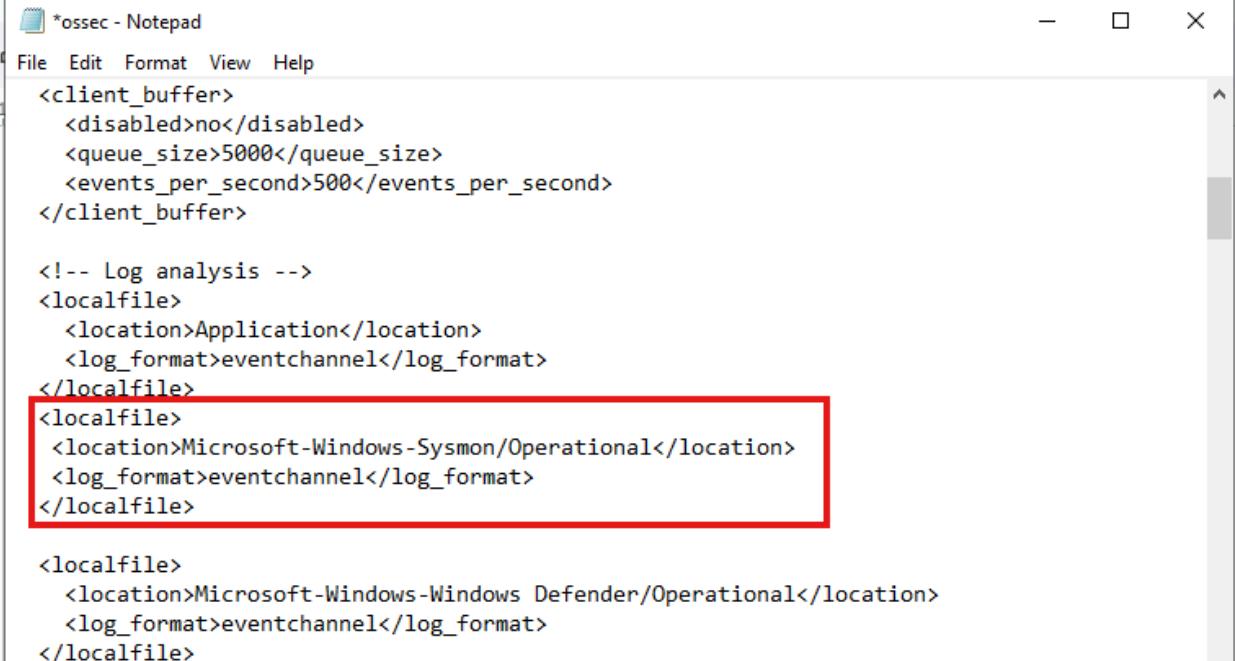
To configure Sysmon logs in the "ossec.conf" file, add the following lines:

```
<localfile>
```

```
  <location>Microsoft-Windows-Sysmon/Operational</location>
```

```
  <log_format>eventchannel</log_format>
```

```
</localfile>
```



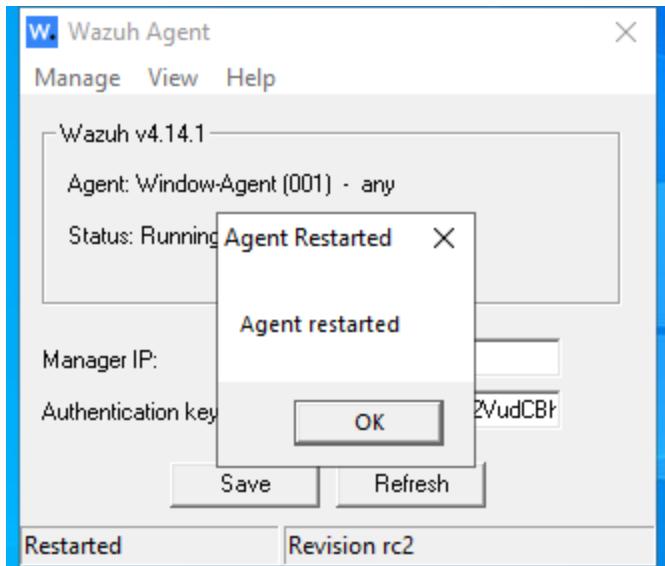
```
*ossec - Notepad
File Edit Format View Help
<client_buffer>
  <disabled>no</disabled>
  <queue_size>5000</queue_size>
  <events_per_second>500</events_per_second>
</client_buffer>

<!-- Log analysis -->
<localfile>
  <location>Application</location>
  <log_format>eventchannel</log_format>
</localfile>
<localfile>
  <location>Microsoft-Windows-Sysmon/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>

<localfile>
  <location>Microsoft-Windows-Defender/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>
```

