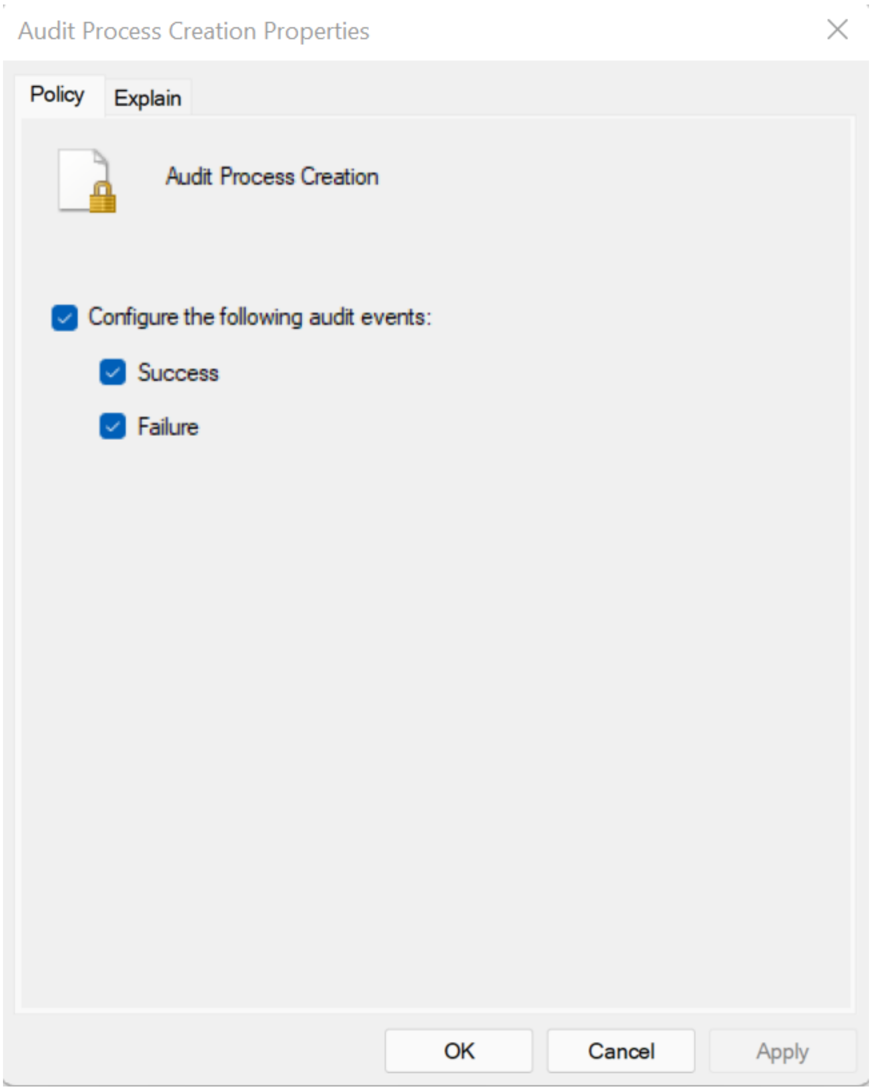
Fileless Malware

The techniques that we employed to monitor processes to detect the execution of fileless malware.

Local Group Policy Editor:

