



## **Integrating Sysmon With Wazuh**

**Created By: Amir Raza**

**Follow Me: [www.linkedin.com/in/amirsoc](https://www.linkedin.com/in/amirsoc)**

## **What is Sysmon?**

Sysmon, short for System Monitor, is a powerful system service and device driver from Microsoft that runs on Windows and provides detailed information about system activities. It is mainly used for security monitoring and incident detection because it records important events like process creations, network connections, file modifications, and more into the Windows Event Log. By doing this, Sysmon helps security analysts and system administrators to track suspicious or malicious behavior that could indicate a cyberattack or malware infection. Its logs are rich in detail, allowing organizations to have better visibility into what happens inside their systems, making it an essential tool in modern threat detection and digital forensics.

## **Integrating Sysmon with Wazuh**

When Sysmon is integrated with Wazuh, it becomes a very effective solution for advanced security monitoring. Sysmon collects detailed information about system activities such as process creation, registry changes, and network connections, while Wazuh acts as a security information and event management (SIEM) tool that analyzes and correlates these logs.

In this integration, Sysmon runs on the endpoint (like a Windows machine) and sends its event logs to the Wazuh agent installed on the same system. The Wazuh agent then forwards these logs to the Wazuh manager or server, where they are processed and analyzed. Wazuh uses its built-in decoders and rules to interpret Sysmon logs, detect potential threats, and generate alerts for suspicious activities. This setup allows organizations to identify malware behavior, lateral movement, and privilege escalation attempts quickly and efficiently.

Overall, the integration of Sysmon with Wazuh provides deep visibility into Windows systems and enhances an organization's ability to detect and respond to security incidents in real time.

## **Benefits Achieved After Integrating Sysmon with Wazuh**

### **Enhanced Threat Detection**

Detects suspicious process executions, file changes, and network activities in real time.

Identifies malware behavior, privilege escalations, and lateral movements.

### **Detailed System Visibility**

Provides deep insights into system activities like process creation, registry modifications, and driver loads.

Helps in understanding how attackers or malicious software operate within the system.

### **Centralized Log Management**

Sysmon logs are forwarded to the Wazuh server, allowing centralized storage and easier management of logs from multiple endpoints.

### **Automated Alerting and Response**

Wazuh automatically analyzes Sysmon logs using its built-in rules.

Generates real-time alerts when suspicious or abnormal behavior is detected.

### **Improved Incident Investigation and Forensics**

Detailed event logs help in tracing the full path of an attack or suspicious activity.

Simplifies root cause analysis during post-incident investigations.

### **Compliance and Reporting Support**

Helps organizations meet regulatory compliance requirements (like PCI DSS, HIPAA) by providing clear and structured system activity records.

## Reduced Attack Surface

Early detection of vulnerabilities or misconfigurations in the system that attackers could exploit.

## Resource Efficiency

Uses lightweight agents, making the integration suitable for large-scale deployment without heavily impacting system performance.

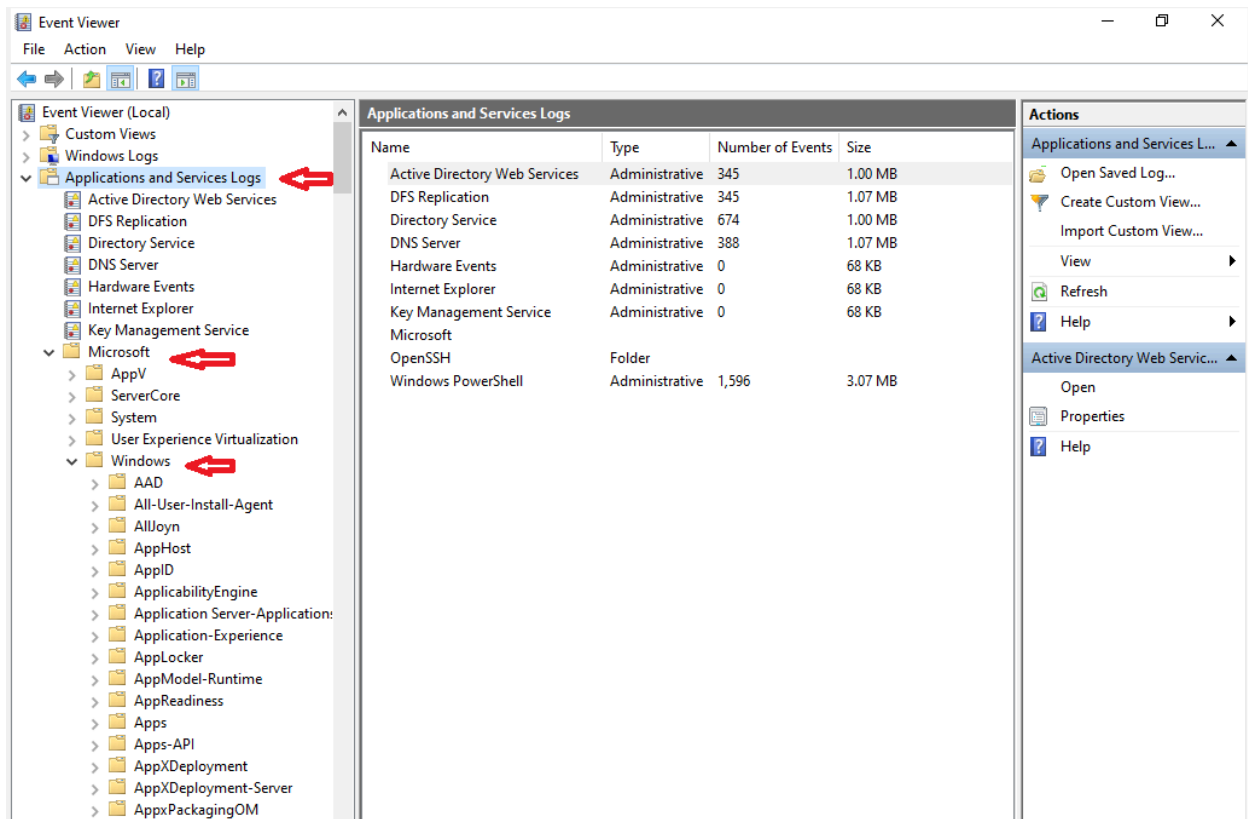
Open Event Viewer. In the left panel, go to **Application and Services Logs > Microsoft > Windows**. Scroll down the list and check — you will see that **Sysmon** is **not present by default**. However, since I have already installed it on my system, it is now available here.

