Get-WinEvent PowerShell cmdlet Cheat Sheet

Abstract

Where to Acquire

PowerShell is natively installed in Windows Vista and newer, and includes the Get-WinEvent cmdlet by default.

Examples/Use Case

Get-WinEvent

View all events in the live system Event Log:

```
Get-WinEvent -LogName system
```

View all events in the live security Event Log (requires administrator PowerShell):

```
Get-WinEvent -LogName security
```

View all events in the file example.evtx, format list (fl) output:

~~~~
Get-WinEvent -Path example.evtx | fl
 ```
View all events in example.evtx, format GridView output:
~~~~

Get-WinEvent -Path example.evtx | Out-GridView

~~~~
Perform long tail analysis of example.evtx:
~~~~

Get-WinEvent -Path example.evtx | Group-Object id -NoElement | sort count

~~~~
Pull events 7030 and 7045 from system.evtx:
~~~~

Get-WinEvent -FilterHashtable @{Path=“system.evtx”; ID=7030,7045}

~~~~
Same as above, but use the live system event log:
~~~~

Get-WinEvent -FilterHashtable @{logname=“system”; id=7030,7045}

~~~~
Search for events containing the string "USB" in the file system.evtx:
~~~~

Get-WinEvent -FilterHashtable @{Path=“system.evtx”} | Where {$_.Message -like “*USB*”}

~~~~
'grep'-style search for lines of events containing the case insensitive string "USB" in the file system
~~~~

Get-WinEvent -FilterHashtable @{Path=“system.evtx”} | fl | findstr /i USB

~~~~
Pull all errors (level=2) from application.evtx:
~~~~

Get-WinEvent -FilterHashtable @{Path=“application.evtx”; level=2}

~~~~
Pull all errors (level=2) from application.evtx and count the number of lines ('wc'-style):
~~~~

Get-WinEvent -FilterHashtable @{Path=“application.evtx”; level=2} | Measure-Object -Line

~~~~
#### AppLocker
Pull all AppLocker logs from the live AppLocker event log (requires Applocker):
~~~~

Get-WinEvent -logname “Microsoft-Windows-AppLocker/EXE and DLL”

~~~~
Search for live AppLocker EXE/MSI block events: "(EXE) was prevented from running":
~~~~

Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Applocker/EXE and DLL"; id=8004}

`Search for live AppLocker EXE/MSI audit events: "(EXE) was allowed to run but would have been prevented`

Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Applocker/EXE and DLL"; id=8003}

```
#### EMET
Pull all EMET logs from the live Application Event log (requires EMET):
```

Get-WinEvent -FilterHashtable @{logname="application"; providername="EMET"}
`Pull all EMET logs from a saved Application Event log (requires EMET):`
Get-WinEvent -FilterHashtable @{path="application.evtx"; providername="EMET"}

```
#### Sysmon
Pull all Sysmon logs from the live Sysmon Event log (requires Sysmon and an admin PowerShell):
```

Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational"

`Pull Sysmon event ID 1 from the live Sysmon Event log`

Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Sysmon/Operational"; id=1}

```
#### Windows Defender
Pull all live Windows Defender event logs
```

Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Windows Defender/Operational"}

`Pull Windows Defender event logs 1116 and 1117 from the live event log`

Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Windows Defender/Operational";id=1116,1117}

`Pull Windows Defender event logs 1116 (malware detected) and 1117 (malware blocked) from a saved evtx f`

Get-WinEvent -FilterHashtable @{path="WindowsDefender.evtx";id=1116,1117}
"'
Additional Info