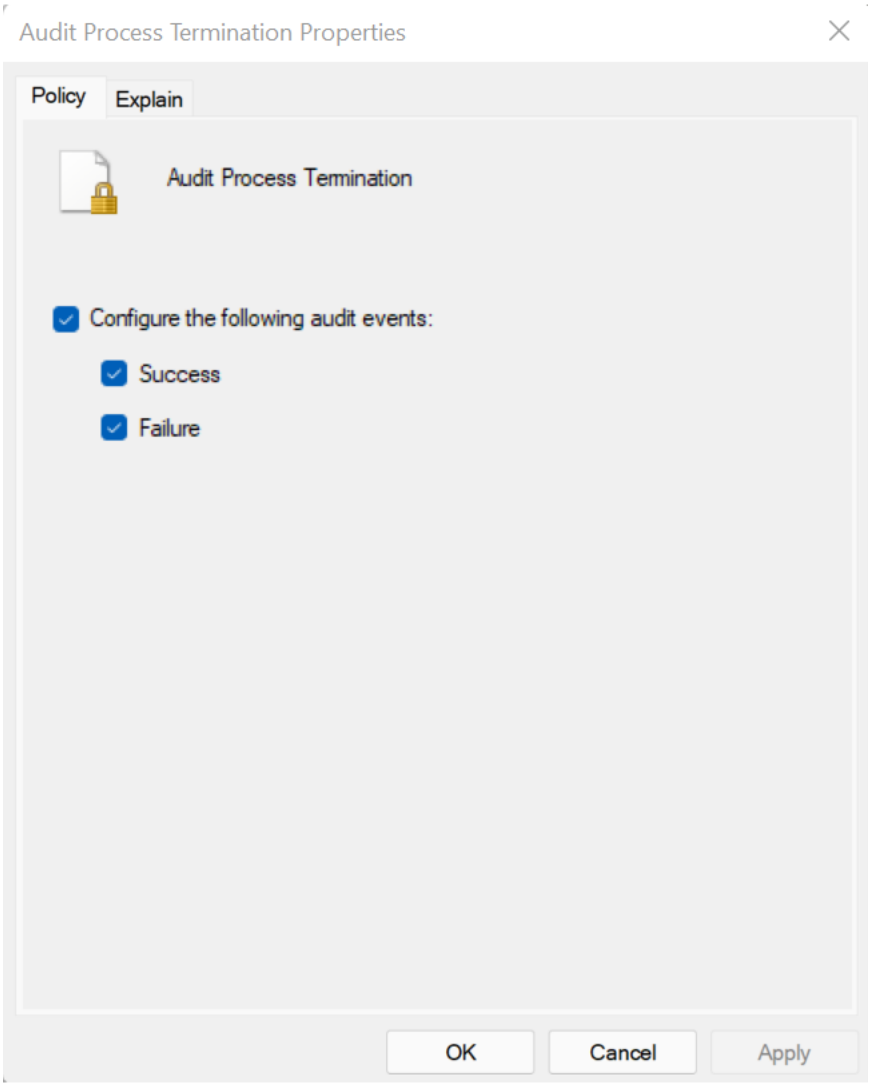
Audit Process Settings-



☑ Configure the following audit events:

☑ Success

☑ Failure



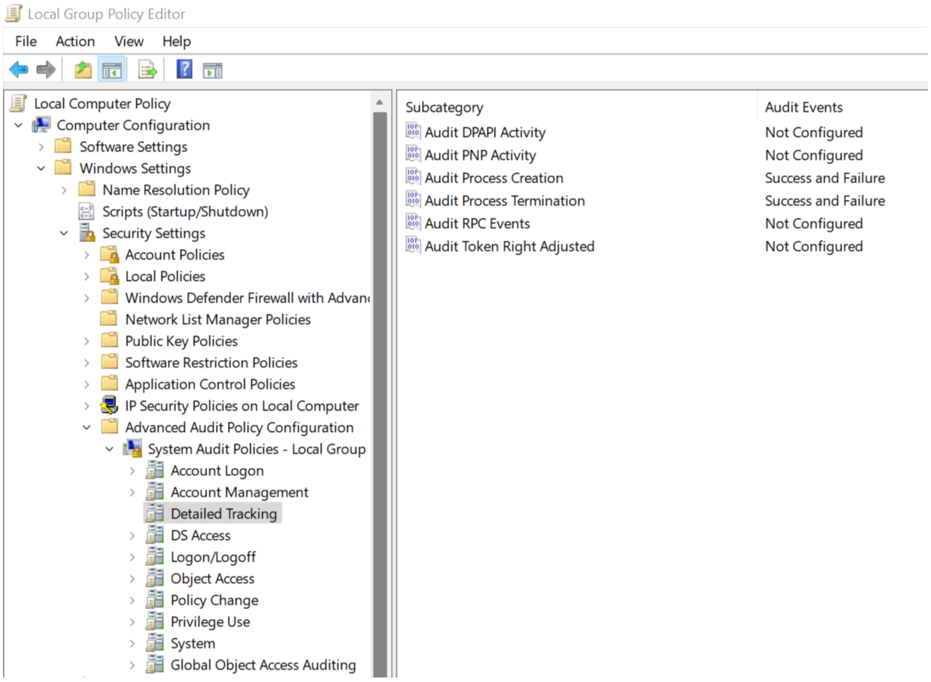
Audit Process Termination Properties

Audit Process Termination

☑ Configure the following audit events:

☑ Success

☑ Failure

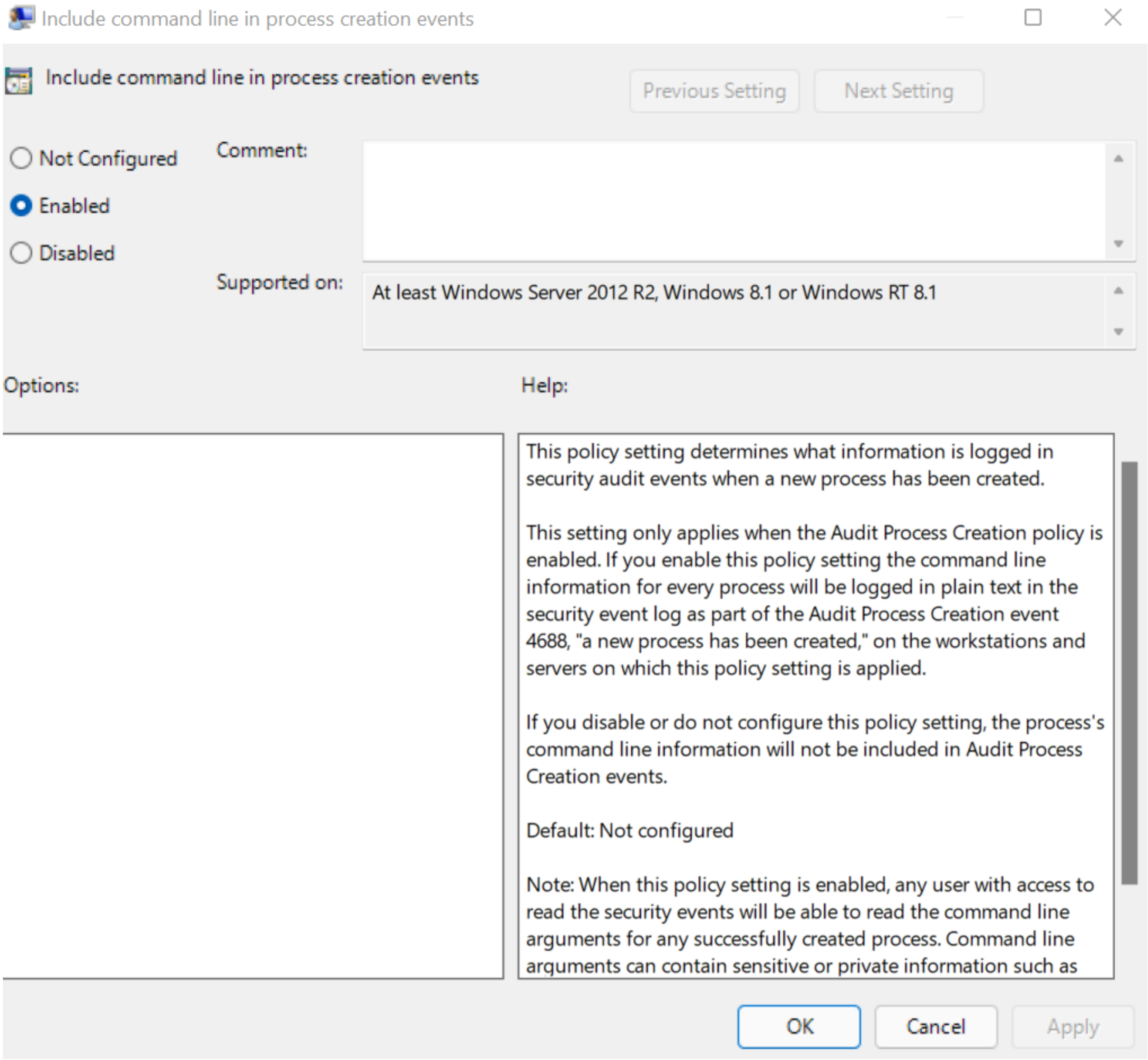


Local Group Policy Editor

Subcategory	Audit Events
Audit DPAPI Activity	Not Configured
Audit PNP Activity	Not Configured
Audit Process Creation	Success and Failure
Audit Process Termination	Success and Failure
Audit RPC Events	Not Configured
Audit Token Right Adjusted	Not Configured

We set up logging for process creation and termination.

Process creation setting-



Include command line in process creation events

Include command line in process creation events

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Windows Server 2012 R2, Windows 8.1 or Windows RT 8.1

Options:

Help:

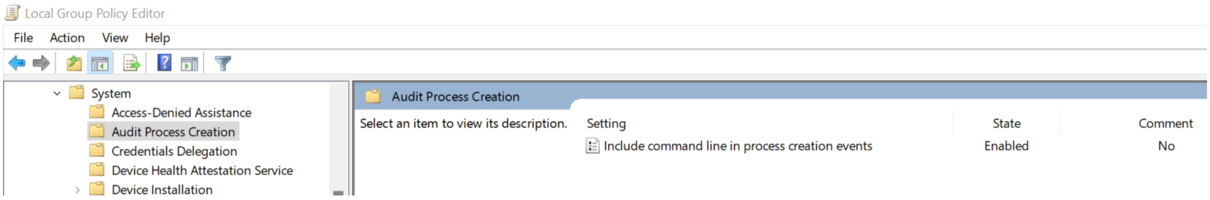
This policy setting determines what information is logged in security audit events when a new process has been created.