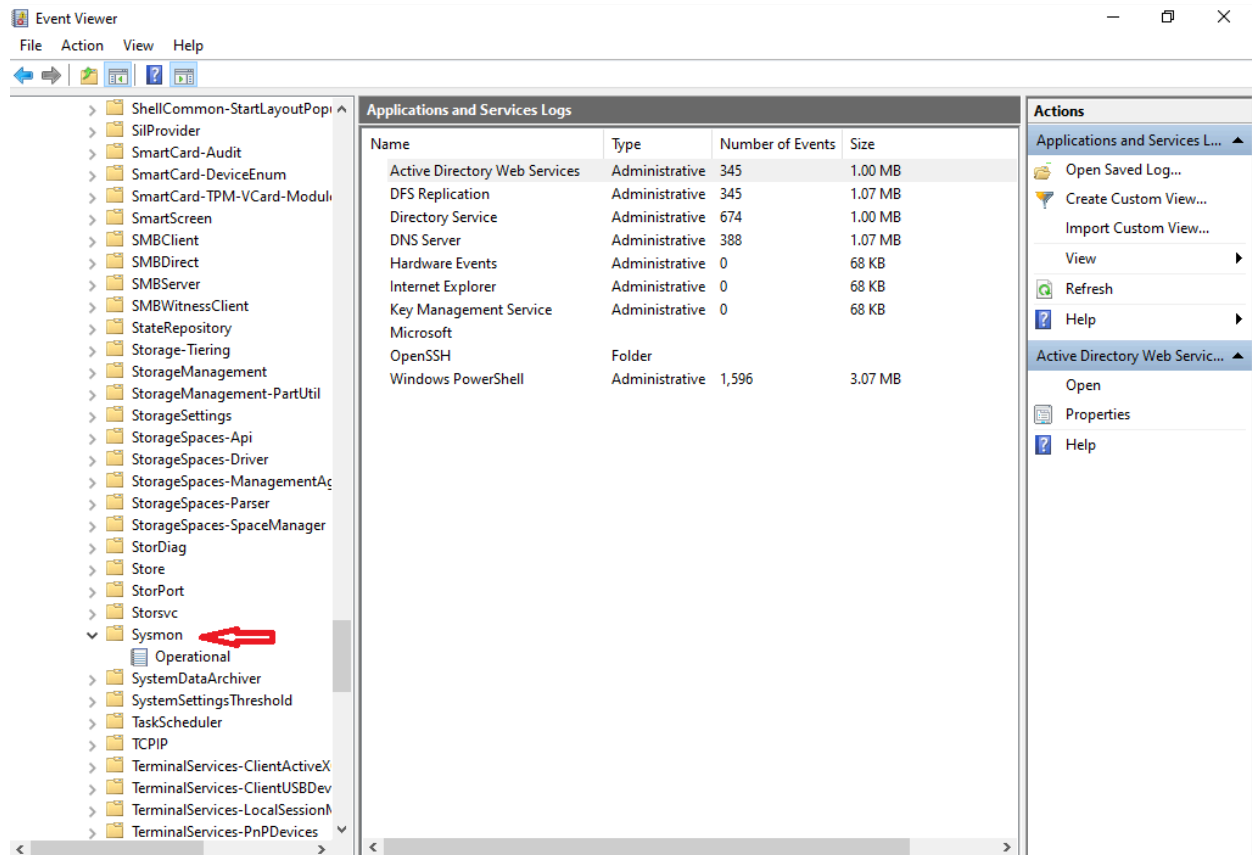
If you haven't installed Sysmon yet, you can download and install it from the following official Microsoft source:

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

The screenshot shows the Sysinternals website. The navigation menu on the left includes links to Home, Downloads, File and Disk Utilities, Networking Utilities, Process Utilities, Security Utilities, System Information, Miscellaneous, Sysinternals Suite, and Microsoft Store. The main content area displays the Sysmon v15.15 download page. The title 'Sysmon v15.15' is highlighted with a red box. Below the title, the authors 'Mark Russinovich and Thomas Garnier' and the publication date 'July 23, 2024' are listed. A red arrow points to the 'Download Sysmon (4.6 MB)' button. Below this, there is a link to 'Download Sysmon for Linux (GitHub)'. The page also includes an 'Introduction' section describing Sysmon as a Windows system service and device driver that monitors and logs system activity.

Now, if you open Event Viewer again, you will see that Sysmon has been added under the Windows logs section. This confirms that Sysmon has been successfully installed on your system.



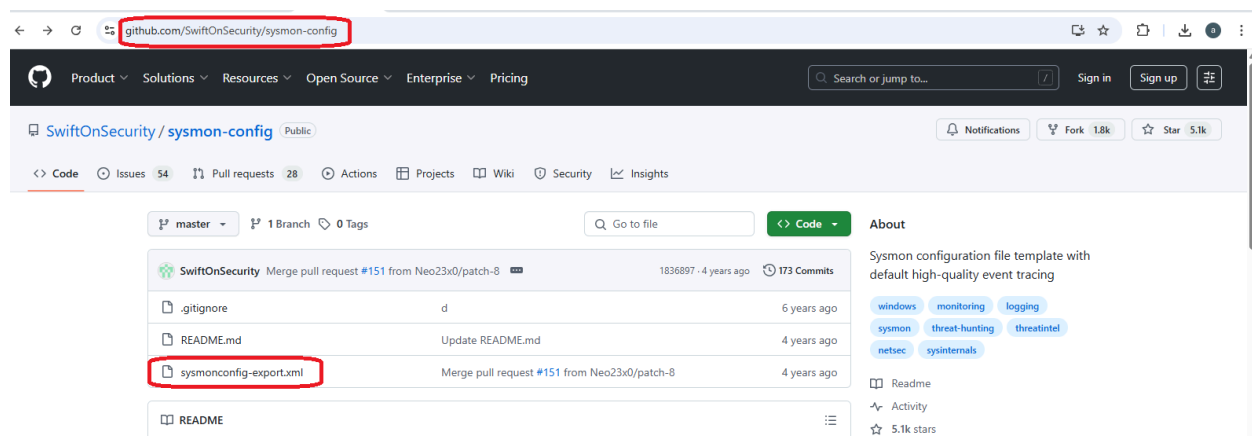


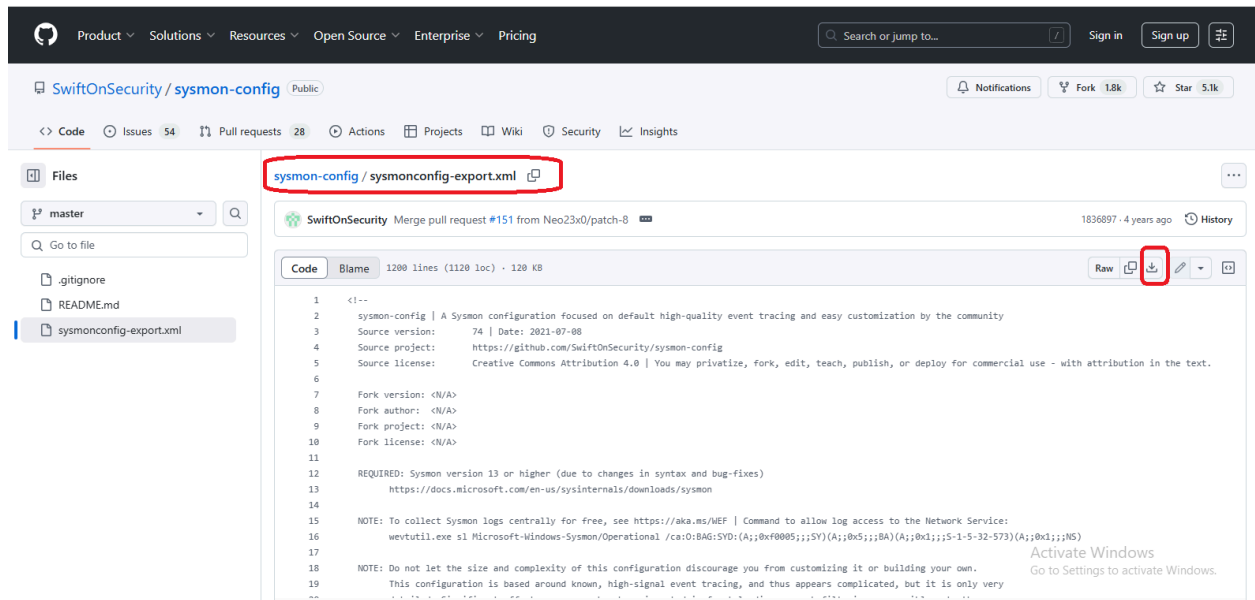
After downloading Sysmon, we also need to download its configuration file from GitHub. The configuration file helps Sysmon know what events to monitor and log.