Now, save the configuration file.

After saving, restart the Wazuh agent



## Setup on Wazuh Server

After configuring and restarting the Wazuh agent, go to the Wazuh server to download the Sysmon rules.

We need to download the Sysmon ruleset from Wazuh's official GitHub repository. These rules help Wazuh to properly analyze and generate alerts based on Sysmon logs.

We can download the Sysmon rules from the following link:

<https://github.com/wazuh/wazuh-ruleset/tree/master/rules>

Download or copy the file "win-sysmon\_rules.xml" from the Wazuh ruleset folder. This file contains the necessary rules for analyzing Sysmon logs.

The screenshot shows a GitHub repository page for 'wazuh/wazuh-ruleset'. The file '0595-win-sysmon\_rules.xml' is selected in the left sidebar under the 'win-sys' folder. The main content area displays the XML code for this rule. The code includes comments indicating it's a Windows Event Channel ruleset for the Sysmon channel, created by Wazuh Inc. in 2015-2020. It defines rules for informational and warning events based on specific Windows system severity values.

```

<!--
- Windows Event Channel ruleset for the Sysmon channel
- Created by Wazuh, Inc.
- Copyright (C) 2015-2020, Wazuh Inc.
- This program is a free software; you can redistribute it and/or modify it under the terms of GPLv2.
- ID range: 61600 - 62099
-->

<var name="MS_FREQ">8</var>
<group name="windows,syson,">
<rule id="61600" level="0">
<if_sid>60004</if_sid>
<field name="win.system.severityValue">^INFORMATION$</field>
<description>Windows Sysmon informational event</description>
<options>no_full_log</options>
</rule>
<rule id="61601" level="0">
<if_sid>60004</if_sid>
<field name="win.system.severityValue">^WARNING$</field>
<description>Windows Sysmon warning event</description>
<options>no_full_log</options>
<group>pgp13_4.12,</group>
</rule>
<rule id="61602" level="5">

```

Add these rules in following file:  
`/var/ossec/etc/rules/local_rules.xml`

```
wazuh@fypserver:~$ sudo nano /var/ossec/etc/rules/local_rules.xml
[sudo] password for wazuh:
wazuh@fypserver:~$ 
```

