Window Powershell Integration

Introduction

In this lab, we integrate Windows PowerShell with the Wazuh SIEM platform to monitor PowerShell activities on a Windows system. PowerShell is a powerful tool used by system administrators to manage Windows systems, but attackers also use it to run malicious commands. Because of this, monitoring PowerShell activity is very important for system security.

By integrating PowerShell with Wazuh, we can collect PowerShell logs, detect suspicious commands, and monitor system behavior from one central place. This helps security teams to quickly identify threats and respond to attacks.

Environment Overview

Windows Server 2022
This system is used as the target endpoint. The Wazuh Agent is installed on this machine to collect system, security, and PowerShell logs.

Ubuntu Server (Wazuh Manager)
The Wazuh Manager is installed on an Ubuntu server. This server acts as the central monitoring and analysis system where all logs are received, security rules are applied, and alerts are generated.

Network Connectivity
A stable network connection is required between the Windows Server and the Ubuntu-based Wazuh Manager so that logs can be sent from the agent to the manager in real time.

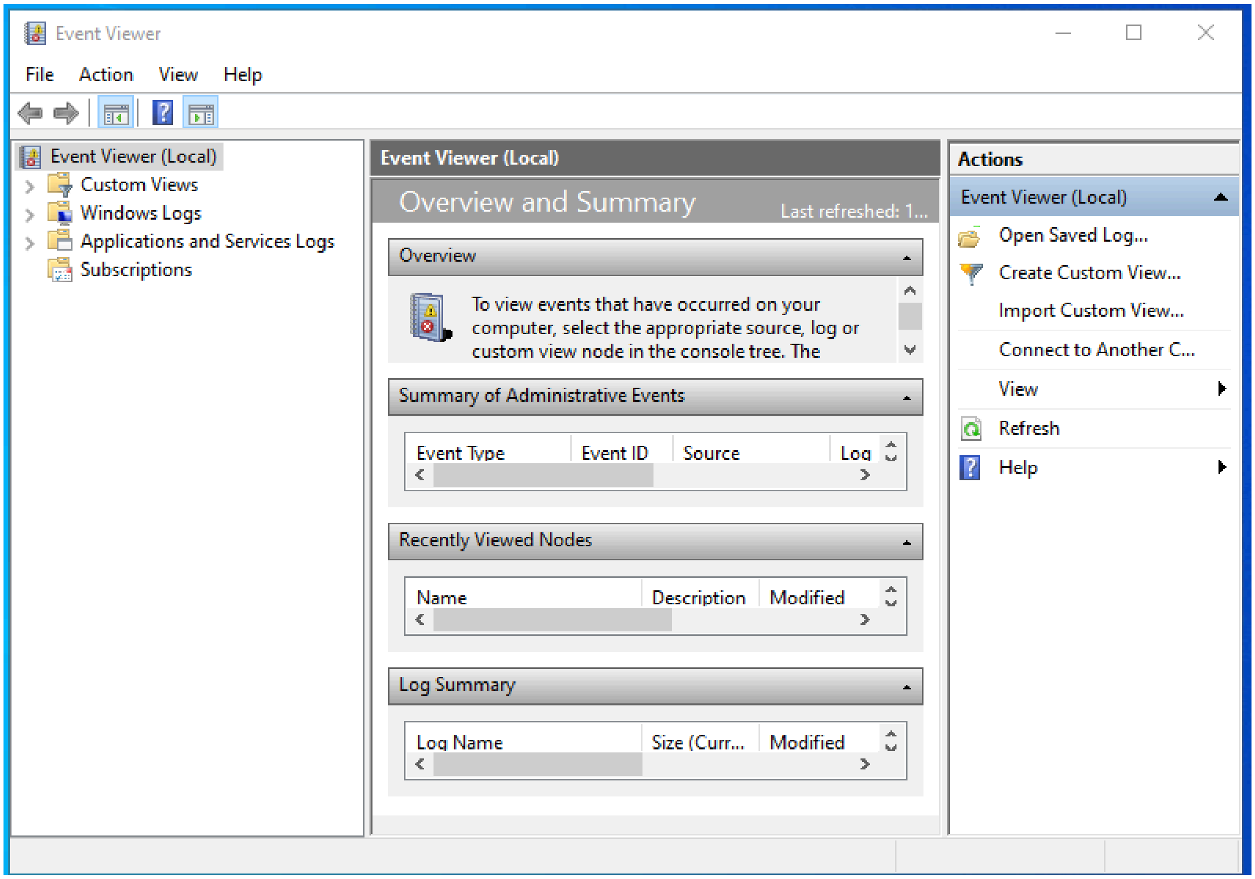
Steps:

Step 1: Enable PowerShell Logging

PowerShell logging is enabled so that Wazuh can read and monitor PowerShell activity on the Windows system. This includes normal commands, scripts, and suspicious actions that may indicate a security threat.

Enable PowerShell Operational Logs

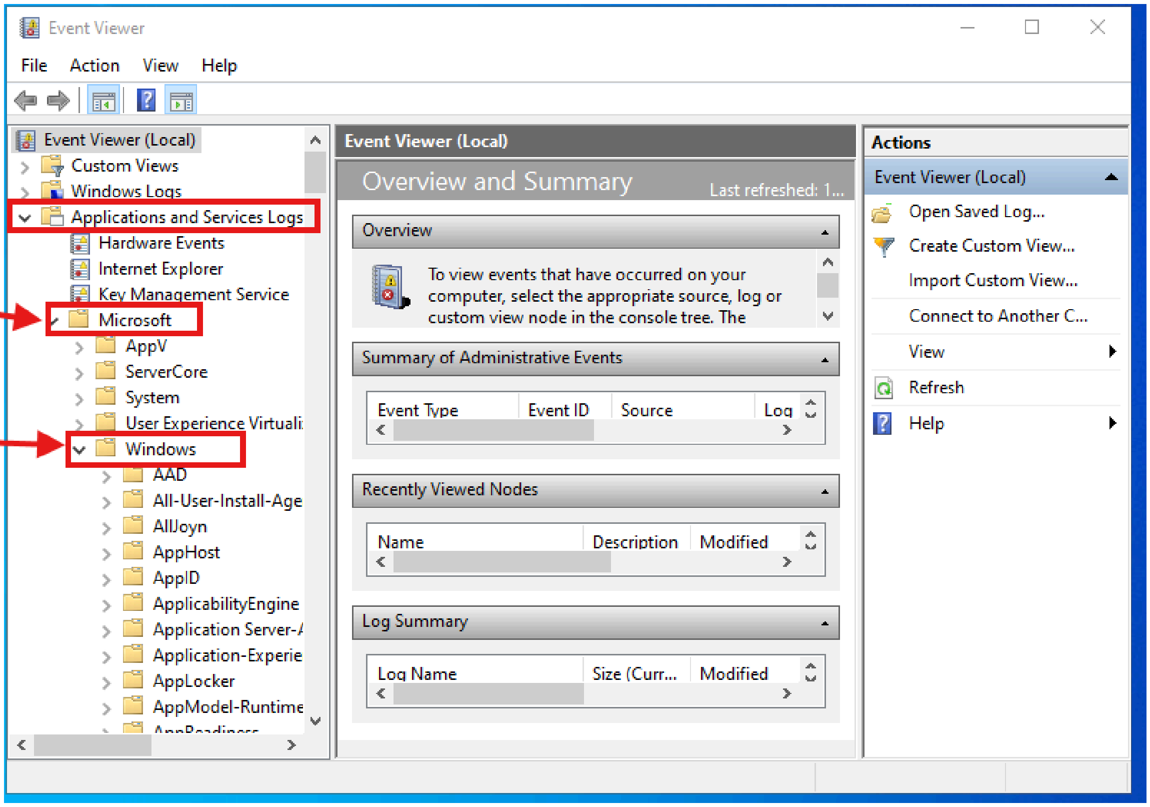
First, PowerShell Operational logs are enabled on Windows Server using Event Viewer. To do this, the Event Viewer is searched from the Start menu and opened.



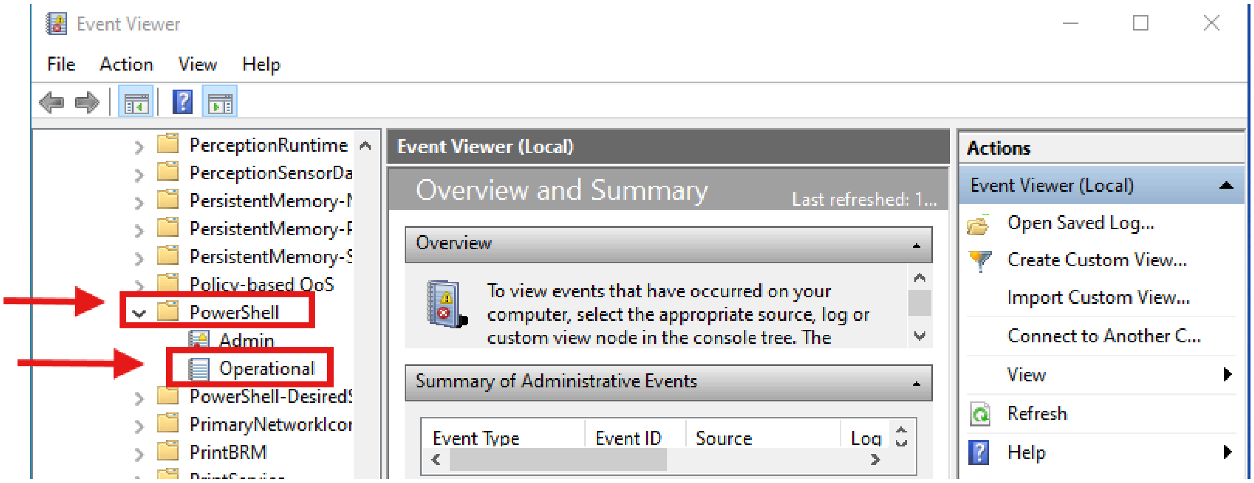
To view events that have occurred on your computer, select the appropriate source, log or custom view node in the console tree. The

In Event Viewer, we navigate to the following path:

Application and Service Logs > Microsoft > Windows Powershell > Operational

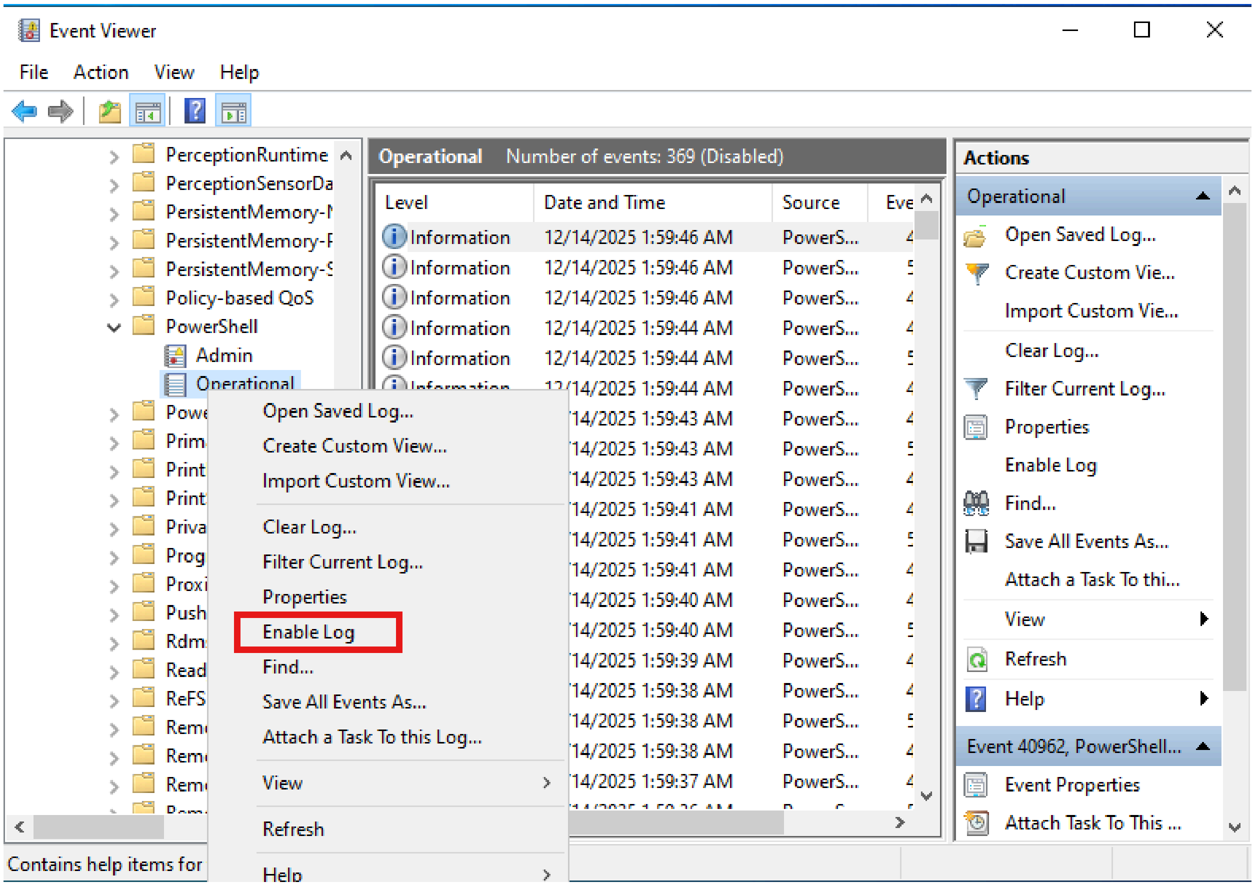


To view events that have occurred on your computer, select the appropriate source, log or custom view node in the console tree. The



To view events that have occurred on your computer, select the appropriate source, log or custom view node in the console tree. The

Right-click on the Operational and select Enable Log.