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

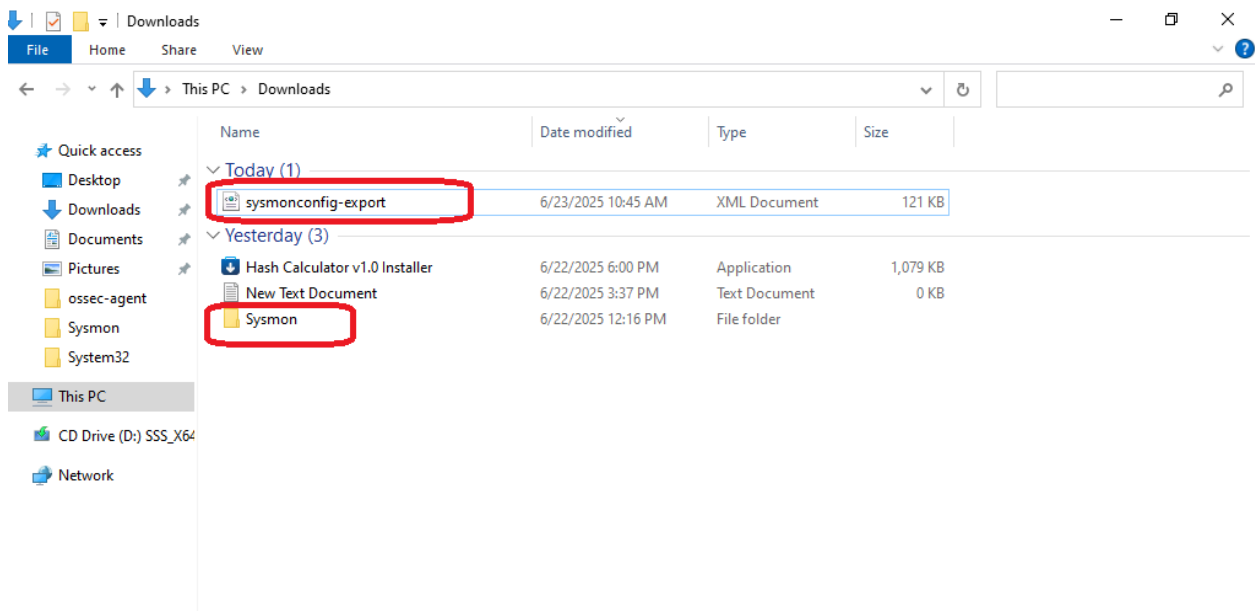
<https://github.com/SwiftOnSecurity/sysmon-config>

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

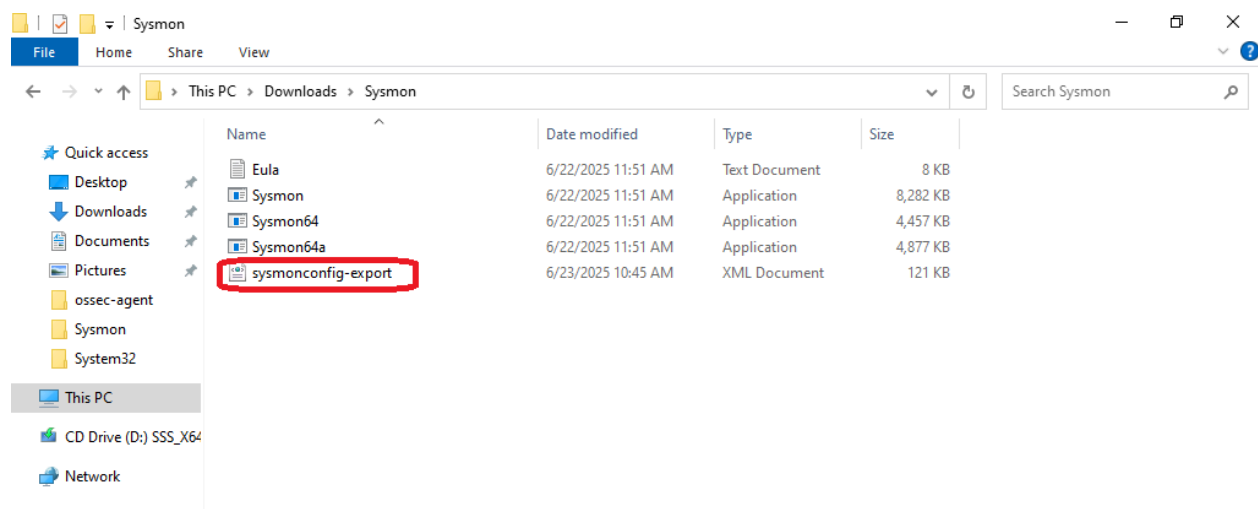




After this step, both Sysmon and the "sysmonconfig-export.xml" file will be downloaded successfully.



After extracting Sysmon, paste the configuration file into the Sysmon folder.



Now, locate the **Sysmon** folder and install Sysmon by following the steps shown in the figure.

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

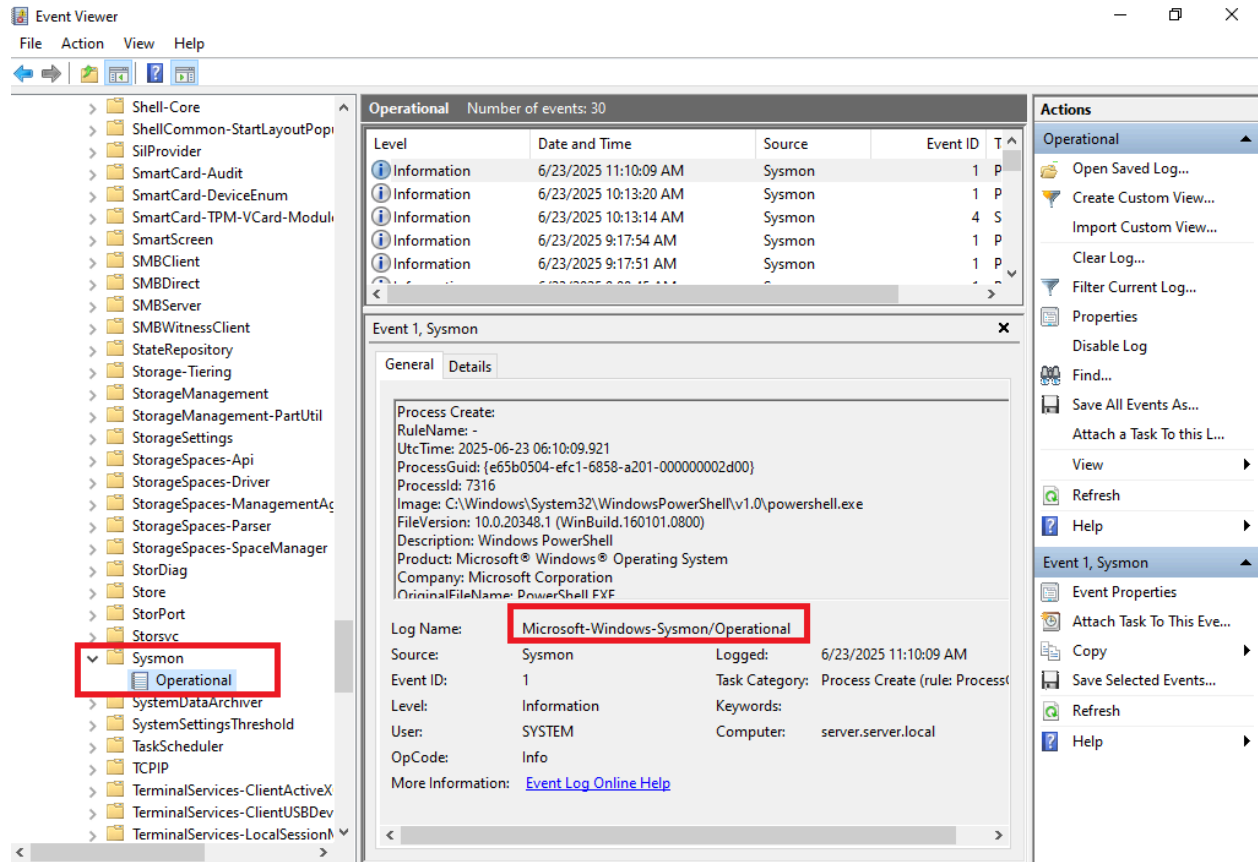
System Monitor v15.15 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.90
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, you need to verify it. Open Event Viewer and check if Sysmon logs are now visible under:

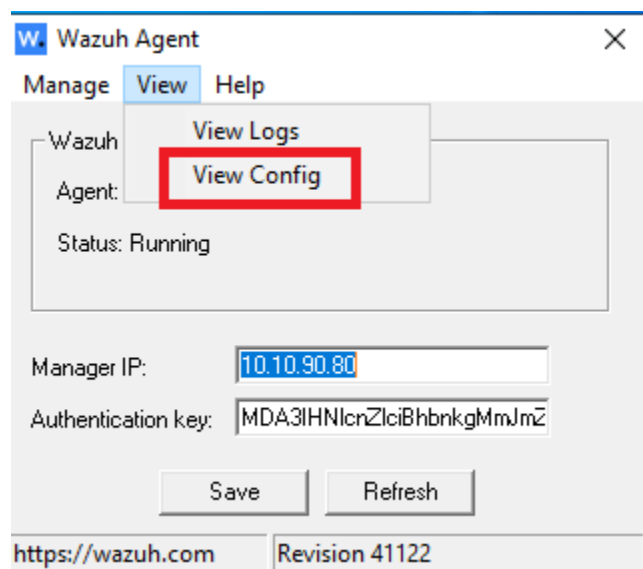
**Application and Services Logs > Microsoft > Windows > Sysmon.**





Now we need to forward Sysmon logs to Wazuh. For this, open the **Wazuh agent** on your Windows 11 system.

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



In the "**ossec.conf**" file, search for the "**localfile**" section as shown in the figure.

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

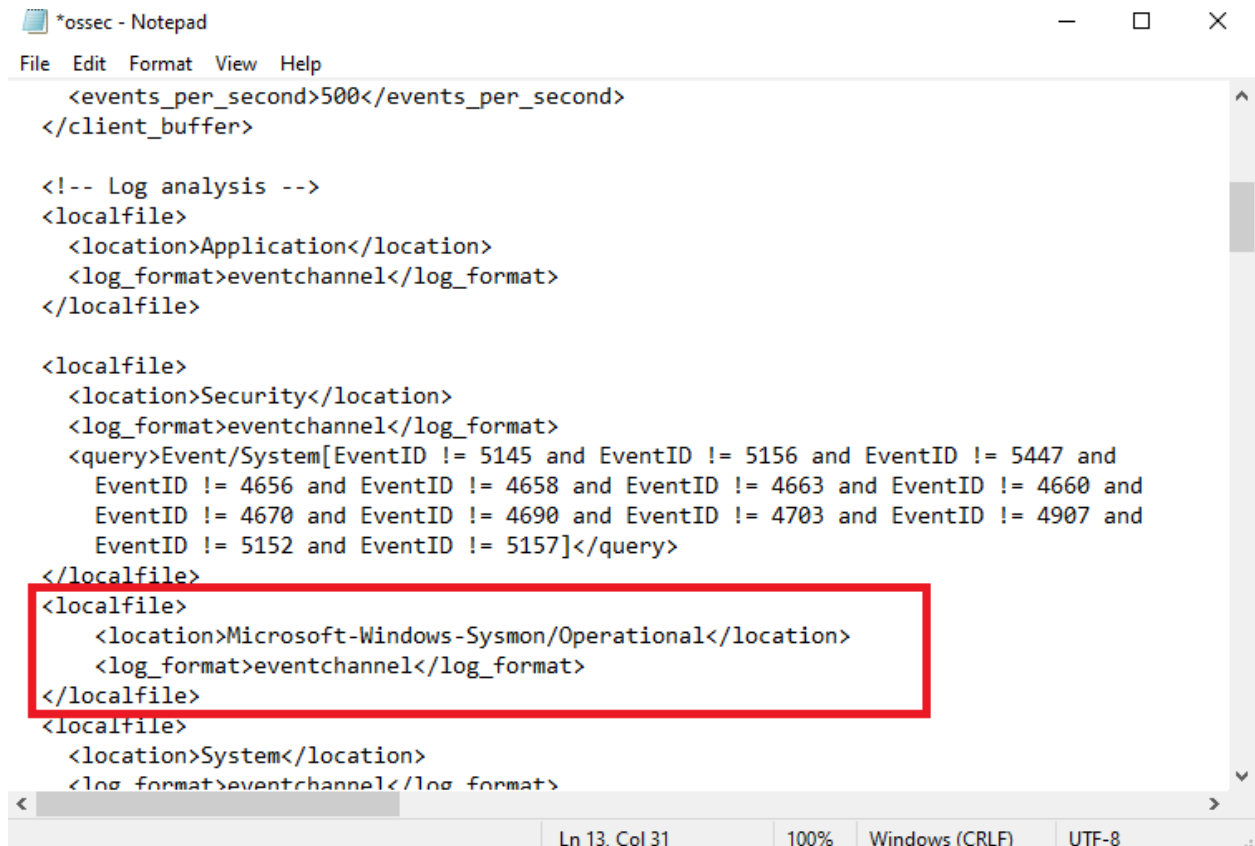
To configure Sysmon logs in the "**ossec.conf**" file, add the following lines:

```
<localfile>

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

  <log_format>eventchannel</log_format>

</localfile>
```



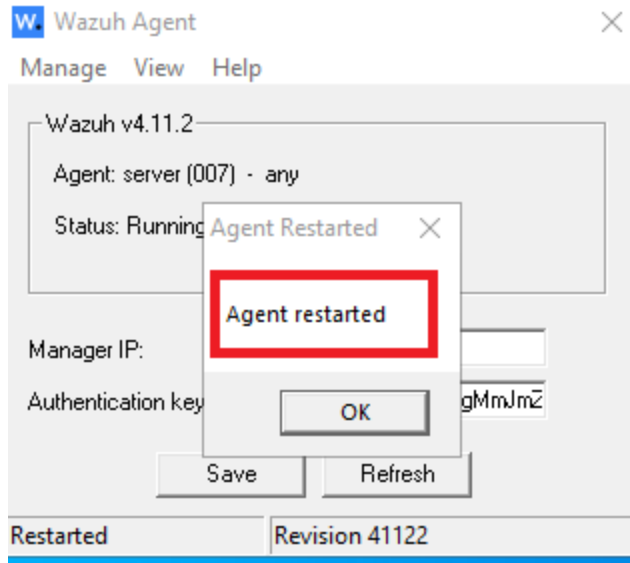
```
*ossec - Notepad
File Edit Format View Help
  <events_per_second>500</events_per_second>
</client_buffer>

<!-- Log analysis -->
<localfile>
  <location>Application</location>
  <log_format>eventchannel</log_format>
</localfile>

<localfile>
  <location>Security</location>
  <log_format>eventchannel</log_format>
  <query>Event/System[EventID != 5145 and EventID != 5156 and EventID != 5447 and
    EventID != 4656 and EventID != 4658 and EventID != 4663 and EventID != 4660 and
    EventID != 4670 and EventID != 4690 and EventID != 4703 and EventID != 4907 and
    EventID != 5152 and EventID != 5157]</query>
</localfile>
<localfile>
  <location>Microsoft-Windows-Sysmon/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>
<localfile>
  <location>System</location>
  <log_format>eventchannel</log_format>
```

Now, save the configuration file.