Currently, I have added only some specific rules in this document.

```
<group name="windows,syson,">
<rule id="61600" level="0">
<if_sid>60004</if_sid>
<field name="win.system.severityValue">^INFORMATION$</field>
<description>Windows Sysmon informational event</description>
<options>no_full_log</options>
</rule>
```

```
<rule id="61601" level="0">

<if_sid>60004</if_sid>

<field name="win.system.severityValue">^WARNING$</field>

<description>Windows Sysmon warning event</description>

<options>no_full_log</options>

<group>gpg13_4.12,</group>

</rule>

<rule id="61602" level="5">

<if_sid>60004</if_sid>

<field name="win.system.severityValue">^ERROR$</field>

<description>Windows Sysmon error event</description>

<options>no_full_log</options>

<group>system_error,gpg13_4.3,gdpr_IV_35.7.d,</group>

</rule>

<rule id="61603" level="0">

<if_sid>61600</if_sid>

<field name="win.system.eventID">^1$</field>

<description>Sysmon - Event 1: Process creation
$(win.eventdata.description)</description>

<options>no_full_log</options>

<group>sysmon_event1,</group>
```

```
</rule>

<rule id="61604" level="0">

    <if_sid>61600</if_sid>

    <field name="win.system.eventID">^2$</field>

        <description>Sysmon - Event 2: A process changed a file creation time by $(win.eventdata.sourceImage)</description>

    <options>no_full_log</options>

    <group>sysmon_event2,</group>

</rule>

<rule id="61605" level="0">

    <if_sid>61600</if_sid>

    <field name="win.system.eventID">^3$</field>

        <description>Sysmon - Event 3: Network connection by $(win.eventdata.sourceImage)</description>

    <options>no_full_log</options>

    <group>sysmon_event3,</group>

</rule>

<rule id="61606" level="0">

    <if_sid>61600</if_sid>

    <field name="win.system.eventID">^4$</field>

        <description>Sysmon - Event 4: Sysmon service state changed by $(win.eventdata.sourceImage)</description>
```

```
<options>no_full_log</options>

<group>sysmon_event4,</group>

</rule>

<rule id="61607" level="0">

<if_sid>61600</if_sid>

<field name="win.system.eventID">^5$</field>

    <description>Sysmon - Event 5: Process terminated by
$(win.eventdata.sourceImage)</description>

<options>no_full_log</options>

<group>sysmon_event5,</group>

</rule>

</group>
```

```

GNU nano 6.2                               /var/ossec/etc/rules/local_rules.xml *

group name="windows,sysmon,">
<rule id="61600" level="0">
<if_sid>60004</if_sid>
<field name="win.system.severityValue">^INFORMATION$</field>
<description>Windows Sysmon informational event</description>
<options>no_full_log</options>
</rule>

<rule id="61601" level="0">
<if_sid>60004</if_sid>
<field name="win.system.severityValue">^WARNING$</field>
<description>Windows Sysmon warning event</description>
<options>no_full_log</options>
<group>gpg13_4.12,</group>
</rule>

<rule id="61602" level="5">
<if_sid>60004</if_sid>
<field name="win.system.severityValue">^ERROR$</field>
<description>Windows Sysmon error event</description>
<options>no_full_log</options>
<group>system_error,gpg13_4.3,gdpr_IV_35.7.d,</group>
</rule>

<rule id="61603" level="0">
<if_sid>61600</if_sid>
<field name="win.system.eventID">^1$</field>
<description>Sysmon - Event 1: Process creation ${win.eventdata.description}</description>
<options>no_full_log</options>
<group>sysmon_event1,</group>
</rule>

<rule id="61604" level="0">
<if_sid>61600</if_sid>
<field name="win.system.eventID">^2$</field>
<description>Sysmon - Event 2: A process changed a file creation time by ${win.eventdata.sourceImage}</description>
<options>no_full_log</options>
<group>sysmon_event2,</group>
</rule>

```

Now saved the configuration file and restart Wazuh manager.

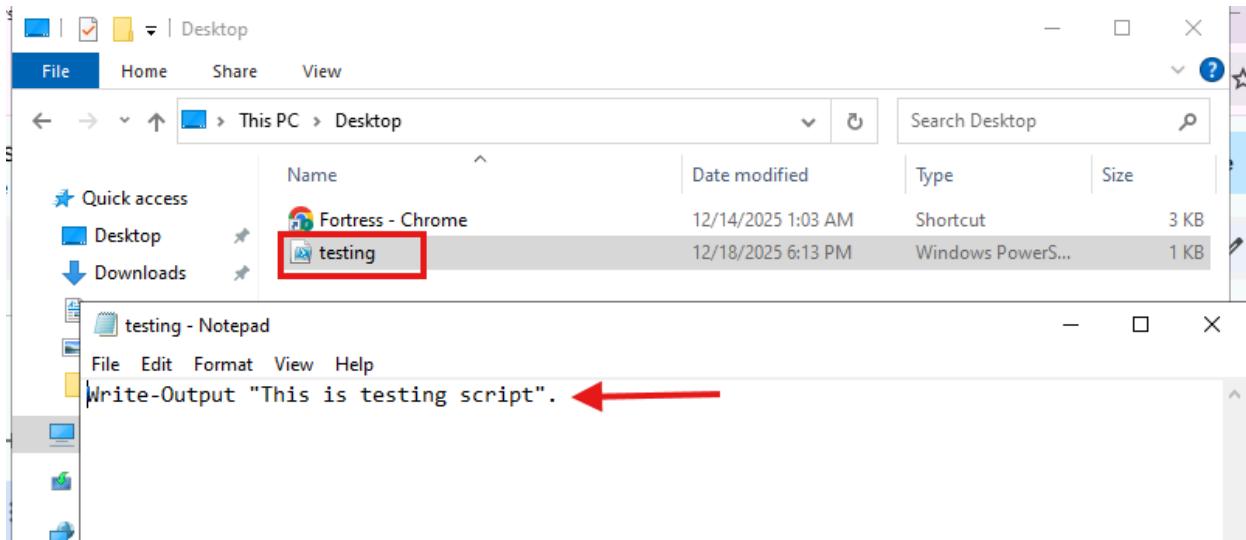
**Sudo systemctl restart wazuh-manager**

```
wazuh@fypserver:~$ sudo systemctl restart wazuh-manager
wazuh@fypserver:~$ 
```