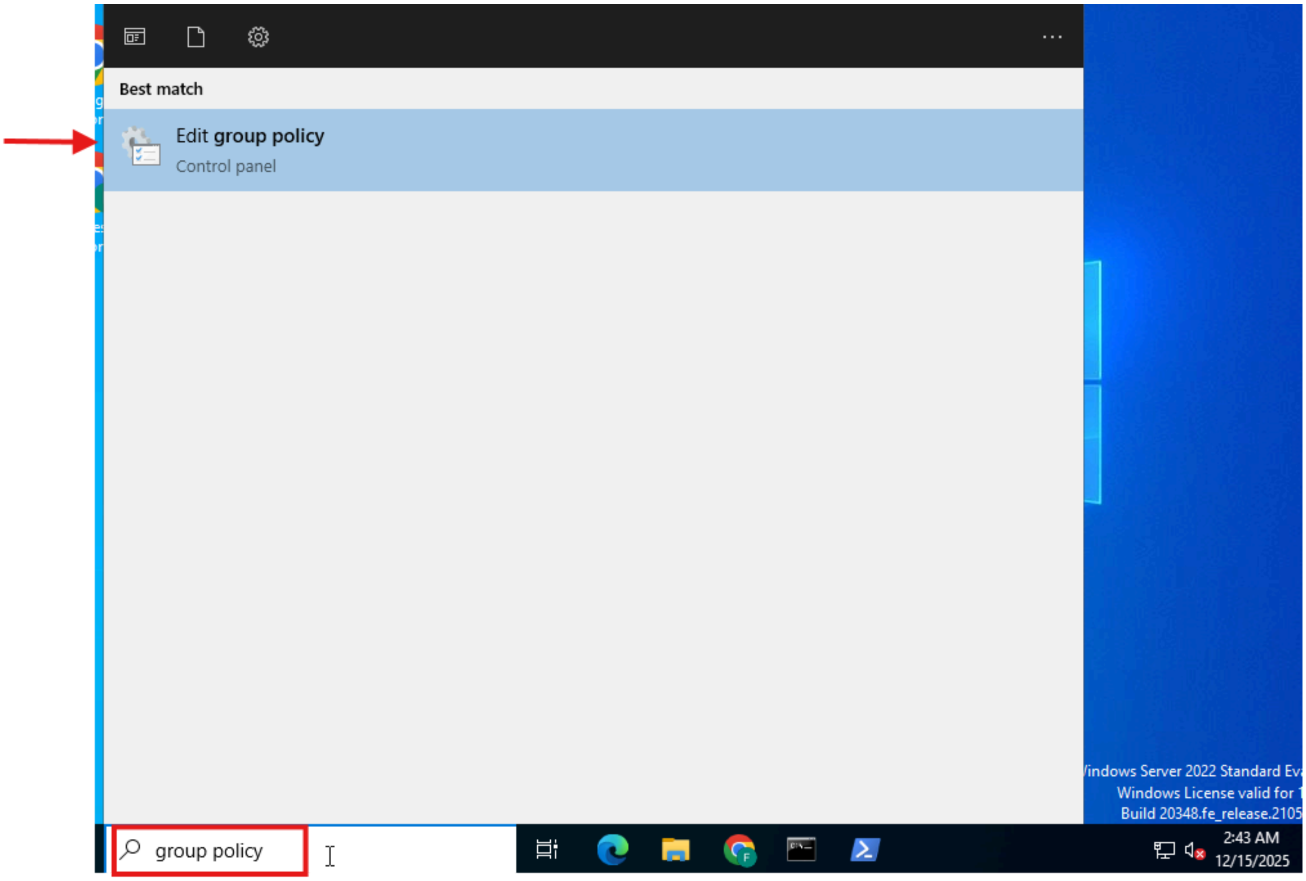
Operational   Number of events: 369 (Disabled)

Level	Date and Time	Source	Eve
Information	12/14/2025 1:59:46 AM	PowerS...	4
Information	12/14/2025 1:59:46 AM	PowerS...	5
Information	12/14/2025 1:59:46 AM	PowerS...	4
Information	12/14/2025 1:59:44 AM	PowerS...	4
Information	12/14/2025 1:59:44 AM	PowerS...	5
Information	12/14/2025 1:59:44 AM	PowerS...	4
	14/2025 1:59:43 AM	PowerS...	4
	14/2025 1:59:43 AM	PowerS...	5
	14/2025 1:59:43 AM	PowerS...	4
	14/2025 1:59:41 AM	PowerS...	4
	14/2025 1:59:41 AM	PowerS...	5
	14/2025 1:59:41 AM	PowerS...	4
	14/2025 1:59:40 AM	PowerS...	4
	14/2025 1:59:40 AM	PowerS...	5
	14/2025 1:59:39 AM	PowerS...	4
	14/2025 1:59:38 AM	PowerS...	4
	14/2025 1:59:38 AM	PowerS...	5
	14/2025 1:59:38 AM	PowerS...	4
	14/2025 1:59:37 AM	PowerS...	4

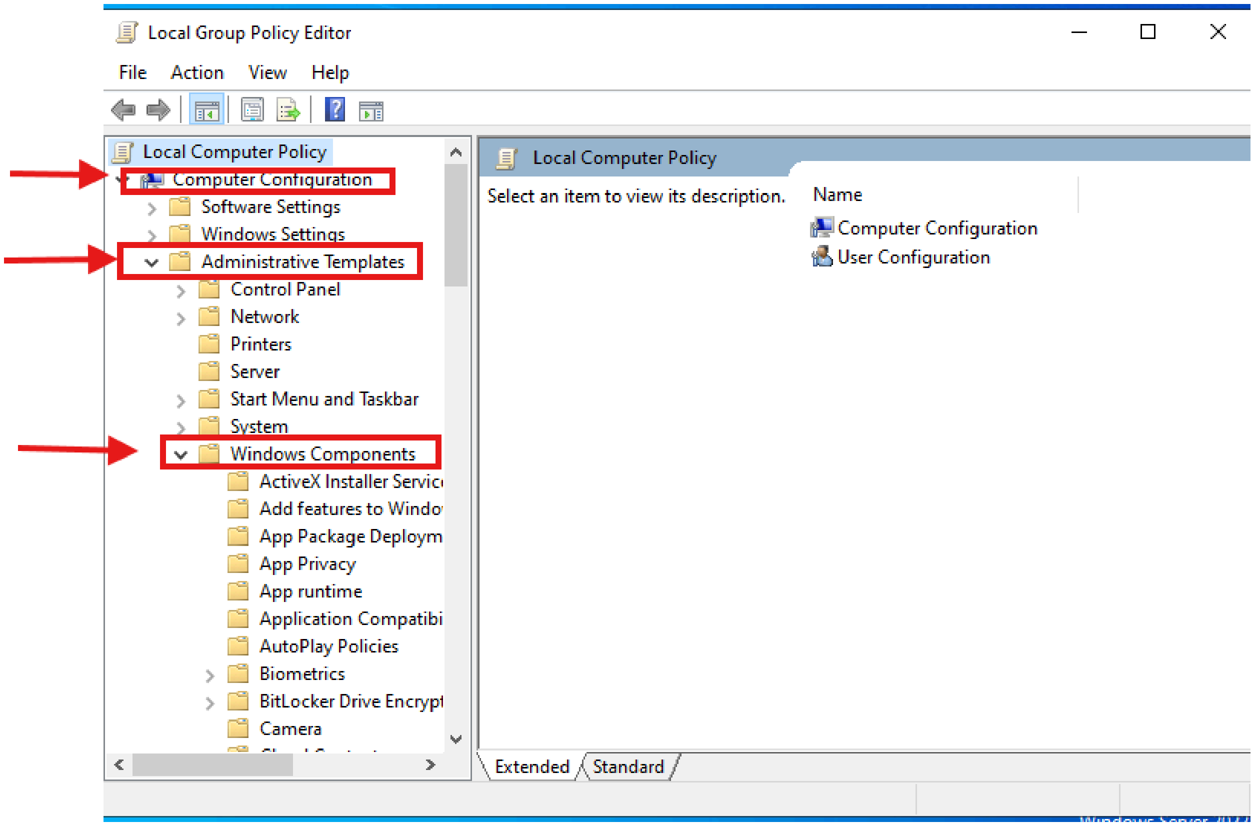
PowerShell Operational logging is now enabled. Windows will record basic PowerShell activity, such as command execution events and errors.

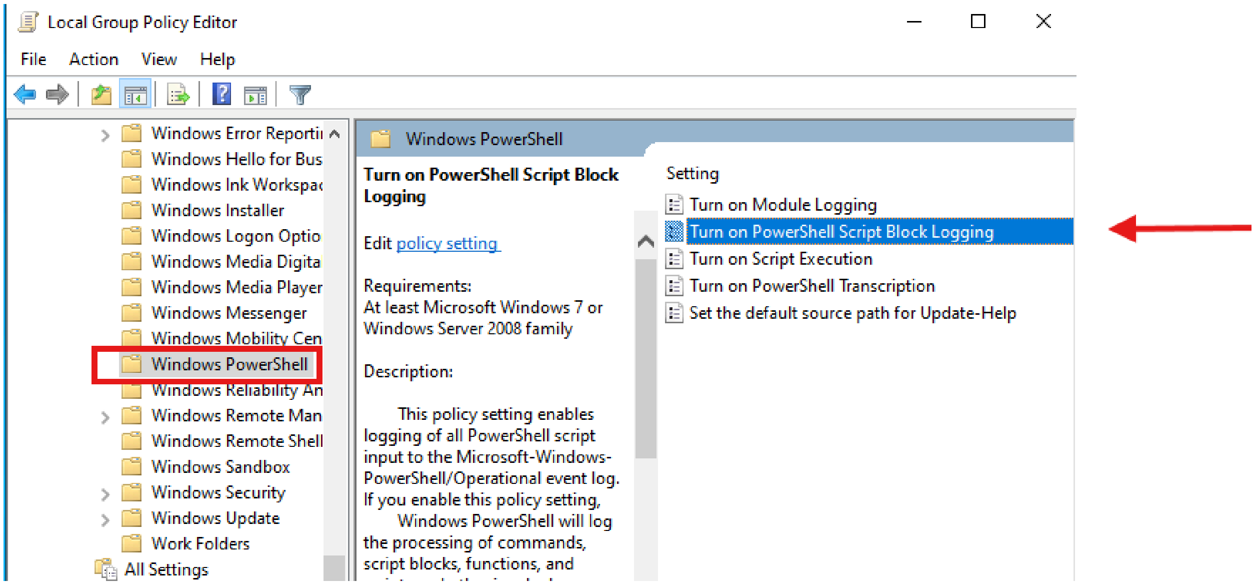
Enable PowerShell Script Block Logging and Module Logging

Script Block Logging records the full PowerShell commands and scripts executed on the system.
To enable Script Block Logging, “Edit Group Policy” is searched from the Start menu and opened.