After saving, restart the Wazuh agent



## Setup on Wazuh Server

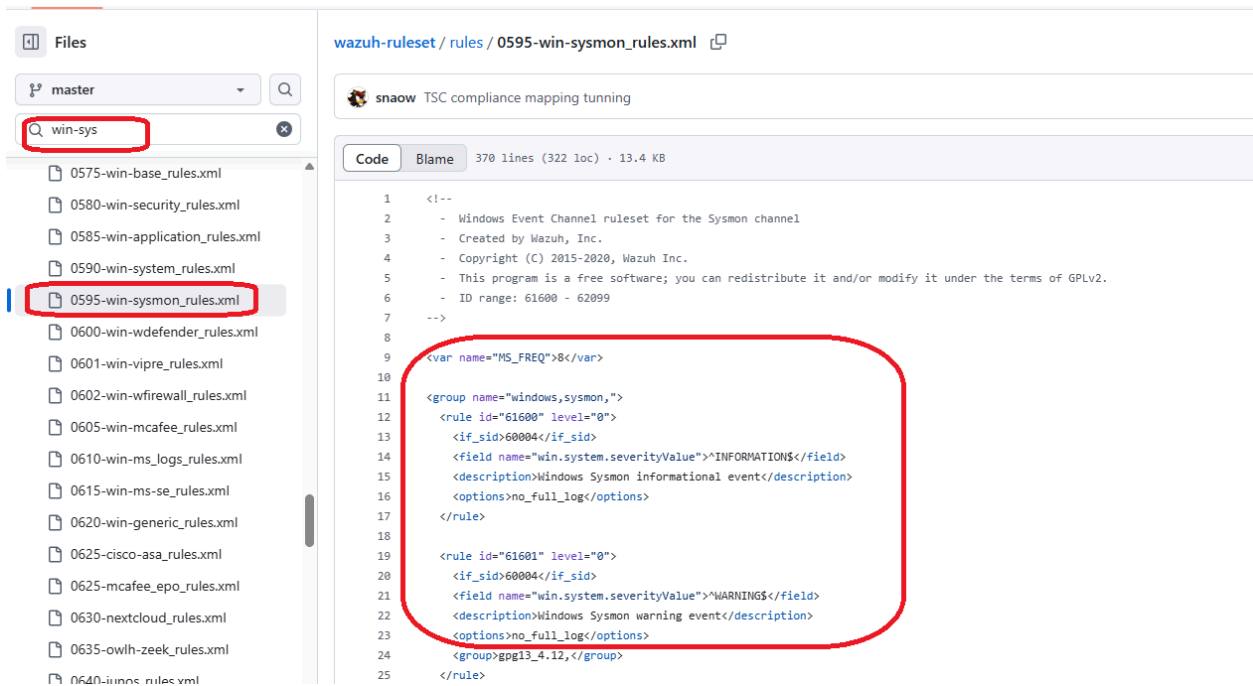
**After configuring and restarting the Wazuh agent,** go to the Wazuh server to download the Sysmon rules.

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.

You 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.



To add these rules in following file:  
/var/ossec/etc/rules/local\_rules.xml

Currently, I have added only some specific rules to this file, such as:

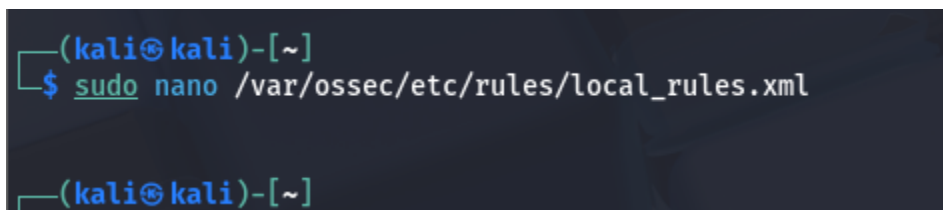
**Sid: 61650** — for *Event 22: DNS Query*

**Sid: 61603** — for *Event 1: Process creation*

**Sid: 61604** — for *Event 2: A process changed a file creation time*

**Sid: 61605** — for *Event 3: Network connection*

You can paste the full configuration file here if you want, but I am selecting rules based on the events I see in my Windows server Sysmon logs.



```

<group name="sysmon">
  <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>
  </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>
  </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>
  </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>
  </rule>
</group>

```

Now restart wazuh manager.

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

```

(kali㉿kali)-[~]
└─$ sudo systemctl restart wazuh-manager
[sudo] password for kali:

(kali㉿kali)-[~]
└─$

```

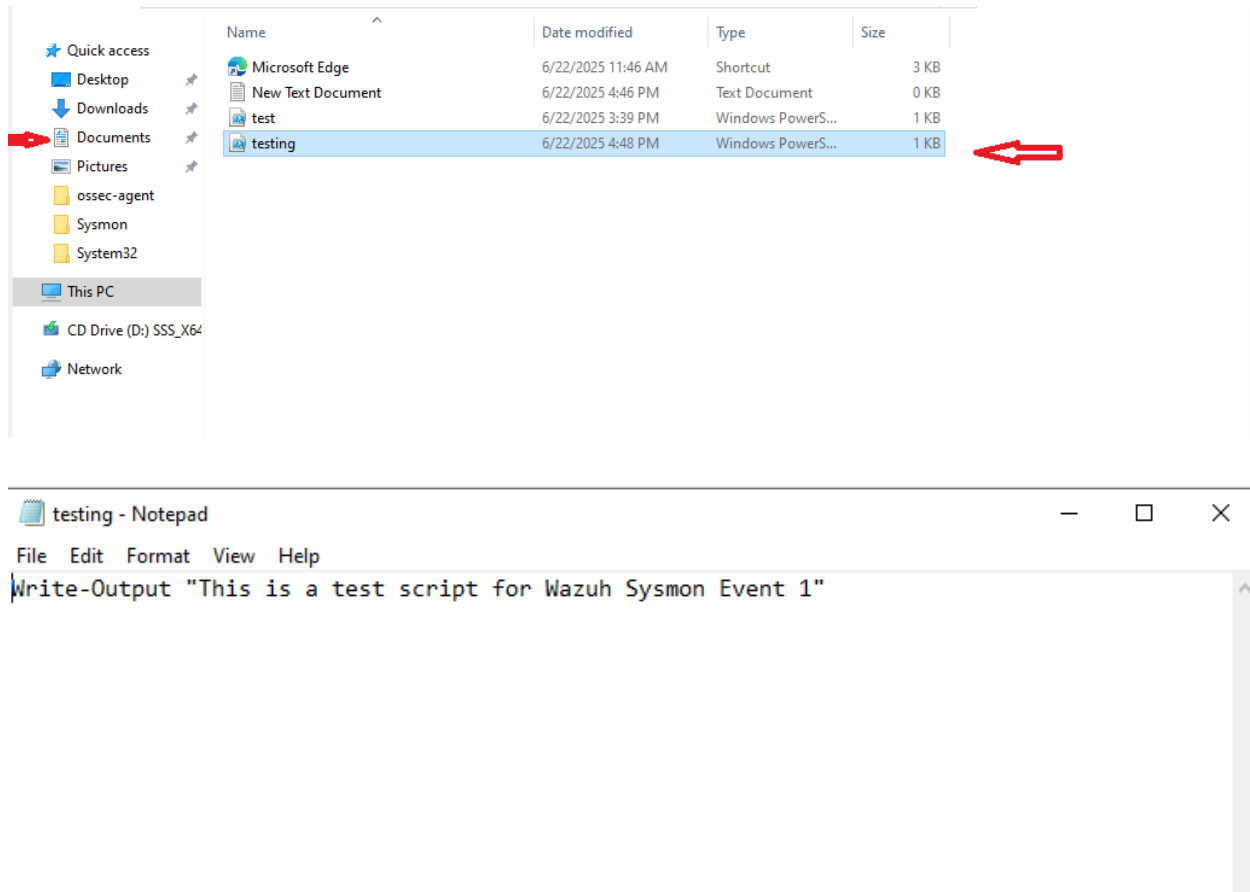
## Test Script Description

First i Create the file in my window server where my agent is installed.