## Testing Phase:

First I create the file in my Desktop folder in window server where my agent is installed.

**File Name:** testing.ps1



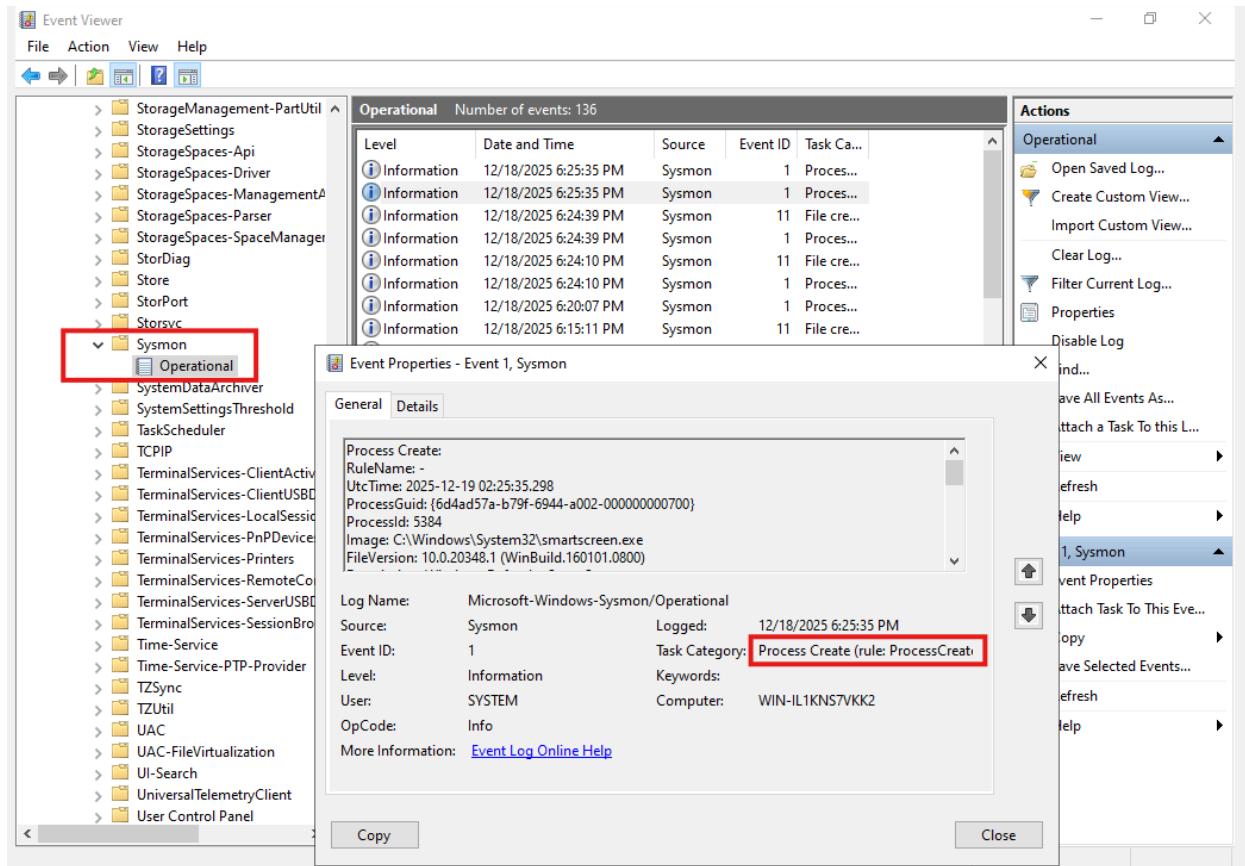
Run this command on powershell.

```
Powershell.exe -ExecutionPolicy Bypass -File "C:\Users\administrator\Desktop\testing.ps1"
```