Best match
Edit group policy
Control panel
group policy
Vindows Server 2022 Standard Eva
Windows License valid for 1
Build 20348.fe_release.2105
2:43 AM
12/15/2025

Navigate to the following path:
Computer Configuration > Administrative Templates > Windows Components > Windows PowerShell





Turn on PowerShell Script Block Logging

Edit policy setting

Requirements:
At least Microsoft Windows 7 or Windows Server 2008 family

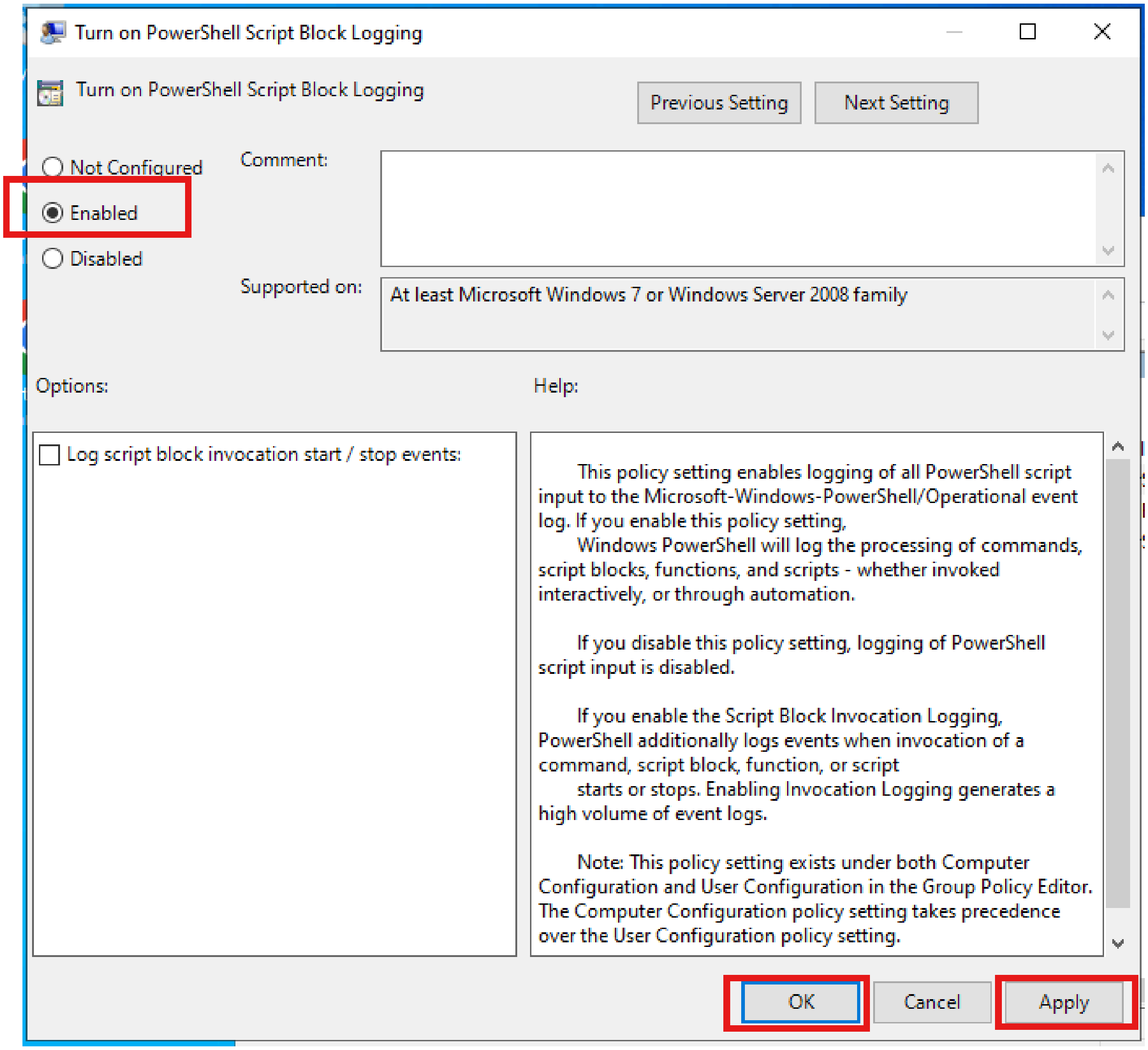
Description:

This policy setting enables logging of all PowerShell script input to the Microsoft-Windows-PowerShell/Operational event log. If you enable this policy setting, Windows PowerShell will log the processing of commands, script blocks, functions, and

Double-click “Turn on PowerShell Script Block Logging”.

Select Enabled.

Click Apply, then click OK.



Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

□ Log script block invocation start / stop events:

This policy setting enables logging of all PowerShell script input to the Microsoft-Windows-PowerShell/Operational event log. If you enable this policy setting,
Windows PowerShell will log the processing of commands, script blocks, functions, and scripts - whether invoked interactively, or through automation.

If you disable this policy setting, logging of PowerShell script input is disabled.

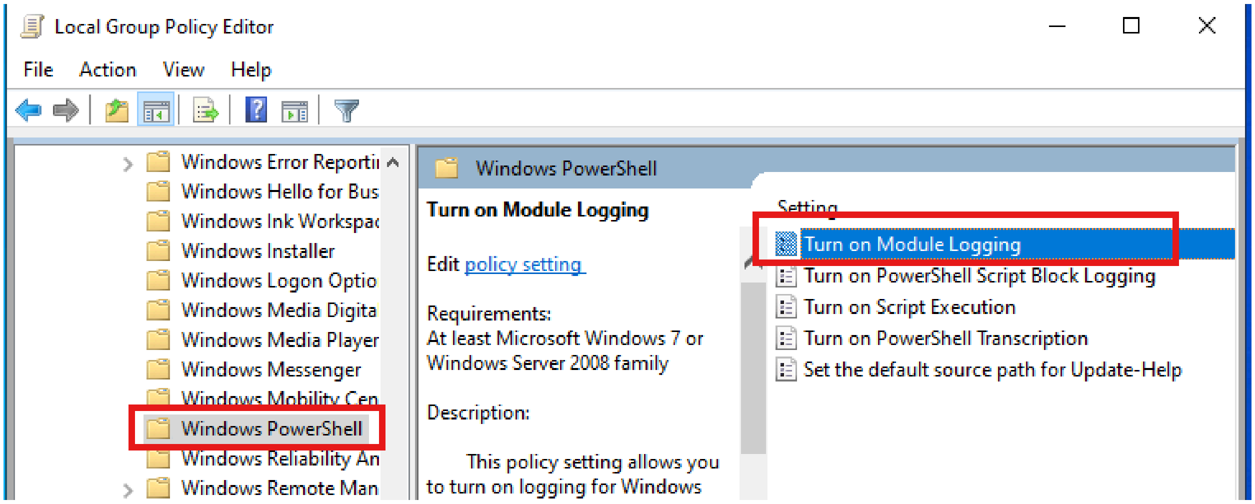
If you enable the Script Block Invocation Logging, PowerShell additionally logs events when invocation of a command, script block, function, or script
starts or stops. Enabling Invocation Logging generates a high volume of event logs.

Note: This policy setting exists under both Computer Configuration and User Configuration in the Group Policy Editor. The Computer Configuration policy setting takes precedence over the User Configuration policy setting.

Now Script Block Logging is enabled.

After enabling Script Block Logging, **Module Logging** is also enabled from the same Windows PowerShell policy location.

The policy “Turn on Module Logging” is opened and set to Enabled, then click Apply, and click OK



Windows PowerShell

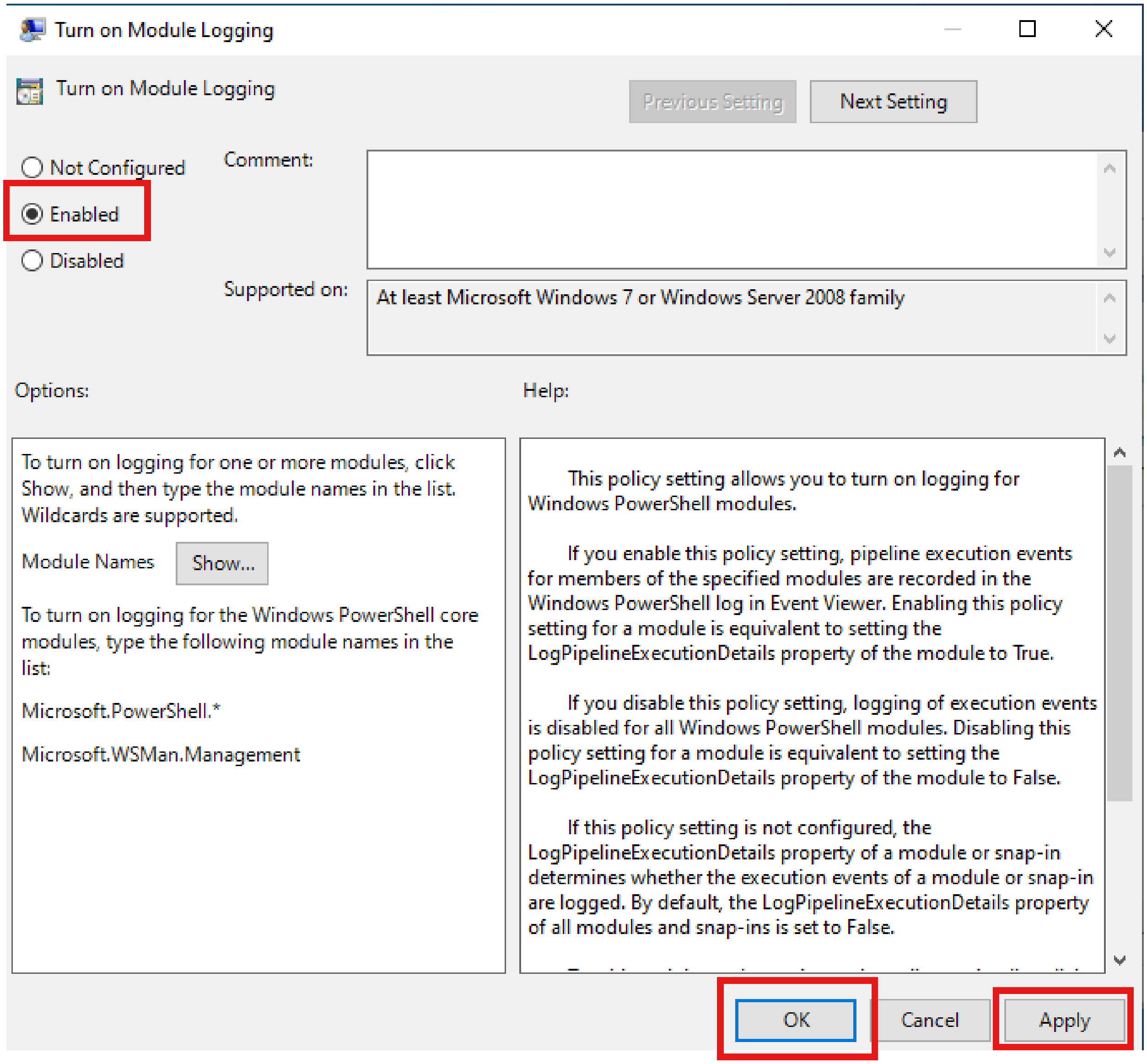
Turn on Module Logging

Edit policy setting

Requirements:
At least Microsoft Windows 7 or Windows Server 2008 family

Description:

This policy setting allows you to turn on logging for Windows



○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

Options:

To turn on logging for one or more modules, click Show, and then type the module names in the list. Wildcards are supported.