**File Name:** testing.ps1

**Purpose:**

This PowerShell script has been created to generate Sysmon Event ID 1 specifically for testing the detection capabilities of Wazuh. The purpose of this script is to simulate the execution of a PowerShell process so that Wazuh rules related to Sysmon can be verified and validated.



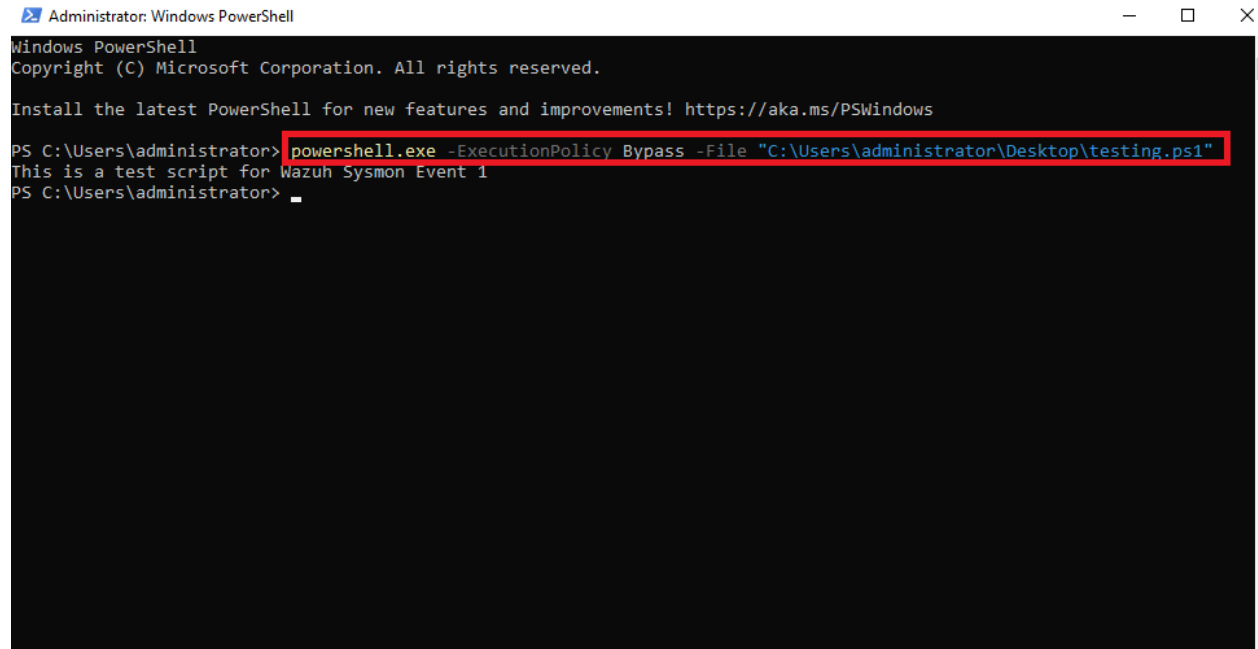
Command Used:

```
powershell.exe-ExecutionPolicBypas -File "C:\Users\administrator\Desktop\testing.ps1"
```

## Purpose of the Command

This PowerShell command is executed to run the `testing.ps1` script located on the Desktop of the system. The `-ExecutionPolicy Bypass` parameter temporarily disables the script execution policy to allow the script to run without any restrictions, even if stricter execution policies are enforced on the system. This command is mainly used for testing purposes to generate a Sysmon Event ID 1, which helps in verifying that the Wazuh monitoring system can detect and log the execution of PowerShell scripts. It ensures that the custom detection rules in Wazuh are functioning correctly by capturing and reporting this script execution event.

Run the powershell in administrative role and run given above command:



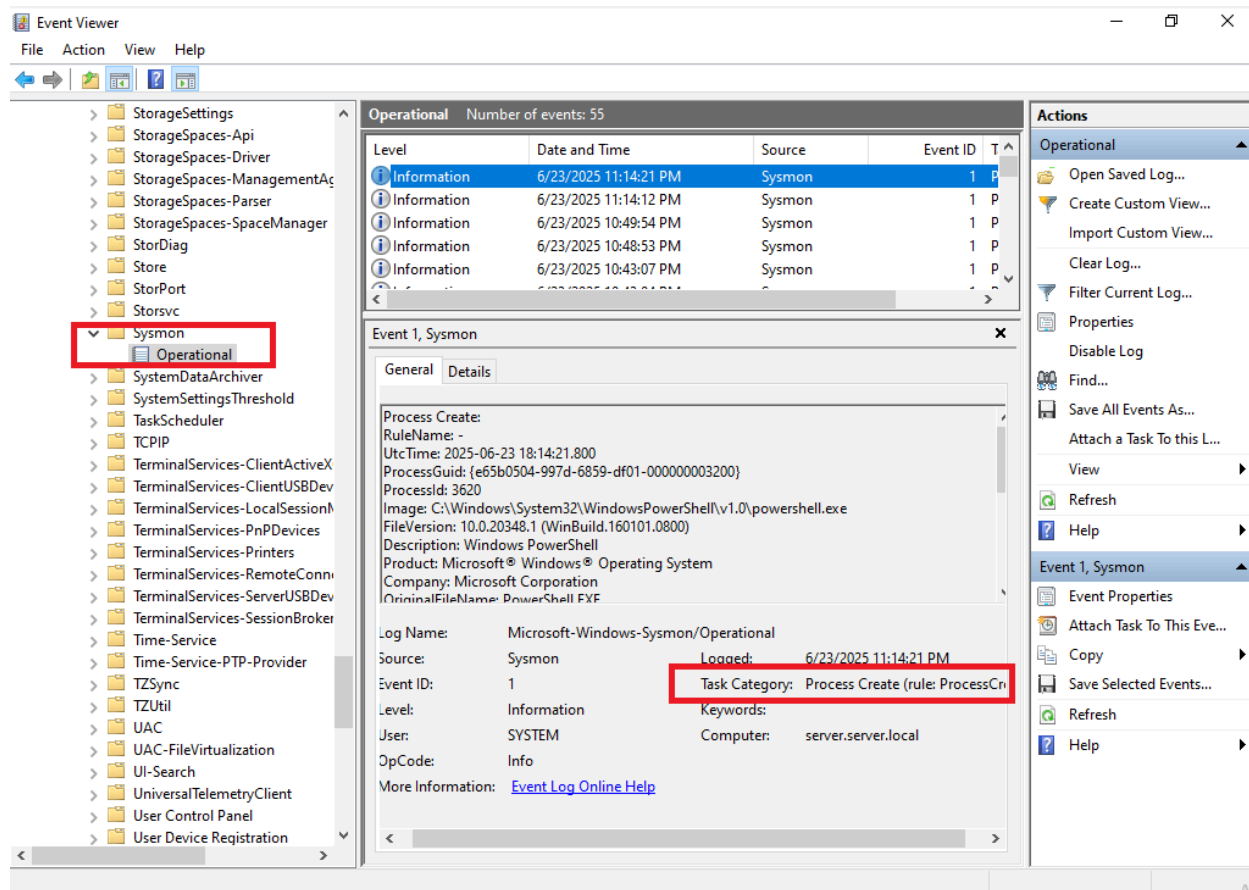
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\administrator> powershell.exe -ExecutionPolicy Bypass -File "C:\Users\administrator\Desktop\testing.ps1"
This is a test script for Wazuh Sysmon Event 1
PS C:\Users\administrator> _
```

Now check it in event viewer for process creation.

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



## Server-Side Log Verification

To verify whether the event generated by the Windows Agent reached the Wazuh Server, the following command was executed on the Wazuh Server terminal:

```
sudo tail -f /var/ossec/logs/alerts/alerts.json | grep PowerShell
```

This command continuously monitors the Wazuh alerts log file (`alerts.json`) in real-time and filters the output to display only those logs that contain the keyword "PowerShell." This ensures that the PowerShell execution event from the agent was successfully received and processed by the Wazuh Server.