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Powershell.exe -ExecutionPolicy Bypass -File C:\Users\administrator\Desktop\testing.ps1
This is testing script
.
PS C:\Users\Administrator>
```

Now check it in the event viewer for process creation.

Applications and Services Logs > Microsoft > Windows > Sysmon > Operational



Now we see these logs in the Wazuh Dashboard.

## Detailed Logs:

wazuh-alerts-4.x-2025.12.18#02IMZsBg047XD7fejPF	
	Table JSON
# @timestamp	Dec 18, 2025 @ 18:24:42.763
# _index	wazuh-alerts-4.x-2025.12.18
# agent.id	001
# agent.ip	10.10.140.31
# agent.name	Window-Agent
# data.win.eventdata.contextInfo	<pre>Severity = Informational Host Name = ConsoleHost Host Version = 5.1.20348.558 Host ID = 6e92f4cf-057d-4dad-85b3-b65d82a15841 Ho st Application = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe_ElevationPolicy_Bypass_File C:\Users\Administrator\Desktop\testing.ps1 Engine Version = 5.1.20348.558 Runspace ID = da71a6c5-c3d2-4029-baf7-6449855ec873 Pipeline ID = 1 Command Name = testing.ps1 Command Type = Ex ternalScript Script Name = Command Path = C:\Users\Administrator\Desktop\testing.ps1 Sequence Number = 18 User = WIN-IL1KNS7VKK2 \Administrator Connected User = Shell ID = Microsoft.PowerShell</pre>
# data.win.eventdata.payload	CommandInvocation(testing.ps1): \"testing.ps1\" CommandInvocation(Out-Default): \"Out-Default\" ParameterBinding(Out-Default): name=\"InputObject\"; value=\"This is test ing script\" ParameterBinding(Out-Default): name=\"InputObject\"; value=\".\"
# data.win.system.channel	Microsoft-Windows-PowerShell/Operational
# data.win.system.computer	WIN-IL1KNS7VKK2
# data.win.system.eventID	4183
# data.win.system.eventRecordID	1252
# data.win.system.keywords	0x8
# data.win.system.level	4
# data.win.system.message	> "CommandInvocation(testing.ps1): \"testing.ps1\" CommandInvocation(Out-Default): \"Out-Default\" ParameterBinding(Out-Default): name=\"InputObject\"; value=\"This is testing script\" ParameterBinding(Out-Default): name=\"InputObject\"; value=\".\""

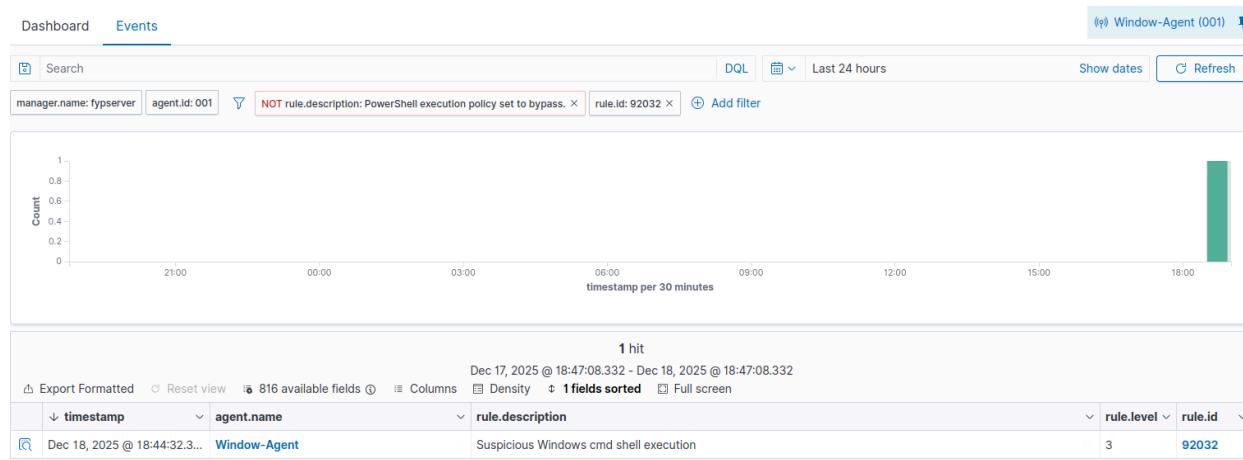
## Test 2:

cmd.exe /c whoami

Run this command on cmd:

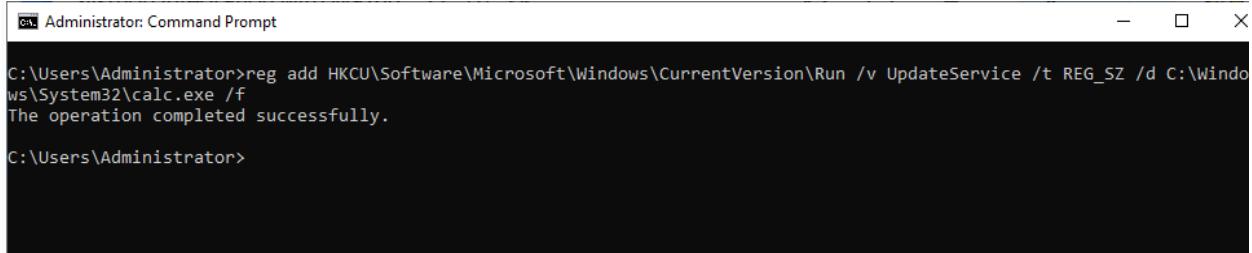


And its alerts appear in Dashboard.



### Test3: Registry changes detection alerts:

```
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v UpdateService /t REG_SZ /d C:\Windows\System32\calc.exe /f
```

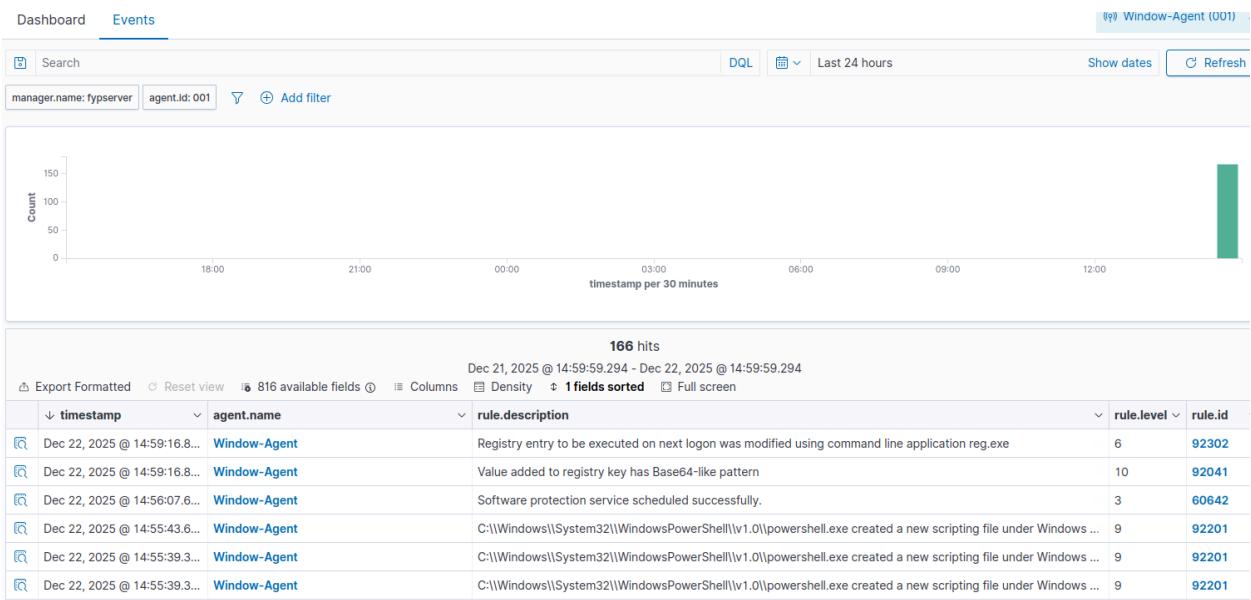