This setting only applies when the Audit Process Creation policy is enabled. If you enable this policy setting the command line information for every process will be logged in plain text in the security event log as part of the Audit Process Creation event 4688, "a new process has been created," on the workstations and servers on which this policy setting is applied.

If you disable or do not configure this policy setting, the process's command line information will not be included in Audit Process Creation events.

Default: Not configured

Note: When this policy setting is enabled, any user with access to read the security events will be able to read the command line arguments for any successfully created process. Command line arguments can contain sensitive or private information such as



Local Group Policy Editor

- System
  - Access-Denied Assistance
  - Audit Process Creation
  - Credentials Delegation
  - Device Health Attestation Service
  - Device Installation

Audit Process Creation

Select an item to view its description.

Setting	State	Comment
Include command line in process creation events	Enabled	No

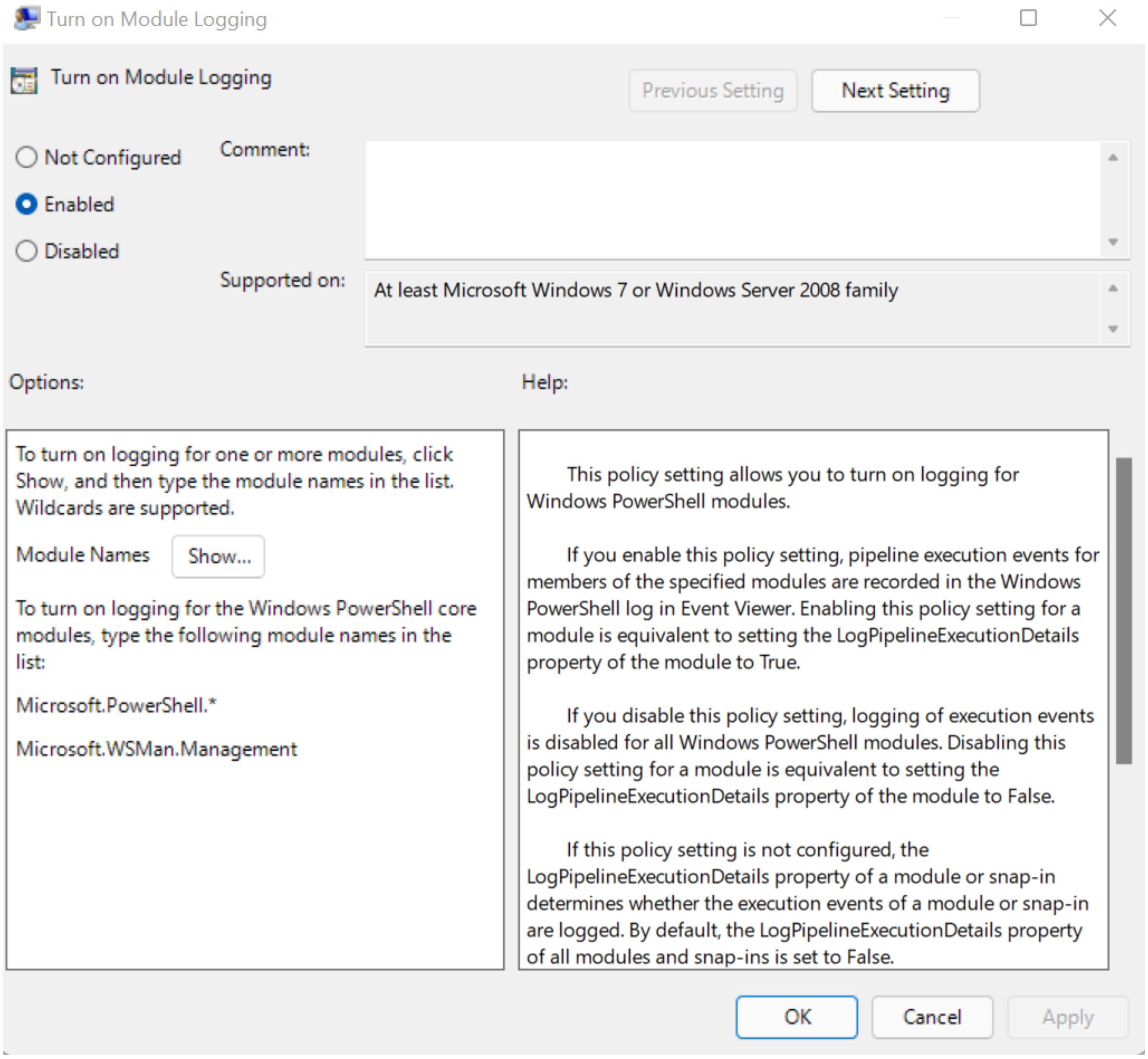
We also need to enable the events triggered through command line process creation.