Module Names

To turn on logging for the Windows PowerShell core modules, type the following module names in the list:

Microsoft.PowerShell.*

Microsoft.WSMan.Management

Help:

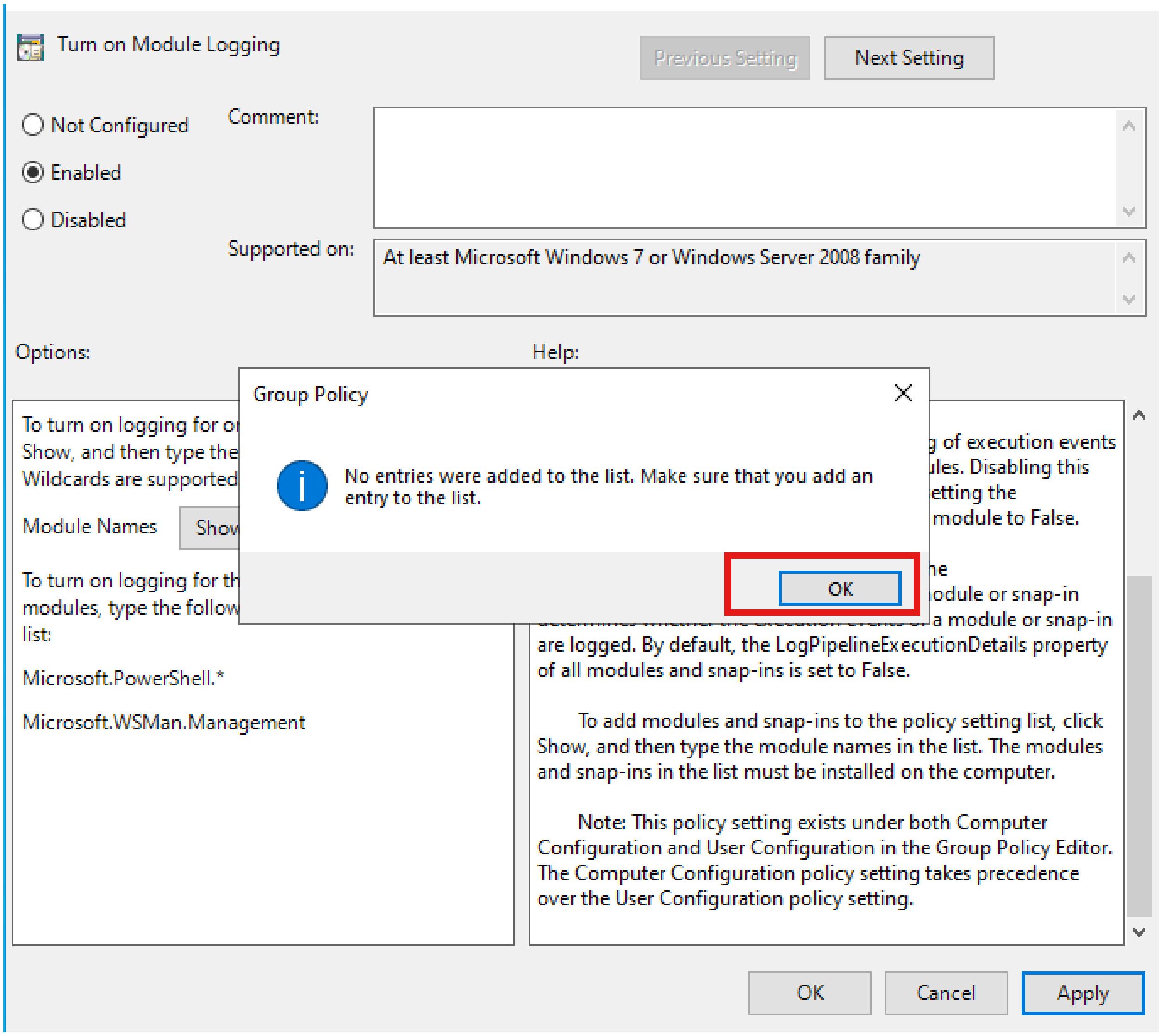
This policy setting allows you to turn on logging for Windows PowerShell modules.

If you enable this policy setting, pipeline execution events for members of the specified modules are recorded in the Windows PowerShell log in Event Viewer. Enabling this policy setting for a module is equivalent to setting the LogPipelineExecutionDetails property of the module to True.

If you disable this policy setting, logging of execution events is disabled for all Windows PowerShell modules. Disabling this policy setting for a module is equivalent to setting the LogPipelineExecutionDetails property of the module to False.

If this policy setting is not configured, the LogPipelineExecutionDetails property of a module or snap-in determines whether the execution events of a module or snap-in are logged. By default, the LogPipelineExecutionDetails property of all modules and snap-ins is set to False.

When you click Apply, a popup may appear saying:
"No entries were added to the list. Make sure that you add an entry to the list."
To resolve this, you need to add the module names in the list before applying the policy.



Turn on Module Logging

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

Options:

To turn on logging for o
Show, and then type the
Wildcards are supported

Module Names

To turn on logging for th
modules, type the follow
list:

Microsoft.PowerShell.*

Microsoft.WSMan.Management

Help:

g of execution events
ules. Disabling this
etting the
module to False.

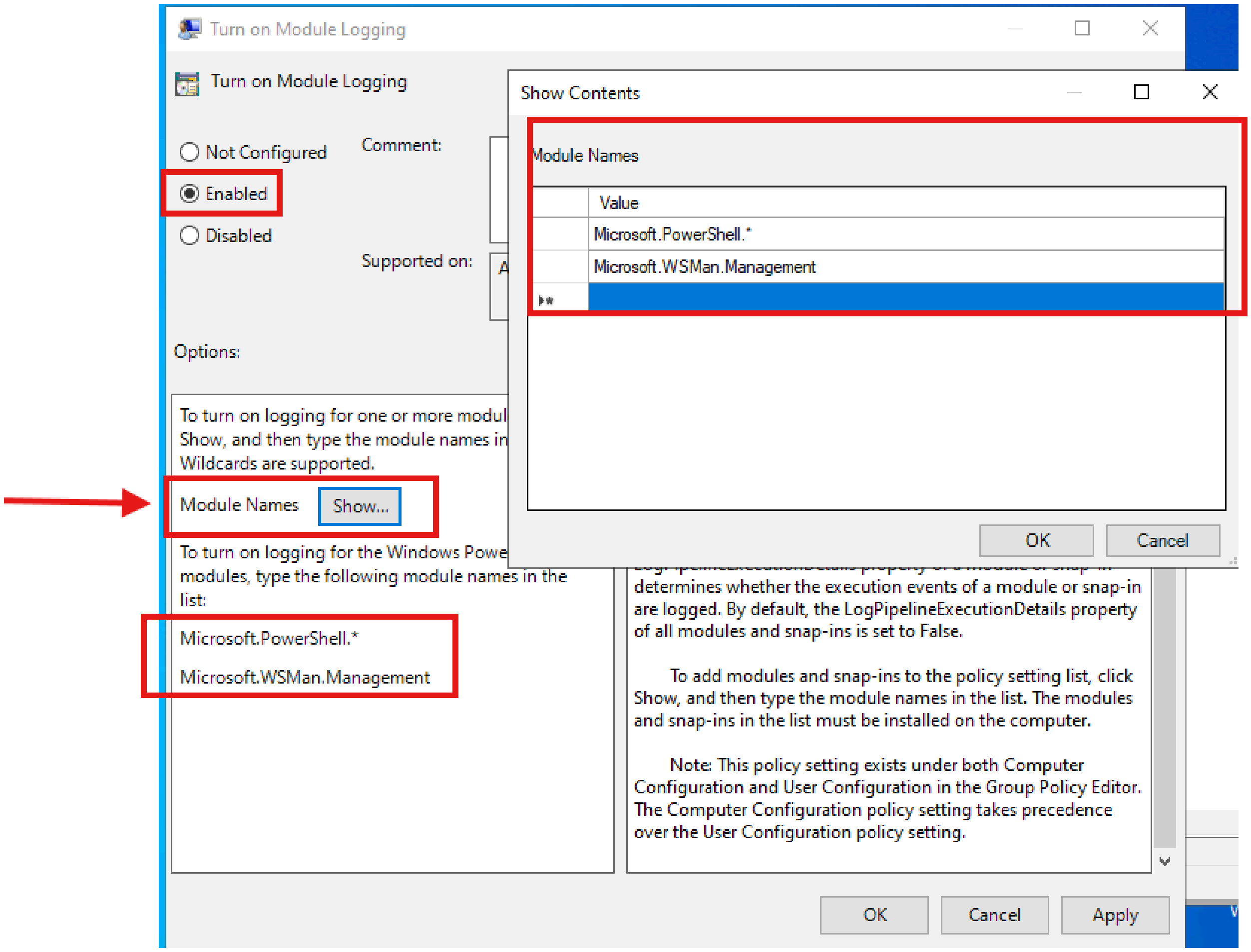
he
odule or snap-in
a module or snap-in
are logged. By default, the LogPipelineExecutionDetails property of all modules and snap-ins is set to False.

To add modules and snap-ins to the policy setting list, click Show, and then type the module names in the list. The modules and snap-ins in the list must be installed on the computer.

Note: This policy setting exists under both Computer Configuration and User Configuration in the Group Policy Editor. The Computer Configuration policy setting takes precedence over the User Configuration policy setting.

Group Policy

No entries were added to the list. Make sure that you add an entry to the list.



Turn on Module Logging

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: A

Options:

To turn on logging for one or more modul
Show, and then type the module names in
Wildcards are supported.

Module Names Show...

To turn on logging for the Windows Powe
modules, type the following module names in the
list:

Microsoft.PowerShell.*

Microsoft.WSMan.Management

determines whether the execution events of a module or snap-in are logged. By default, the LogPipelineExecutionDetails property of all modules and snap-ins is set to False.

To add modules and snap-ins to the policy setting list, click Show, and then type the module names in the list. The modules and snap-ins in the list must be installed on the computer.

Note: This policy setting exists under both Computer Configuration and User Configuration in the Group Policy Editor. The Computer Configuration policy setting takes precedence over the User Configuration policy setting.

Show Contents

Module Names

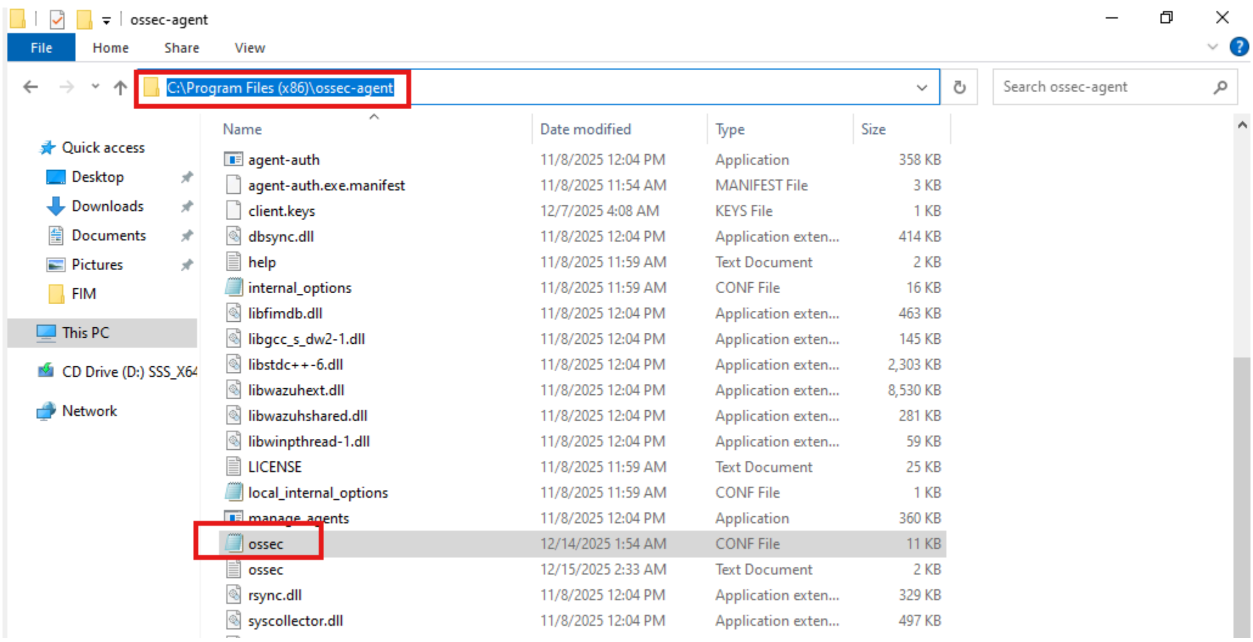
Value
Microsoft.PowerShell.*
Microsoft.WSMan.Management

After adding the module names, click OK, then click Apply, and finally click OK again.