```
Administrator: Command Prompt

C:\Users\Administrator>reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v UpdateService /t REG_SZ /d C:\Windows\System32\calc.exe /f
The operation completed successfully.

C:\Users\Administrator>
```

Alerts appear in the Dashboard.



Details:

☰ W. Discover wazuh-alerts-4.x-2025.12.22#p41\_RZsBg047XD7f0EBr ⚙

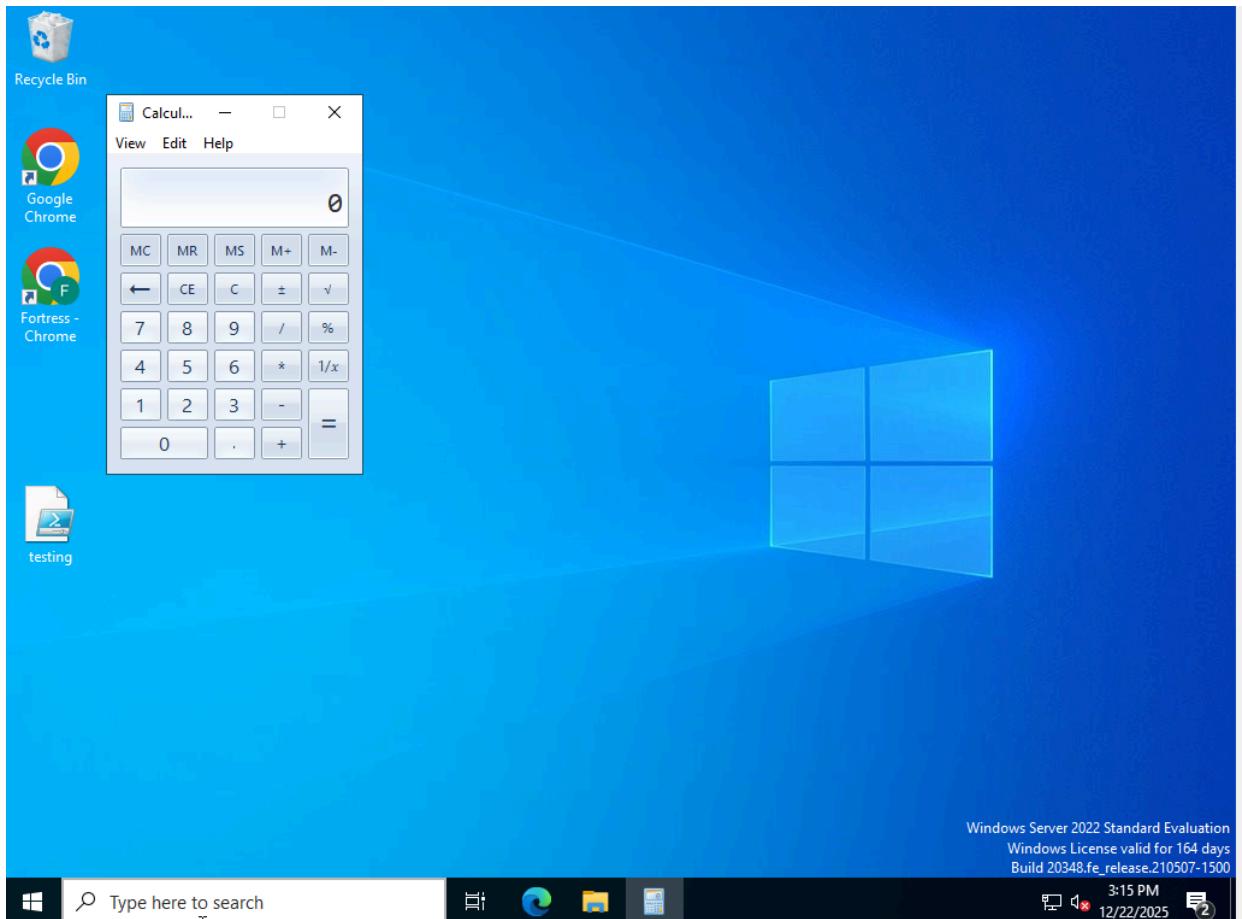
Table JSON

⌚ @timestamp	Dec 22, 2025 @ 14:59:16.892
t _index	wazuh-alerts-4.x-2025.12.22
t agent.id	001
t agent.ip	10.10.140.31
t agent.name	Window-Agent
⌚ data.win.eventdata.details	⚠ C:\Windows\System32\calc.exe
⌚ data.win.eventdata.eventtype	⚠ SetValue
t data.win.eventdata.image	C:\Windows\system32\reg.exe
t data.win.eventdata.processGuid	{6d4ad57a-cd42-6949-f301-000000000000}
t data.win.eventdata.processId	1136
⌚ data.win.eventdata.ruleName	⚠ T1060_RunKey
⌚ data.win.eventdata.targetObject	HKU\S-1-5-21-1290862803-2618783053-241933849-500\Software\Microsoft\Windows\CurrentVersion\Run\UpdateService
t data.win.eventdata.user	WIN-IL1KNS7VKK\Administrator
t data.win.eventdata.utcTime	2025-12-22 22:59:14.418
t data.win.system.channel	Microsoft-Windows-Sysmon/Operational
t data.win.system.computer	WIN-IL1KNS7VKK2
t data.win.system.eventID	13
t data.win.system.eventRecordID	913
t data.win.system.keywords	0x0000000000000000
t data.win.system.level	4
<hr/>	
t data.win.system.message	> "Registry value set: RuleName: T1060_RunKey EventType: SetValue UtcTime: 2025-12-22 22:59:14.418 ProcessGuid: {6d4ad57a-cd42-6949-f301-000000000000} ProcessId: 1136 Tmao - C:\Windows\custom2\run.eva