Enable PowerShell Logging and Transcripts:

We then enable PowerShell specific logging to log commands executed in PowerShell.

They are:

i) Module logging
ii) Script block logging
iii) Transcript logging



Turn on Module Logging

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 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

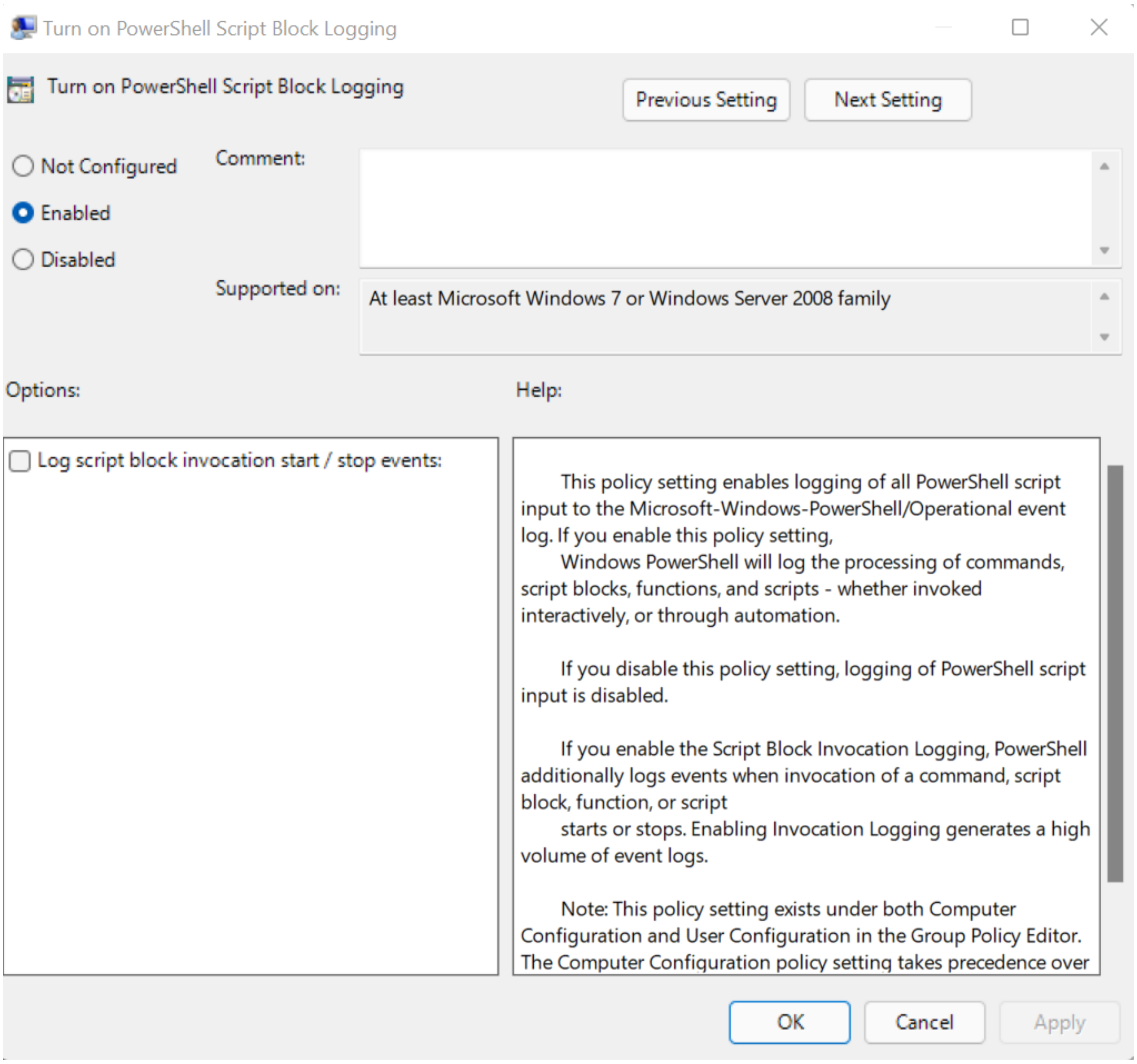
Help:

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.