```
(kali@kali)-[~]
└─$ sudo tail -f /var/ossec/logs/alerts/alerts.json | grep PowerShell

{"timestamp":"2025-06-23T13:41:00.541-0400","rule":{"level":6,"description":"PowerShell executed script from suspicious location","id":"92029","mitre":{"id":["T1059.001"],"tactic":["Execution"],"technique":["PowerShell"]},"firedtimes":1,"mail":false,"groups":{"sysmon","sysmon_eid1_detections","windows"},"agent":{"id":"007","name":"server","ip":"10.0.2.15"},"manager":{"name":"kali"},"id":"1750700400.17149248","decoder":{"name":"windows_eventchannel"},"data":{"win":{"system":{"providerName":"Microsoft-Windows-Sysmon","providerGuid":{"5770385f-c22a-43e0-bf4c-06f5698ff6b9"},"eventID":"1","version":"5","level":"4","task":"1","opcode":"0","keywords":"0x8000000000000000","systemTime":"2025-06-23T17:42:29.2840673Z"},"eventRecordID":"49","processID":"3308","threadID":"4480","channel":"Microsoft-Windows-Sysmon/Operational"},"computer":"server.server.local","severityValue":"INFORMATION","message":"\\Process Create:\\r\\nRuleName: -\\r\\nUtcTime: 2025-06-23 17:42:29.275\\r\\nProcessGuid: {e65b0504-92b5-6859-c201-000000003200}\\r\\nProcessId: 6644\\r\\nImage: C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\\r\\nFileVersion: 10.0.20348.1 (WinBuild.160101.0800)\\r\\nDescription: Windows PowerShell\\r\\nProduct: Microsoft® Windows® Operating System\\r\\nCompany: Microsoft Corporation\\r\\nOriginalFileName: Powershell.EXE\\r\\nCommandLine: \\C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\\ -ExecutionPolicy Bypass -File C:\\Users\\Administrator\\Desktop\\testing.ps1\\r\\nCurrentDirectory: C:\\Users\\Administrator\\r\\nUser: SERVER0\\Administrator\\r\\nLogonGuid: {e65b0504-92b5-6859-c201-000000003200}\\r\\nLogonId: 0x7A6B9\\r\\nTerminalSessionId: 1\\r\\nIntegrityLevel: High\\r\\nHashes: MD5=2E0C8B27064856E3D55017FA2D33A7B9\\r\\nParentProcessGuid: {e65b0504-91cc-6859-bf01-000000003200}\\r\\nParentProcessId: 5740\\r\\nParentImage: C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\\r\\nParentCommandLine: \\C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\\ \\r\\nParentUser: SERVER0\\Administrator\\r\\n"},"eventdata":{"utcTime":"2025-06-23 17:42:29.275","processGuid":{"e65b0504-92b5-6859-c201-000000003200"},"processId":"6644","image":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe","fileVersion":"10.0.20348.1 (WinBuild.160101.0800)","description":"Windows PowerShell","product":"Microsoft® Windows® Operating System","company":"Microsoft Corporation","originalFileName":"Powershell.EXE","commandLine":"\\C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\\ -ExecutionPolicy Bypass -File C:\\Users\\Administrator\\Desktop\\testing.ps1","currentDirectory":"C:\\Users\\Administrator\\","user":"SERVER0\\Administrator","logonGuid":{"e65b0504-91cc-6859-bf01-000000003200"},"parentProcessId":"5740","parentImage":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe","parentCommandLine":"\\C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\\ \\r\\nParentUser: SERVER0\\Administrator\\r\\n"},"location":"EventChannel"},"timestamp":"2025-06-23T13:41:38.680-0400","rule":{"level":6,"description":"PowerShell executed script from suspicious location","id":"92029","mitre":{"id":["T1059.001"],"tactic":["Execution"],"technique":["PowerShell"]},"firedtimes":2,"mail":false,"groups":{"sysmon","sysmon_eid1_detections","windows"},"agent":{"id":"007","name":"server","ip":"10.0.2.15"},"manager":{"name":"kali"},"id":"1750700498.17155117","decoder":{"name":"windows_eventchannel"},"data":{"win":{"system":{"providerName":"Microsoft-Windows-Sysmon","providerGuid":{"5770385f-c22a-43e0-bf4c-06f5698ff6b9"},"eventID":"1","version":"5","level":"4","task":"1","opcode":"0","keywords":"0x8000000000000000","systemTime":"2025-06-23T17:43:07.3854659Z"},"eventRecordID":"51","processID":"3308","threadID":"4480","channel":"Microsoft-Windows-Sysmon/Operational"},"computer":"server.server.local","severityValue":"INFORMATION","message":"\\Process Create:\\r\\nRuleName: -\\r\\nUtcTime: 2025-06-23 17:43:07.367\\r\\nProcessGuid: {e65b0504-922b-6859-c501-000000003200}\\r\\nProcessId: 4636\\r\\nImage: C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\\r\\nFileVersion: 10.0.20348.1 (WinBuild.160101.0800)\\r\\nDescription: Windows PowerShell\\r\\nProduct: Microsoft® Windows® Operating System\\r\\nCompany: Microsoft Corporation\\r\\nOriginalFileName: Powershell.EXE\\r\\nCommandLine: \\C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\\ -ExecutionPolicy Bypass -File C:\\Users\\Administrator\\Desktop\\testing.ps1\\r\\nCurrentDirectory: C:\\Users\\Administrator\\r\\nUser: SERVER0\\Administrator\\r\\nLogonGuid: {e65b0504-91cc-6859-bf01-000000003200}\\r\\nLogonId: 0x7A6B9\\r\\nTerminalSessionId: 1\\r\\nIntegrityLevel: High\\r\\nHashes: MD5=2E0C8B27064856E3D55017FA2D33A7B9\\r\\nParentProcessGuid: {e65b0504-922b-6859-c301-000000003200}\\r\\nParentProcessId: 4460\\r\\nParentImage: C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\\r\\nParentCommandLine: \\C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\\ \\r\\nParentUser: SERVER0\\Administrator\\r\\n"},"eventdata":{"utcTime":"2025-06-23 17:43:07.367","processGuid":{"e65b0504-922b-6859-c501-000000003200"},"processId":"4636","image":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe","fileVersion":"10.0.20348.1 (WinBuild.160101.0800)","description":"Windows PowerShell","product":"Microsoft® Windows® Operating Sys
```

# Log Verification through Wazuh Dashboard

After confirming the logs on the server-side, we further verified the PowerShell execution event through the Wazuh web interface:

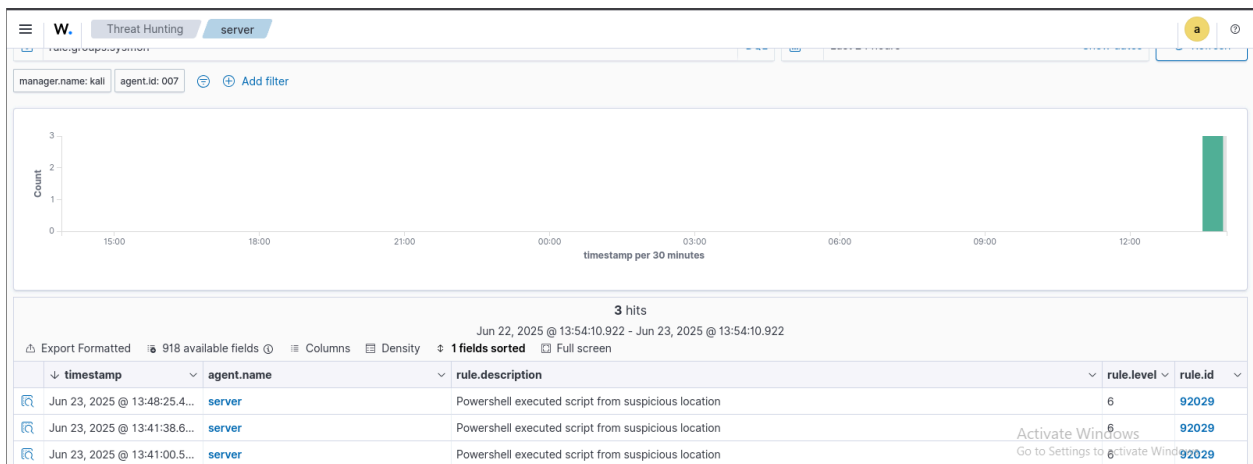
Opened the Wazuh Dashboard using Firefox browser.

Navigated to the Threat Hunting section.

Searched for the related logs using keywords like "PowerShell" or eventID 1.