t data.win.system.opcode	0
t data.win.system.processID	2716
t data.win.system.providerGuid	{5770385f-c22a-43e0-bf4c-06f5698ffbd9}
t data.win.system.providerName	Microsoft-Windows-Sysmon
t data.win.system.severityValue	INFORMATION
t data.win.system.systemTime	2025-12-22T22:59:14.4269523Z
t data.win.system.task	13
t data.win.system.threadID	4960
t data.win.system.version	2
t decoder.name	windows_eventchannel
t id	1766397556.598567
t input.type	log
t location	EventChannel
t manager.name	fypserver
t rule.description	Registry entry to be executed on next logon was modified using command line application reg.exe
# rule.firedtimes	2
t rule.groups	sysmon, sysmon_eid13_detections, windows
t rule.id	92302
# rule.level	6

After adding the registry entry using the Command Prompt (CMD), the system behavior was tested.

During the next login to the Windows Server, the Calculator application opened automatically.

This confirms that the registry Run key was successfully modified and executed as expected during user login. .



### Summary:

We successfully connected Sysmon with Wazuh to watch what happens on a Windows system. Sysmon collects detailed actions like running programs, using commands, and changing the registry. The Wazuh Agent sends these logs to the Wazuh Manager, which checks them and shows alerts on the Dashboard.

During testing, running PowerShell scripts, CMD commands, and adding registry entries were all detected correctly. This shows that Sysmon with Wazuh can help see suspicious activity and keep the system secure.