Turn on PowerShell Script Block Logging

○ Not Configured
● Enabled
○ Disabled

Comment:

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

Options:

□ Log script block invocation start / stop events:

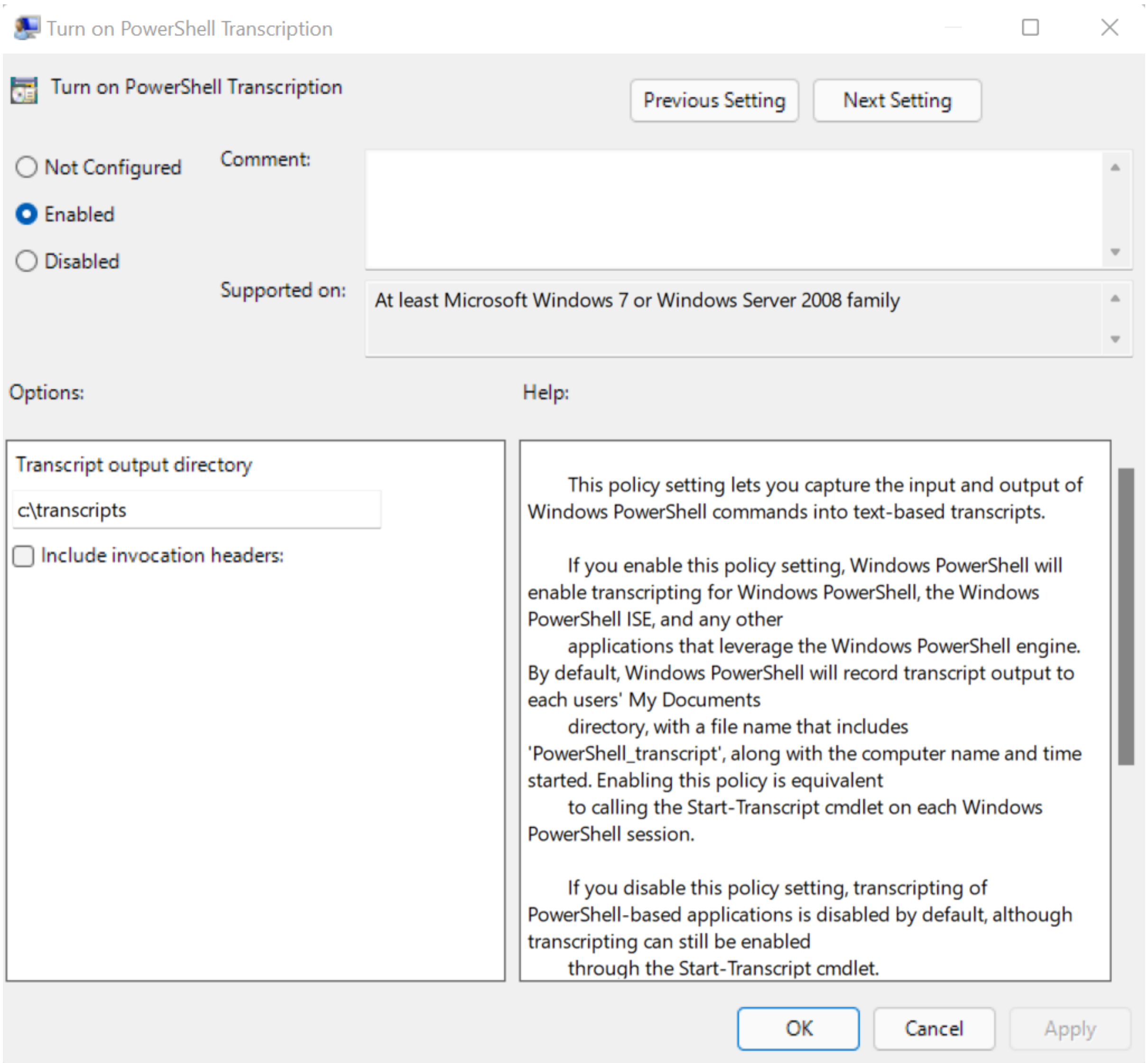
Help:

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



Turn on PowerShell Transcription

○ Not Configured

● Enabled

○ Disabled

Comment:

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

Options:

Transcript output directory

c:\transcripts

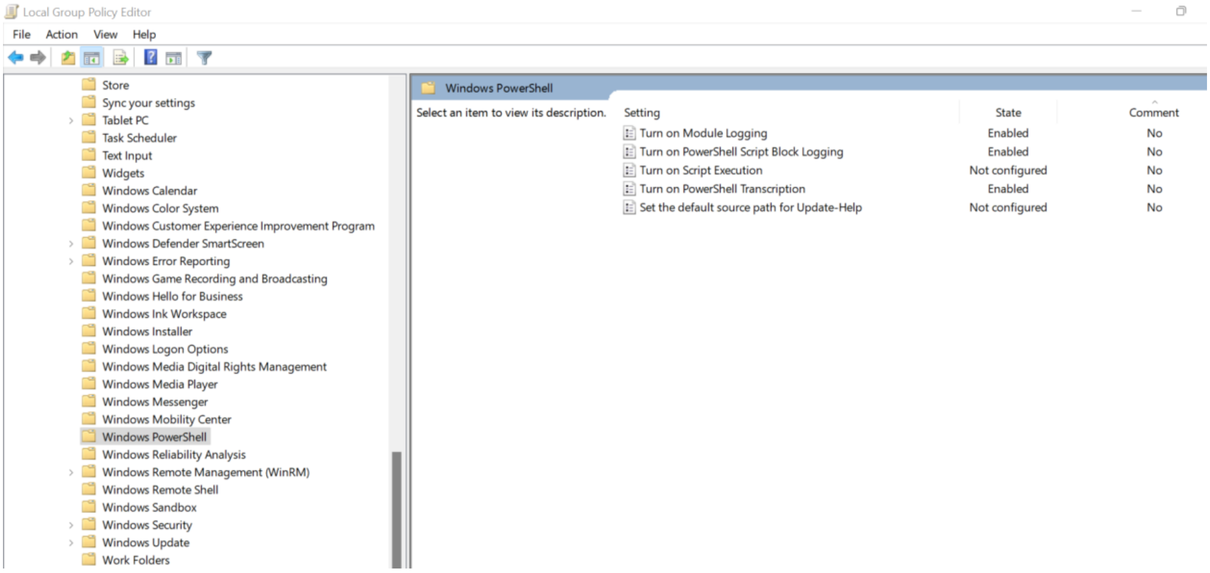
□ Include invocation headers:

Help:

This policy setting lets you capture the input and output of Windows PowerShell commands into text-based transcripts.

If you enable this policy setting, Windows PowerShell will enable transcripting for Windows PowerShell, the Windows PowerShell ISE, and any other applications that leverage the Windows PowerShell engine. By default, Windows PowerShell will record transcript output to each users' My Documents directory, with a file name that includes 'PowerShell_transcript', along with the computer name and time started. Enabling this policy is equivalent to calling the Start-Transcript cmdlet on each Windows PowerShell session.

If you disable this policy setting, transcripting of PowerShell-based applications is disabled by default, although transcripting can still be enabled through the Start-Transcript cmdlet.



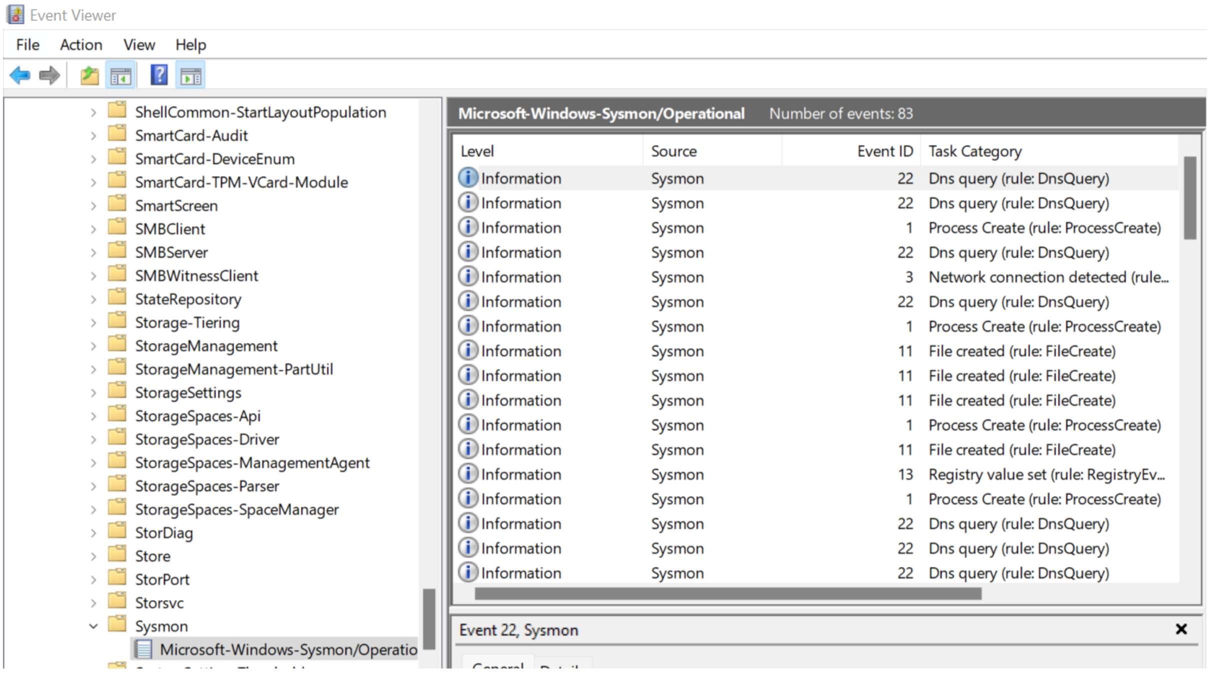
Select an item to view its description.