Configure Wazuh Agent

Open the Wazuh agent configuration file on Windows (default path):

```
notepad "C:\Program Files (x86)\ossec-agent\ossec.conf"
```

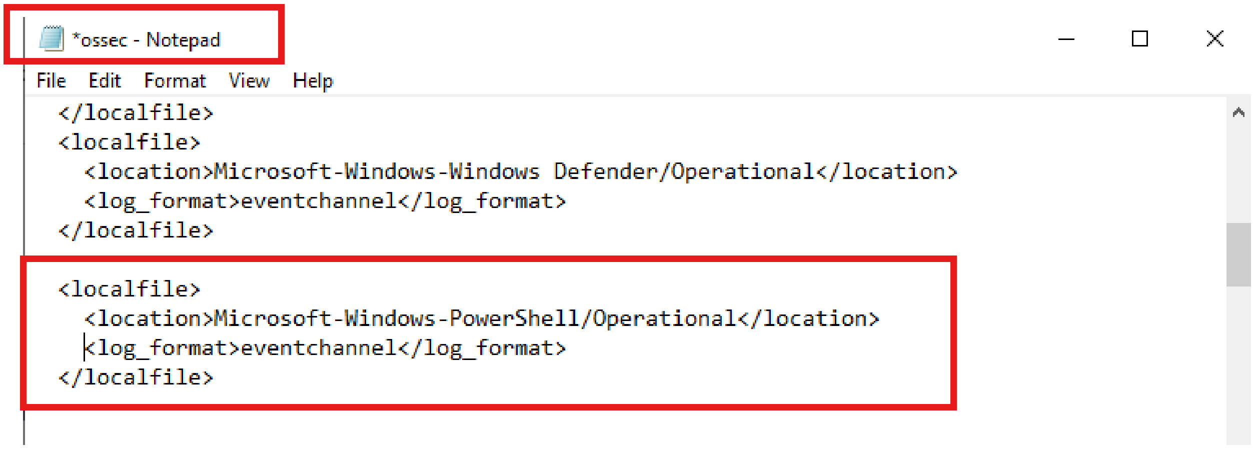


C:\Program Files (x86)\ossec-agent

Name	Date modified	Type	Size
agent-auth	11/8/2025 12:04 PM	Application	358 KB
agent-auth.exe.manifest	11/8/2025 11:54 AM	MANIFEST File	3 KB
client.keys	12/7/2025 4:08 AM	KEYS File	1 KB
dbsync.dll	11/8/2025 12:04 PM	Application exten...	414 KB
help	11/8/2025 11:59 AM	Text Document	2 KB
internal_options	11/8/2025 11:59 AM	CONF File	16 KB
libfimdb.dll	11/8/2025 12:04 PM	Application exten...	463 KB
libgcc_s_dw2-1.dll	11/8/2025 12:04 PM	Application exten...	145 KB
libstdc++-6.dll	11/8/2025 12:04 PM	Application exten...	2,303 KB
libwazuhext.dll	11/8/2025 12:04 PM	Application exten...	8,530 KB
libwazuhshared.dll	11/8/2025 12:04 PM	Application exten...	281 KB
libwinpthread-1.dll	11/8/2025 12:04 PM	Application exten...	59 KB
LICENSE	11/8/2025 11:59 AM	Text Document	25 KB
local_internal_options	11/8/2025 11:59 AM	CONF File	1 KB
manage_agents	11/8/2025 12:04 PM	Application	360 KB
ossec	12/14/2025 1:54 AM	CONF File	11 KB
ossec	12/15/2025 2:33 AM	Text Document	2 KB
rsync.dll	11/8/2025 12:04 PM	Application exten...	329 KB
syscollector.dll	11/8/2025 12:04 PM	Application exten...	497 KB

Add the following inside <ossec_config> to instruct the agent to read the PowerShell Operational event channel

```
<localfile>

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

  <log_format>eventchannel</log_format>

</localfile>
```

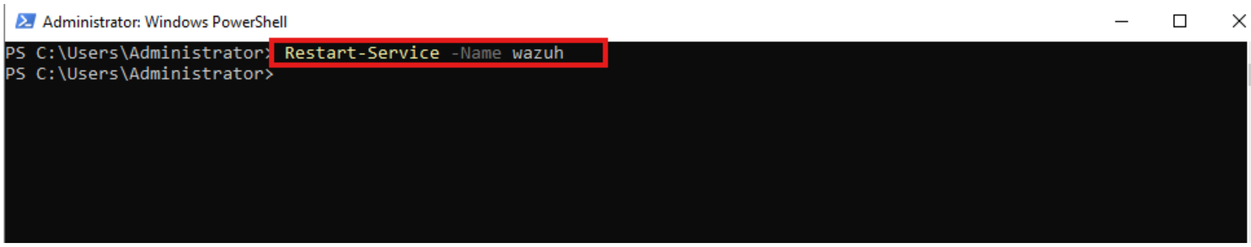


```
</localfile>
<localfile>
  <location>Microsoft-Windows-Windows Defender/Operational</location>
  <log_format>eventchannel</log_format>
</localfile>

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

After configuration, save the changes.

Restart the Wazuh agent service for changes to take effect:

Restart-Service -Name wazuh

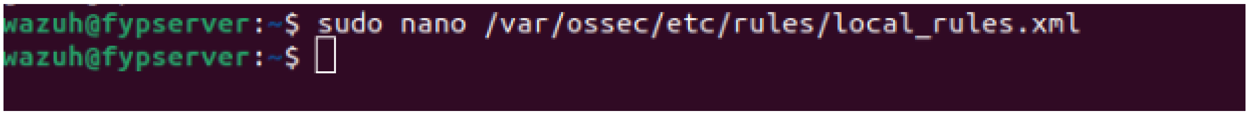


```
PS C:\Users\Administrator> Restart-Service -Name wazuh
PS C:\Users\Administrator>
```

Configure Wazuh Manager

The manager uses rules to identify suspicious or malicious activities from the collected logs.
To configure this, edit the local rules file on the Ubuntu server where the Wazuh Manager is installed:

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



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

Insert the following comprehensive rules for PowerShell exploitation detection:

<group name="windows,powershell">

<!-- 100201: Encoded PowerShell command detected -->

<rule id="100201" level="8">

<if_sid>60009</if_sid>

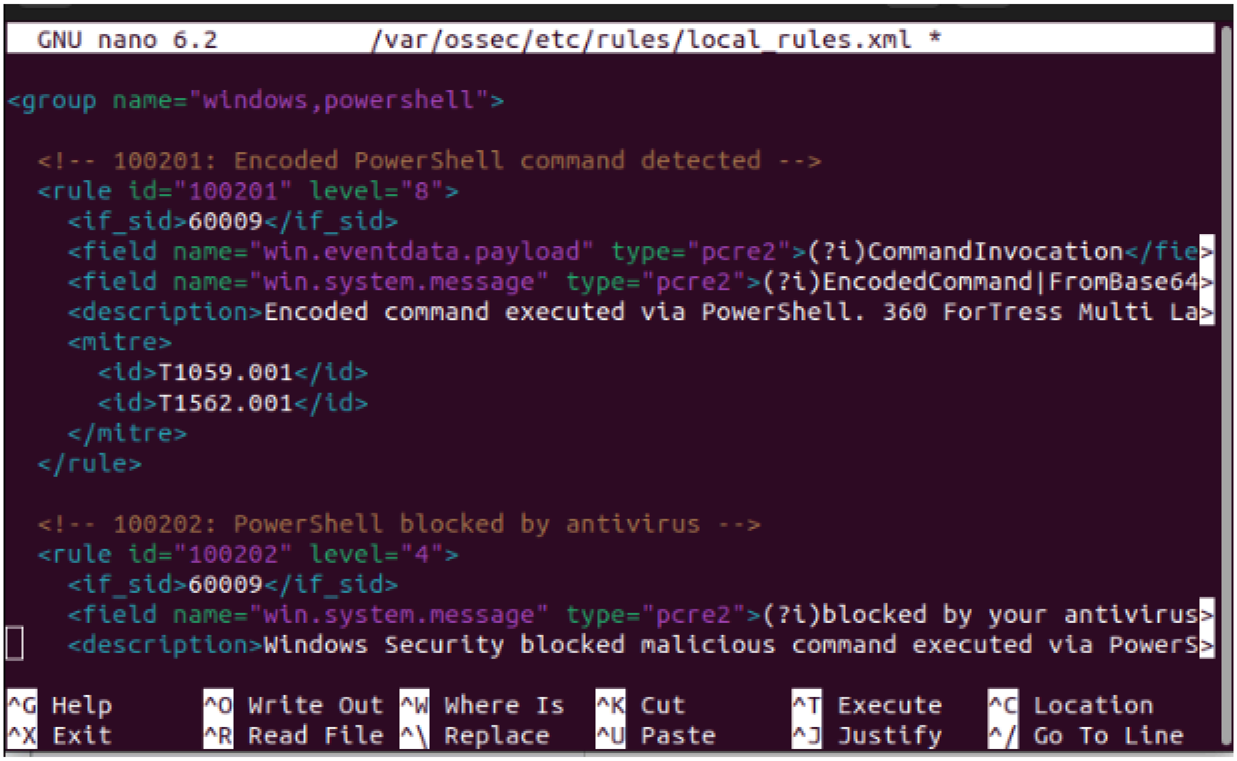
<field name="win.eventdata.payload" type="pcre2">(?i)CommandInvocation</field>

<field name="win.system.message" type="pcre2">(?i)EncodedCommand|FromBase64String|EncodedArguments|-e\b|-enco\b|-en\b</field>

<description>Encoded command executed via PowerShell. 360 ForTress Multi Layer Cyber Protection System</description>

```
    <mitre>
      <id>T1059.001</id>
      <id>T1562.001</id>
    </mitre>
  </rule>

  <!-- 100202: PowerShell blocked by antivirus -->
  <rule id="100202" level="4">
    <if_sid>60009</if_sid>
    <field name="win.system.message" type="pcre2">(?i)blocked by your antivirus software</field>
    <description>Windows Security blocked malicious command executed via PowerShell.</description>
    <mitre>
      <id>T1059.001</id>
    </mitre>
  </rule>

  <!-- 100203: Malicious cmdlet like Invoke-Mimikatz detected -->
  <rule id="100203" level="10">
    <if_sid>60009</if_sid>
    <field name="win.eventdata.payload" type="pcre2">(?i)CommandInvocation</field>
    <field name="win.system.message" type="pcre2">(?i)Add-Persistence|Find-AVSignature|Invoke-Mimikatz|Invoke-Shellcode|Set-MasterBootRecord</field>
    <description>Risky CMDLet executed. Possible malicious activity detected. 360 ForTress Multi Layer Cyber Protection System</description>
    <mitre>
      <id>T1059.001</id>
    </mitre>
  </rule>
  <!-- 100204: mshta used for suspicious download -->
  <rule id="100204" level="8">
    <if_sid>91802</if_sid>
    <field name="win.eventdata.scriptBlockText"
type="pcre2">(?i)mshta.*GetObject|mshta.*new ActiveXObject</field>
    <description>Mshta used to download a file. Possible malicious activity detected. 360
ForTress Multi Layer Cyber Protection System</description>
    <mitre>
      <id>T1059.001</id>
    </mitre>
  </rule>

  <!-- 100205: Execution policy set to bypass -->
  <rule id="100205" level="5">
    <if_sid>60009</if_sid>
    <field name="win.eventdata.contextInfo"
type="pcre2">(?i)ExecutionPolicy\s*bypass|exec\s*bypass</field>
    <description>PowerShell execution policy set to bypass.</description>
    <mitre>
      <id>T1059.001</id>
    </mitre>
  </rule>

  <!-- 100206: Invoke-WebRequest or IWR used -->
  <rule id="100206" level="5">
    <if_sid>60009</if_sid>
    <field name="win.eventdata.contextInfo"
type="pcre2">(?i)Invoke-WebRequest|IWR.*-url|IWR.*-InFile</field>
    <description>Invoke-WebRequest executed, possible download cradle
detected.</description>
```

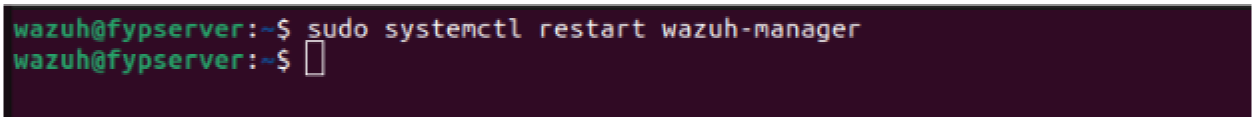
```
    <mitre>
      <id>T1059.001</id>
    </mitre>
  </rule>