Verified that the event generated from the Windows Agent (PowerShell script execution) was successfully captured and displayed in the Wazuh Dashboard.

This step confirmed that the Sysmon Event ID 1 (Process Creation) logs were properly received, processed, and visible in the Wazuh Dashboard for threat analysis.



The screenshot shows a web browser window with the address bar displaying a URL from a Kali Linux machine. The browser's address bar shows the URL: `https://10.10.90.80/app/discover#/doc/wazuh-alerts-4.x-2025.06.23?id=49XnnZcBy7PtZEp_--vT`. The browser's address bar also shows the text "Wazuh" and "Discover". The main content area displays a JSON alert from Wazuh. The alert details include:

- `@timestamp`: Jun 23, 2025 @ 13:48:25.435
- `_index`: wazuh-alerts-4.x-2025.06.23
- `agent.id`: 007
- `agent.ip`: 10.0.2.15
- `agent.name`: server
- `data.win.eventdata.commandLine`: `"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe" -ExecutionPolicy Bypass -File C:\\Users\\administrator\\Desktop\\testing.ps1`
- `data.win.eventdata.company`: Microsoft Corporation
- `data.win.eventdata.currentDirectory`: `C:\\Users\\administrator\\`
- `data.win.eventdata.description`: Windows PowerShell
- `data.win.eventdata.fileVersion`: 10.0.20348.1 (WinBuild.160101.0800)
- `data.win.eventdata.hashes`: MD5=2E0CCB279648563D505077FA2D33A7B9
- `data.win.eventdata.image`: `C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe`
- `data.win.eventdata.integrityLevel`: High
- `data.win.eventdata.logonGuid`: {e65b0504-7eff-6859-b9a6-070000000000}
- `data.win.eventdata.logonId`: 0x7a6b9

The command line field is highlighted with a red box. The command line is: `"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe" -ExecutionPolicy Bypass -File C:\\Users\\administrator\\Desktop\\testing.ps1`. The text "Activate Windows" and "Go to Settings to activate Windows." is visible in the bottom right corner of the browser window.

W. Discover wazuh-alerts-4.x-2025.06.23#49XnnZcBy7PizEp_--vT		a	0
data.win.eventdata.originalFileName	PowerShell.EXE		
data.win.eventdata.parentCommandLine	"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe"		
data.win.eventdata.parentImage	C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe		
data.win.eventdata.parentProcessGuid	{e65b0504-93c2-6859-cf01-000000003200}		
data.win.eventdata.parentProcessId	7612		
data.win.eventdata.parentUser	SERVER0\\Administrator		
data.win.eventdata.processGuid	{e65b0504-93c2-6859-cf01-000000003200}		
data.win.eventdata.processId	6028		
data.win.eventdata.product	Microsoft Windows Operating System		
data.win.eventdata.terminalSessionId	1		
data.win.eventdata.user	SERVER0\\Administrator		
data.win.eventdata.utcTime	2025-06-23 17:49:54.057		
data.win.system.channel	Microsoft-Windows-Sysmon/Operational		
data.win.system.computer	server.server.local		
data.win.system.eventId	1		
data.win.system.eventRecordId	53		
data.win.system.keywords	0x8000000000000000		
Activate Windows Go to Settings to activate Windows.			
data.win.system.message	"Process Create: RuleName: - UtcTime: 2025-06-23 17:49:54.057 ProcessGuid: {e65b0504-93c2-6859-cf01-000000003200} ProcessId: 6028 Image: C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe FileVersion: 10.0.20348.1 /WinDbg1d 16d101 00001		
data.win.system.opcode	0		
data.win.system.processId	3308		
data.win.system.providerGuid	{5770385f-c22a-43e0-bf4c-06f5698ffbd9}		
data.win.system.providerName	Microsoft-Windows-Sysmon		
data.win.system.severityValue	INFORMATION		
data.win.system.systemTime	2025-06-23T17:49:54.0698099Z		
data.win.system.task	1		
data.win.system.threadId	4480		
data.win.system.version	5		
decoder.name	windows_eventchannel		
id	1750700905.17300965		
input.type	log		
location	EventChannel		
Activate Windows Go to Settings to activate Windows.			

## Summary:

### Setup & Configuration

Sysmon was installed and configured on the Windows Server to capture detailed system events, specifically process creation events involving PowerShell execution.

Wazuh agent was installed on the Windows Server to forward event logs to the Wazuh manager running on Kali Linux.

The Wazuh manager was configured to receive, parse, and analyze Windows Event Channel logs, including custom Sysmon events.

### Log Capture & Processing

Upon execution of a PowerShell script (`testing.ps1`) on the Windows Server using a bypass execution policy command, Sysmon generated detailed event logs describing the process creation, command line arguments, user details, and process hierarchy.

The Wazuh agent forwarded these logs in real-time to the Wazuh manager.

The Wazuh manager parsed the logs using built-in decoders and rules, matched a relevant detection rule (id: 92029) that identifies suspicious PowerShell execution activities, and generated alerts accordingly.

### Verification & Testing

The logs and alerts were verified on the Wazuh manager, confirming successful ingestion and rule triggering.

Troubleshooting involved validating XML configurations, correcting ruleset inclusions, and ensuring that the Wazuh manager service was running without errors.

### Outcome

This setup enables real-time detection and alerting of key Sysmon events on monitored Windows hosts, such as process creation, file timestamp changes, network connections, and Sysmon errors. Monitoring process creation is especially important because it reveals every program or script that runs, helping to identify suspicious activities like unauthorized PowerShell executions. Tracking changes to file creation times can uncover attempts to hide or manipulate files. Observing network connections helps detect unusual communications from the system. Additionally, logging Sysmon errors ensures that any problems with monitoring are quickly noticed. Together, these rules provide a strong and effective way to continuously monitor system activity and enhance security.