Setting	State	Comment
Turn on Module Logging	Enabled	No
Turn on PowerShell Script Block Logging	Enabled	No
Turn on Script Execution	Not configured	No
Turn on PowerShell Transcription	Enabled	No
Set the default source path for Update-Help	Not configured	No

Sysmon:

Upon installing Sysmon we were able to log the events triggered by the Fileless Malware that we created.

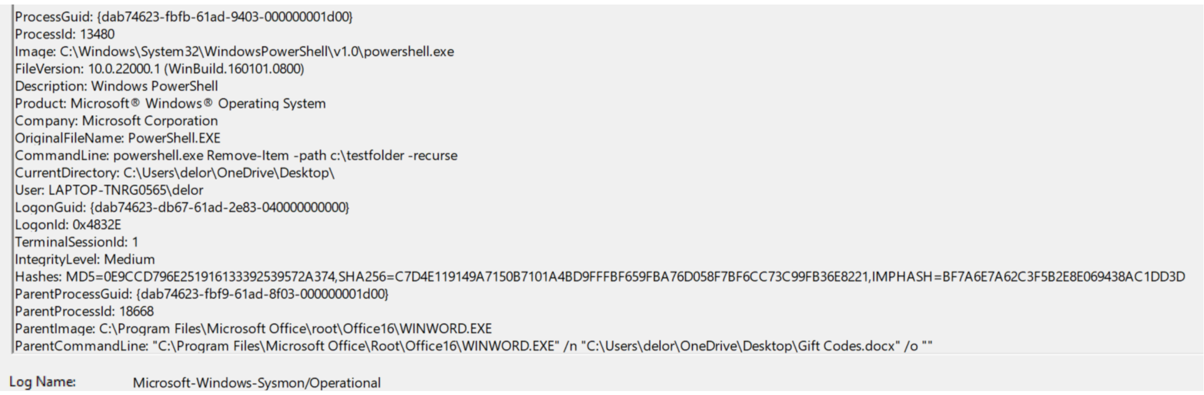
Through Event Viewer,



Microsoft-Windows-Sysmon/Operational   Number of events: 83

Level	Source	Event ID	Task Category
Information	Sysmon	22	Dns query (rule: DnsQuery)
Information	Sysmon	22	Dns query (rule: DnsQuery)
Information	Sysmon	1	Process Create (rule: ProcessCreate)
Information	Sysmon	22	Dns query (rule: DnsQuery)
Information	Sysmon	3	Network connection detected (rule...
Information	Sysmon	22	Dns query (rule: DnsQuery)
Information	Sysmon	1	Process Create (rule: ProcessCreate)
Information	Sysmon	11	File created (rule: FileCreate)
Information	Sysmon	11	File created (rule: FileCreate)
Information	Sysmon	11	File created (rule: FileCreate)
Information	Sysmon	1	Process Create (rule: ProcessCreate)
Information	Sysmon	11	File created (rule: FileCreate)
Information	Sysmon	13	Registry value set (rule: RegistryEv...
Information	Sysmon	1	Process Create (rule: ProcessCreate)
Information	Sysmon	22	Dns query (rule: DnsQuery)
Information	Sysmon	22	Dns query (rule: DnsQuery)
Information	Sysmon	22	Dns query (rule: DnsQuery)

The Event Viewer Log for the attack:



```
ProcessGuid: {dab74623-fbfb-61ad-9403-000000001d00}
ProcessId: 13480
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
FileVersion: 10.0.22000.1 (WinBuild.160101.0800)
Description: Windows PowerShell
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: PowerShell.EXE
CommandLine: powershell.exe Remove-Item -path c:\testfolder -recurse
CurrentDirectory: C:\Users\delor\OneDrive\Desktop\
User: LAPTOP-TNRG0565\delor
LogonGuid: {dab74623-db67-61ad-2e83-040000000000}
LogonId: 0x4832E
TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: MD5=0E9CCD796E251916133392539572A374,SHA256=C7D4E119149A7150B7101A4BD9FFFBF659FBA76D058F7BF6CC73C99FB36E8221,IMPHASH=BF7A6E7A62C3F5B2E8E069438AC1DD3D
ParentProcessGuid: {dab74623-fbf9-61ad-8f03-000000001d00}
ParentProcessId: 18668
ParentImage: C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE
ParentCommandLine: "C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE" /n "C:\Users\delor\OneDrive\Desktop\Gift Codes.docx" /o ""

Log Name:      Microsoft-Windows-Sysmon/Operational
```