</group>
```



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

<group name="windows,powershell">

  <!-- 100201: Encoded PowerShell command detected -->
  <rule id="100201" level="8">
    <if_sid>60009</if_sid>
    <field name="win.eventdata.payload" type="pcre2">(?i)CommandInvocation</fie>
    <field name="win.system.message" type="pcre2">(?i)EncodedCommand|FromBase64>
    <description>Encoded command executed via PowerShell. 360 ForTress Multi La>
    <mitre>
      <id>T1059.001</id>
      <id>T1562.001</id>
    </mitre>
  </rule>

  <!-- 100202: PowerShell blocked by antivirus -->
  <rule id="100202" level="4">
    <if_sid>60009</if_sid>
    <field name="win.system.message" type="pcre2">(?i)blocked by your antivirus>
    <description>Windows Security blocked malicious command executed via PowerS>

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

Restart Wazuh manager to apply rules:

sudo systemctl restart wazuh-manager

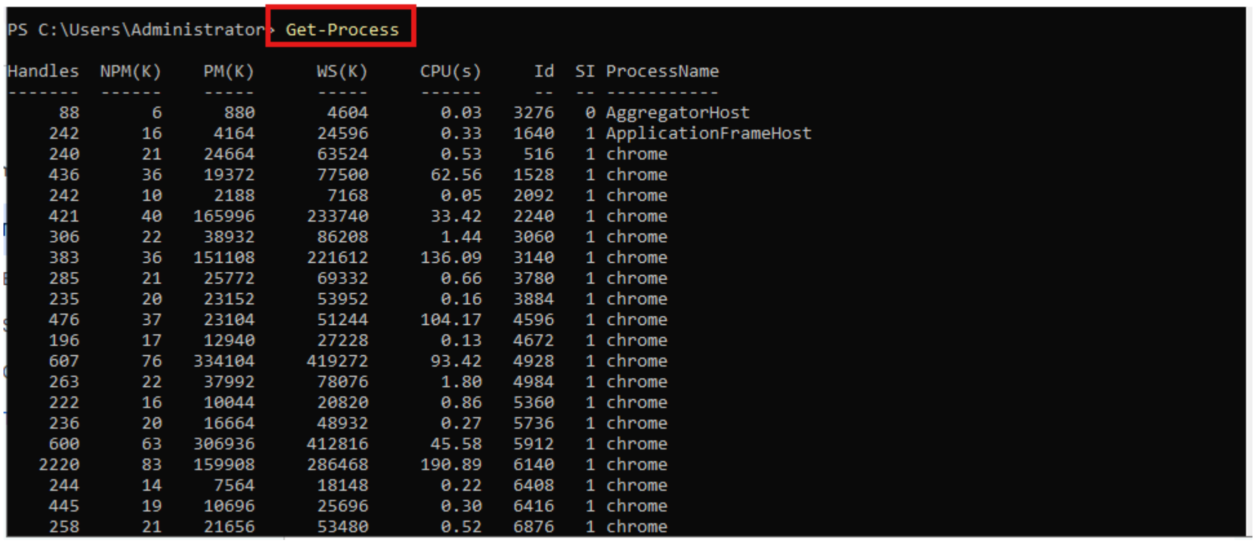


```
wazuh@fypserver:~$ sudo systemctl restart wazuh-manager
wazuh@fypserver:~$ 
```

Testing Phase:

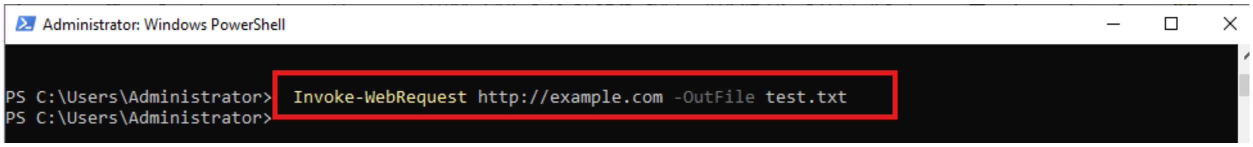
Basic PowerShell Command:

Test1: Get-Process



```
PS C:\Users\Administrator> Get-Process

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)     Id  SI ProcessName
-------  ------    -----      -----     ------     --  -- -----------
     88       6      880       4604       0.03   3276   0 AggregatorHost
    242      16     4164      24596       0.33   1640   1 ApplicationFrameHost
    240      21    24664      63524       0.53    516   1 chrome
    436      36    19372      77500      62.56   1528   1 chrome
    242      10     2188       7168       0.05   2092   1 chrome
    421      40   165996     233740      33.42   2240   1 chrome
    306      22    38932      86208       1.44   3060   1 chrome
    383      36   151108     221612     136.09   3140   1 chrome
    285      21    25772      69332       0.66   3780   1 chrome
    235      20    23152      53952       0.16   3884   1 chrome
    476      37    23104      51244     104.17   4596   1 chrome
    196      17    12940      27228       0.13   4672   1 chrome
    607      76   334104     419272      93.42   4928   1 chrome
    263      22    37992      78076       1.80   4984   1 chrome
    222      16    10044      20820       0.86   5360   1 chrome
    236      20    16664      48932       0.27   5736   1 chrome
    600      63   306936     412816      45.58   5912   1 chrome
   2220      83   159908     286468     190.89   6140   1 chrome
    244      14     7564      18148       0.22   6408   1 chrome
    445      19    10696      25696       0.30   6416   1 chrome
    258      21    21656      53480       0.52   6876   1 chrome
```

Test2: Invoke-WebRequest http://example.com -OutFile test.txt

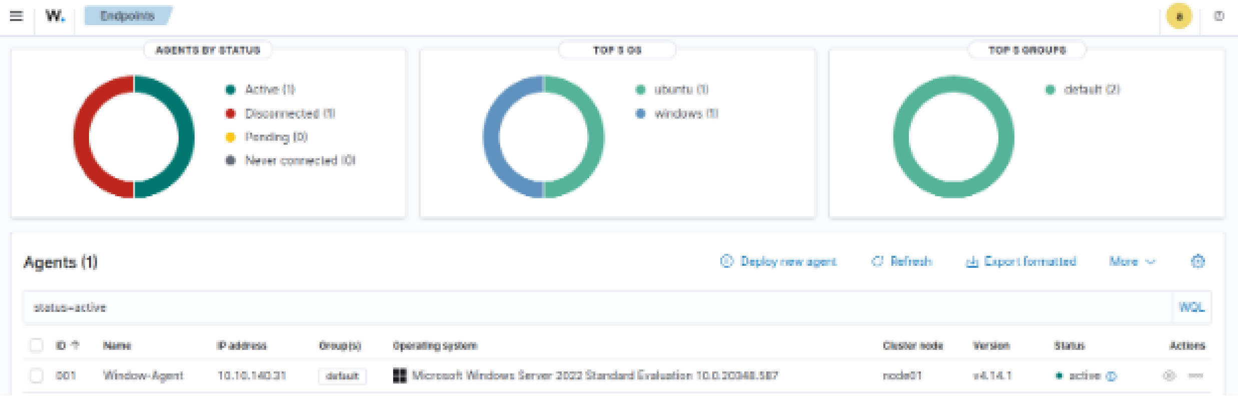


```
PS C:\Users\Administrator> Invoke-WebRequest http://example.com -OutFile test.txt
PS C:\Users\Administrator>
```

Now, Open the Wazuh Dashboard and navigate to Security Events

Confirm that the alert for the PowerShell encoded command (Invoke-WebRequest download cradle) is visible in the logs.

First see the Wazuh Dashboard for connectivity between the Window Agent and Manager.

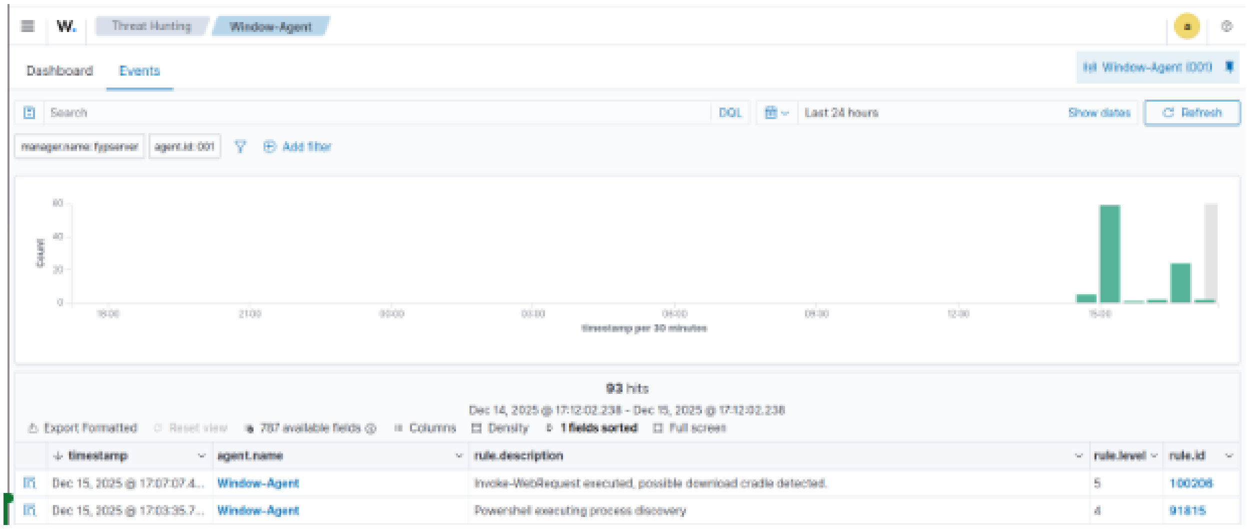


Agents (1)

status=active

ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions
001	Window-Agent	10.10.140.31	default	Microsoft Windows Server 2022 Standard Evaluation 10.0.20348.587	node01	v4.14.1	active	

Here are the alerts:

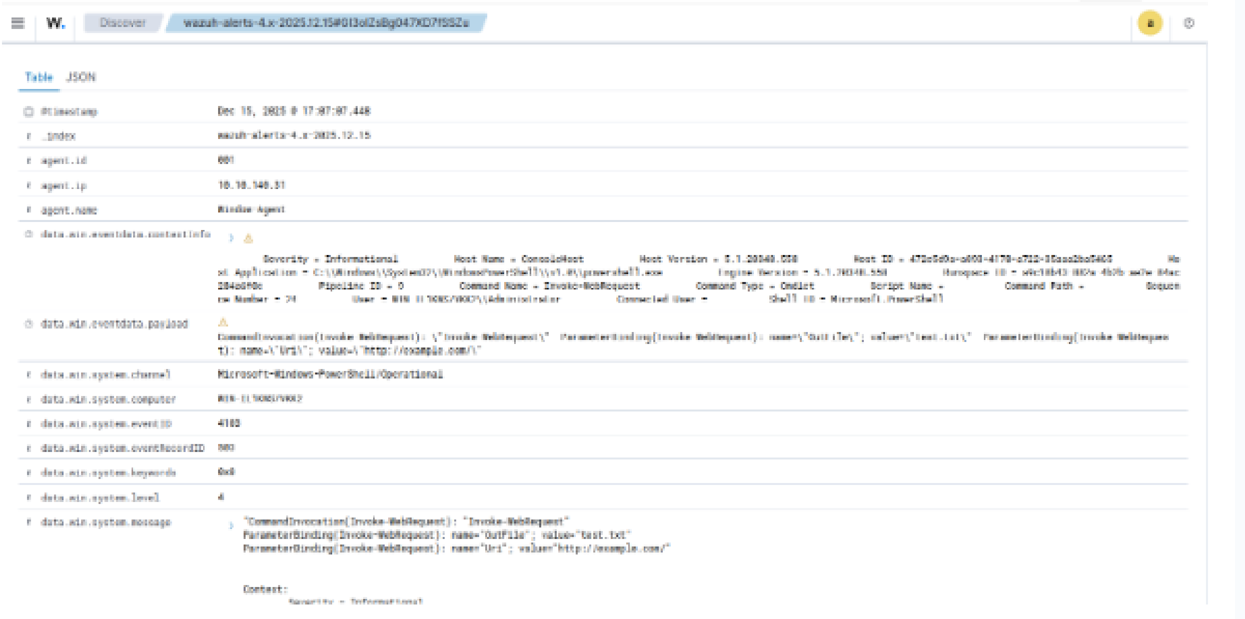


93 hits

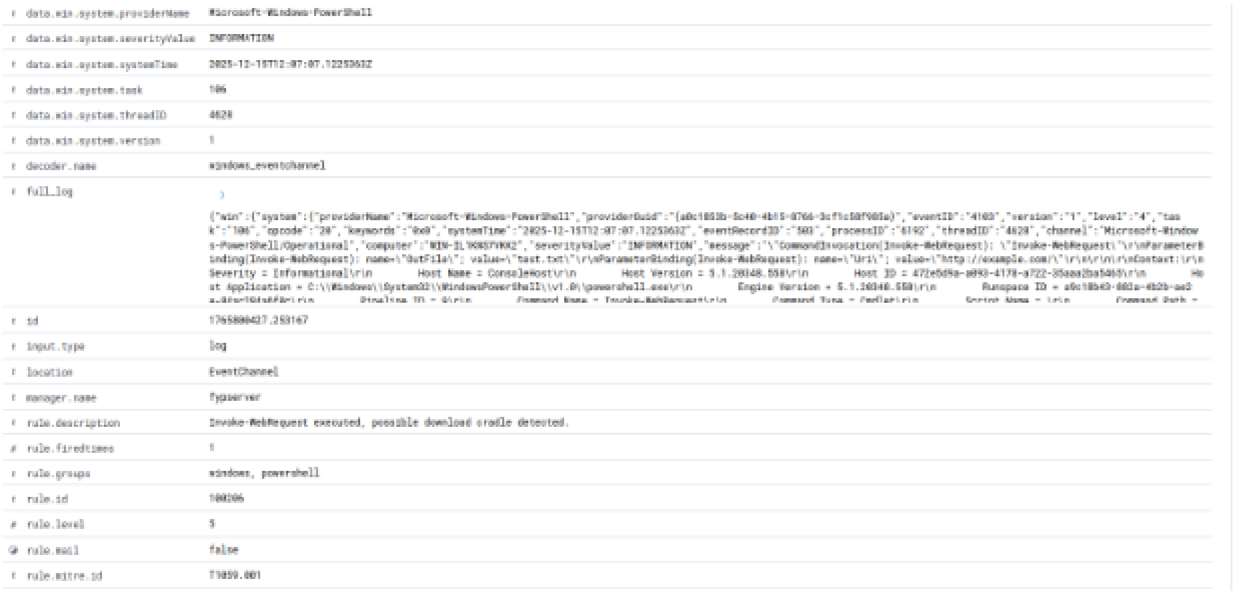
Dec 14, 2025 @ 17:12:02.238 - Dec 15, 2025 @ 17:12:02.238

timestamp	agent.name	rule.description	rule.level	rule.id
Dec 15, 2025 @ 17:07:07.4...	Window-Agent	Invoke-WebRequest executed, possible download cradle detected.	5	100208
Dec 15, 2025 @ 17:03:35.7...	Window-Agent	Powershell executing process discovery	4	91815

By clicking on one alert we see the details of alerts.

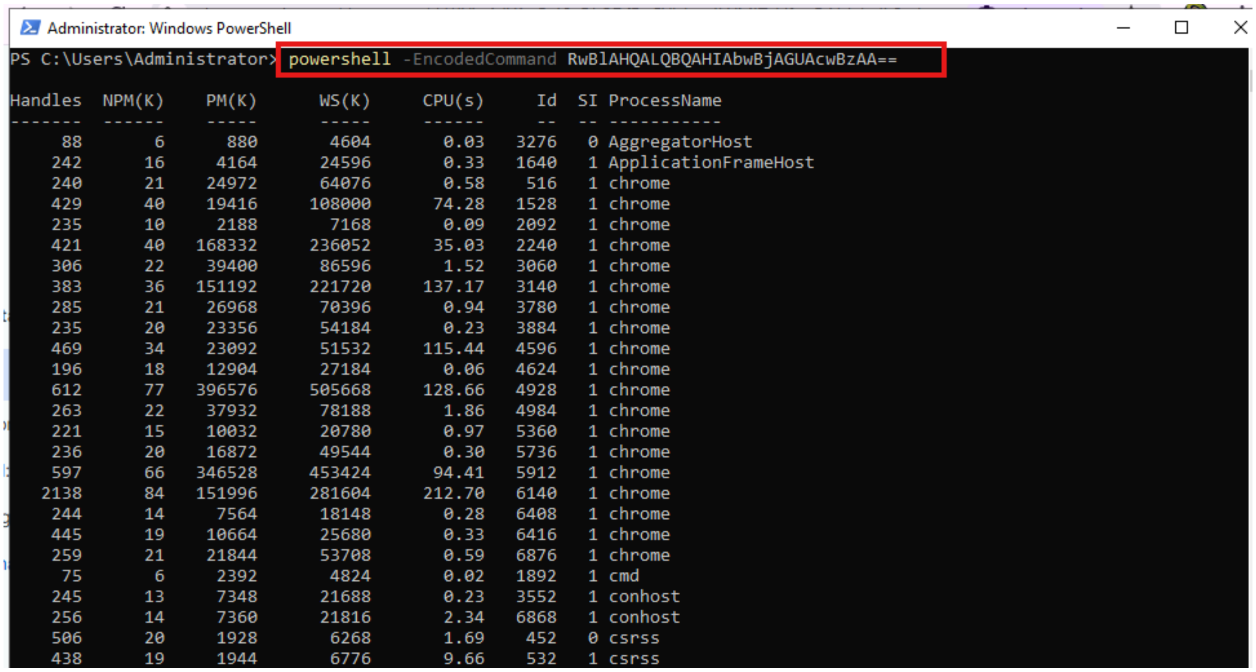


```
@timestamp                        Dec 15, 2025 @ 17:07:07.448
_index                            wazuh-alerts-4.x-2025.12.15
agent.id                          001
agent.ip                          10.10.140.31
agent.name                        Window-Agent
data.win.eventdata.contextInfo
                                          Severity = Informational          Host Name = ConsoleHost          Host Version = 5.1.20348.558          Host ID = 472e5d9a-a093-4170-a722-35aaa2ba5465          Ho
                                  st Application = C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe          Engine Version = 5.1.20348.558          Runspace ID = a9c18b43-082a-4b2b-ae2e-84ec
                                  284a0f0c          Pipeline ID = 0          Command Name = Invoke-WebRequest          Command Type = Cmdlet          Script Name =          Command Path =          Sequen
                                  ce Number = 24          User = WIN-IL1K0S7VKK2\\Administrator          Connected User =          Shell ID = Microsoft.PowerShell
data.win.eventdata.payload
                                  CommandInvocation(Invoke-WebRequest): \"Invoke-WebRequest\"  ParameterBinding(Invoke-WebRequest): name=\"OutFile\"; value=\"test.txt\"  ParameterBinding(Invoke-WebReques
                                  t): name=\"Uri\"; value=\"http://example.com/\"
data.win.system.channel           Microsoft-Windows-PowerShell/Operational
data.win.system.computer          WIN-IL1K0S7VKK2
data.win.system.eventID           4103
data.win.system.eventRecordID     503
data.win.system.keywords          0x0
data.win.system.level             4
data.win.system.message             "CommandInvocation(Invoke-WebRequest): "Invoke-WebRequest"
                                    ParameterBinding(Invoke-WebRequest): name="OutFile"; value="test.txt"
                                    ParameterBinding(Invoke-WebRequest): name="Uri"; value="http://example.com/"


                                    Context:
                                            Severity = Informational
```



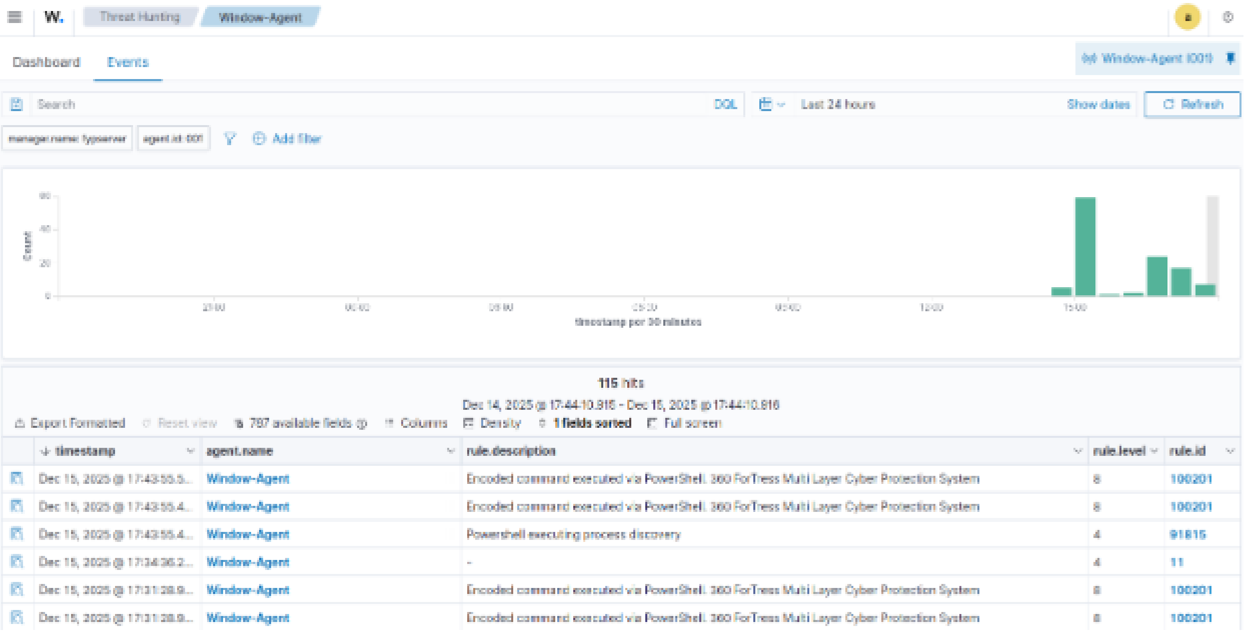
```
data.win.system.providerName      Microsoft-Windows-PowerShell
data.win.system.severityValue     INFORMATION
data.win.system.systemTime        2025-12-15T12:07:07.1225363Z
data.win.system.task              106
data.win.system.threadID          4628
data.win.system.version           1
decoder.name                      windows_eventchannel
full_log

                                  {"win":{"system":{"providerName":"Microsoft-Windows-PowerShell","providerGuid":"{a0c1853b-5c40-4b15-8766-3cf1c58f985a}","eventID":"4103","version":"1","level":"4","tas
                                  k":"106","opcode":"20","keywords":"0x0","systemTime":"2025-12-15T12:07:07.1225363Z","eventRecordID":"503","processID":"6192","threadID":"4628","channel":"Microsoft-Window
                                  s-PowerShell/Operational","computer":"WIN-IL1K0S7VKK2","severityValue":"INFORMATION","message":"\"CommandInvocation(Invoke-WebRequest): \"Invoke-WebRequest\"\r\nParameterB
                                  inding(Invoke-WebRequest): name=\"OutFile\"; value=\"test.txt\"\r\nParameterBinding(Invoke-WebRequest): name=\"Uri\"; value=\"http://example.com/\"\r\n\r\n\r\nContext:\r\n
                                  Severity = Informational\r\n          Host Name = ConsoleHost\r\n          Host Version = 5.1.20348.558\r\n          Host ID = 472e5d9a-a093-4170-a722-35aaa2ba5465\r\n          Ho
                                  st Application = C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\r\n          Engine Version = 5.1.20348.558\r\n          Runspace ID = a9c18b43-082a-4b2b-ae2
                                  e-84ec284a0f0c\r\n          Pipeline ID = 0\r\n          Command Name = Invoke-WebRequest\r\n          Command Type = Cmdlet\r\n          Script Name = \r\n          Command Path =
id                                1765800427.253167
input.type                        log
location                          EventChannel
manager.name                      fypserver
rule.description                  Invoke-WebRequest executed, possible download cradle detected.
rule.firedtimes                   1
rule.groups                       windows, powershell
rule.id                           100206
rule.level                        5
rule.mail                         false
rule.mitre.id                     T1059.001
```

Test 3: powershell -EncodedCommand RwBlAHQALQBQAHIAbwBjAGUAcwBzAA==



```
PS C:\Users\Administrator> powershell -EncodedCommand RwBlAHQALQBQAHIAbwBjAGUAcwBzAA==

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)     Id  SI ProcessName
-------  ------    -----      -----     ------     --  -- -----------
     88       6      880       4604       0.03   3276   0 AggregatorHost
    242      16     4164      24596       0.33   1640   1 ApplicationFrameHost
    240      21    24972      64076       0.58    516   1 chrome
    429      40    19416     108000      74.28   1528   1 chrome
    235      10     2188       7168       0.09   2092   1 chrome
    421      40   168332     236052      35.03   2240   1 chrome
    306      22    39400      86596       1.52   3060   1 chrome
    383      36   151192     221720     137.17   3140   1 chrome
    285      21    26968      70396       0.94   3780   1 chrome
    235      20    23356      54184       0.23   3884   1 chrome
    469      34    23092      51532     115.44   4596   1 chrome
    196      18    12904      27184       0.06   4624   1 chrome
    612      77   396576     505668     128.66   4928   1 chrome
    263      22    37932      78188       1.86   4984   1 chrome
    221      15    10032      20780       0.97   5360   1 chrome
    236      20    16872      49544       0.30   5736   1 chrome
    597      66   346528     453424      94.41   5912   1 chrome
   2138      84   151996     281604     212.70   6140   1 chrome
    244      14     7564      18148       0.28   6408   1 chrome
    445      19    10664      25680       0.33   6416   1 chrome
    259      21    21844      53708       0.59   6876   1 chrome
     75       6     2392       4824       0.02   1892   1 cmd
    245      13     7348      21688       0.23   3552   1 conhost
    256      14     7360      21816       2.34   6868   1 conhost
    506      20     1928       6268       1.69    452   0 csrss
    438      19     1944       6776       9.66    532   1 csrss
```

On Dashboard:

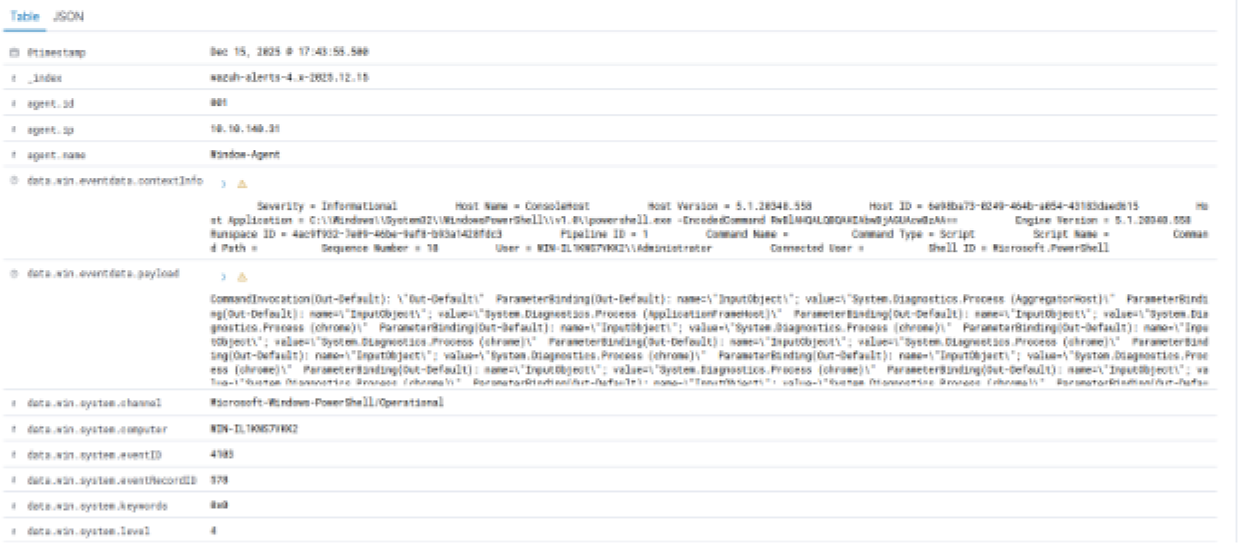


115 hits

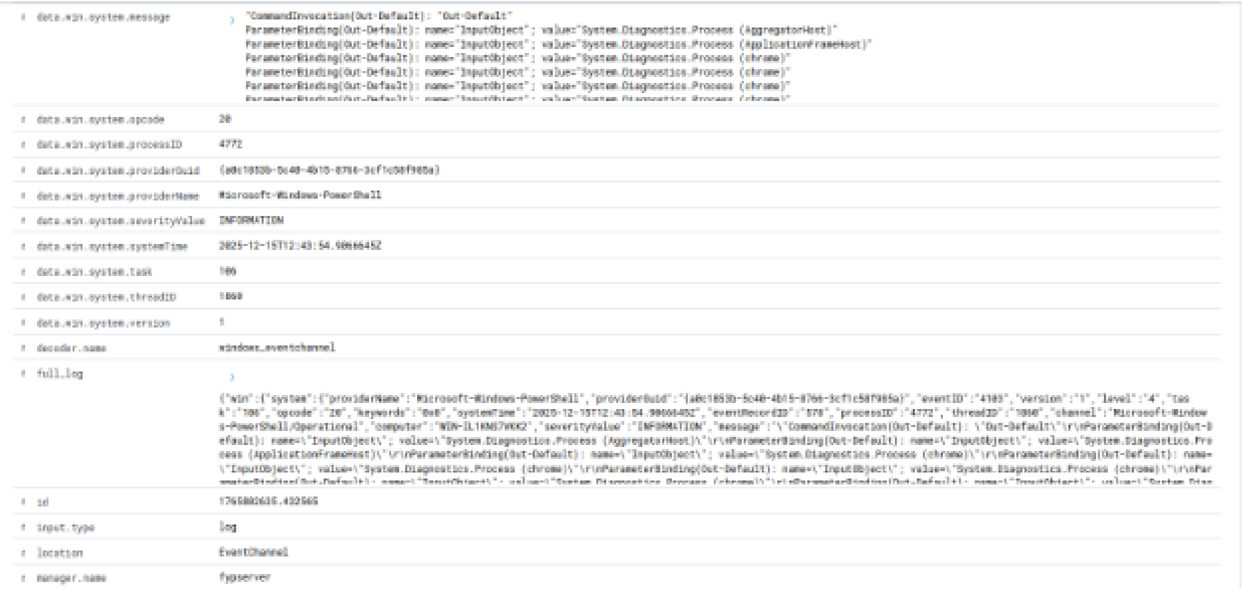
Dec 14, 2025 @ 17:44:10.816 - Dec 15, 2025 @ 17:44:10.816

timestamp	agent.name	rule.description	rule.level	rule.id
Dec 15, 2025 @ 17:43:55.5...	Window-Agent	Encoded command executed via PowerShell. 360 ForTress Multi Layer Cyber Protection System	8	100201
Dec 15, 2025 @ 17:43:55.4...	Window-Agent	Encoded command executed via PowerShell. 360 ForTress Multi Layer Cyber Protection System	8	100201
Dec 15, 2025 @ 17:43:55.4...	Window-Agent	Powershell executing process discovery	4	91815
Dec 15, 2025 @ 17:34:36.2...	Window-Agent	-	4	11
Dec 15, 2025 @ 17:31:28.9...	Window-Agent	Encoded command executed via PowerShell. 360 ForTress Multi Layer Cyber Protection System	8	100201
Dec 15, 2025 @ 17:31:28.9...	Window-Agent	Encoded command executed via PowerShell. 360 ForTress Multi Layer Cyber Protection System	8	100201

Details Alerts:



```
@timestamp                      Dec 15, 2025 @ 17:43:55.580
_index                          wazuh-alerts-4.x-2025.12.15
agent.id                        001
agent.ip                        10.10.140.31
agent.name                      Window-Agent
data.win.eventdata.contextInfo
        Severity = Informational         Host Name = ConsoleHost         Host Version = 5.1.20348.558         Host ID = 6e98ba73-0249-464b-a854-43183daed615         Ho
st Application = C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -EncodedCommand RwBlAHQALQBQAHIAbwBjAGUAcwBzAA==         Engine Version = 5.1.20348.558
Runspace ID = 4ac9f932-7e09-46be-9af8-b93a1428fdc3         Pipeline ID = 1         Command Name =         Command Type = Script         Script Name =         Comman
d Path =         Sequence Number = 18         User = WIN-IL1KN57VKK2\\Administrator         Connected User =         Shell ID = Microsoft.PowerShell
data.win.eventdata.payload
CommandInvocation(Out-Default): \"Out-Default\"  ParameterBinding(Out-Default): name=\"InputObject\"; value=\"System.Diagnostics.Process (AggregatorHost)\"  ParameterBindi
ng(Out-Default): name=\"InputObject\"; value=\"System.Diagnostics.Process (ApplicationFrameHost)\"  ParameterBinding(Out-Default): name=\"InputObject\"; value=\"System.Dia
gnostics.Process (chrome)\"  ParameterBinding(Out-Default): name=\"InputObject\"; value=\"System.Diagnostics.Process (chrome)\"  ParameterBinding(Out-Default): name=\"Inpu
tObject\"; value=\"System.Diagnostics.Process (chrome)\"  ParameterBinding(Out-Default): name=\"InputObject\"; value=\"System.Diagnostics.Process (chrome)\"  ParameterBind
ing(Out-Default): name=\"InputObject\"; value=\"System.Diagnostics.Process (chrome)\"  ParameterBinding(Out-Default): name=\"InputObject\"; value=\"System.Diagnostics.Proc
ess (chrome)\"  ParameterBinding(Out-Default): name=\"InputObject\"; value=\"System.Diagnostics.Process (chrome)\"  ParameterBinding(Out-Default): name=\"InputObject\"; va
data.win.system.channel         Microsoft-Windows-PowerShell/Operational
data.win.system.computer        WIN-IL1KN57VKK2
data.win.system.eventID         4103
data.win.system.eventRecordID   578
data.win.system.keywords        0x0
data.win.system.level           4
```



```
data.win.system.message         "CommandInvocation(Out-Default): "Out-Default"
                                ParameterBinding(Out-Default): name="InputObject"; value="System.Diagnostics.Process (AggregatorHost)"
                                ParameterBinding(Out-Default): name="InputObject"; value="System.Diagnostics.Process (ApplicationFrameHost)"
                                ParameterBinding(Out-Default): name="InputObject"; value="System.Diagnostics.Process (chrome)"
                                ParameterBinding(Out-Default): name="InputObject"; value="System.Diagnostics.Process (chrome)"
                                ParameterBinding(Out-Default): name="InputObject"; value="System.Diagnostics.Process (chrome)"
                                ParameterBinding(Out-Default): name="InputObject"; value="System.Diagnostics.Process (chrome)"
data.win.system.opcode          20
data.win.system.processID       4772
data.win.system.providerGuid    {a0c1853b-5c40-4b15-8766-3cf1c58f985a}
data.win.system.providerName    Microsoft-Windows-PowerShell
data.win.system.severityValue   INFORMATION
data.win.system.systemTime      2025-12-15T12:43:54.9866645Z
data.win.system.task            106
data.win.system.threadID        1060
data.win.system.version         1
decoder.name                    windows_eventchannel
full_log
{"win":{"system":{"providerName":"Microsoft-Windows-PowerShell","providerGuid":"{a0c1853b-5c40-4b15-8766-3cf1c58f985a}","eventID":"4103","version":"1","level":"4","tas
k":"106","opcode":"20","keywords":"0x0","systemTime":"2025-12-15T12:43:54.9866645Z","eventRecordID":"578","processID":"4772","threadID":"1060","channel":"Microsoft-Window
s-PowerShell/Operational","computer":"WIN-IL1KN57VKK2","severityValue":"INFORMATION","message":"\"CommandInvocation(Out-Default): \"Out-Default\"\r\nParameterBinding(Out-D
efault): name=\"InputObject\"; value=\"System.Diagnostics.Process (AggregatorHost)\"\r\nParameterBinding(Out-Default): name=\"InputObject\"; value=\"System.Diagnostics.Pro
cess (ApplicationFrameHost)\"\r\nParameterBinding(Out-Default): name=\"InputObject\"; value=\"System.Diagnostics.Process (chrome)\"\r\nParameterBinding(Out-Default): name=
\"InputObject\"; value=\"System.Diagnostics.Process (chrome)\"\r\nParameterBinding(Out-Default): name=\"InputObject\"; value=\"System.Diagnostics.Process (chrome)\"\r\nPar
id                              1765800635.432565
input.type                      log
location                        EventChannel
manager.name                    fypserver
```

Summary:

In this project, Windows PowerShell logging was successfully integrated with the Wazuh SIEM platform to monitor and detect suspicious activities. PowerShell Operational Logging, Script Block Logging, and Module Logging were enabled to capture normal commands, full scripts, and module usage. The Wazuh manager on Ubuntu was configured with custom rules to detect encoded and potentially malicious PowerShell commands. During testing, safe PowerShell commands, including encoded and hidden window commands, were executed, generating logs that were successfully captured by the agent and forwarded to the Wazuh Manager. The Wazuh manager then monitored these logs and successfully generated alerts for any suspicious or potentially harmful activity.