PowerShell Get-Help on command 'Set-NetEventWFPCaptureProvider'

PS C:\Users\wahid> Get-Help Set-NetEventWFPCaptureProvider

```
NAME
    Set-NetEventWFPCaptureProvider


SYNOPSIS
    Modifies the configuration of a WFP capture provider.



SYNTAX
    Set-NetEventWFPCaptureProvider [[-Level] <Byte>] [[-MatchAnyKeyword] <UInt64>]
    [[-MatchAllKeyword] <UInt64>] [[-CaptureLayerSet] {IPv4Inbound | IPv4Outbound
    | IPv6Inbound | IPv6Outbound}] [[-IPAddresses] <String[]>] [[-TCPPorts]
    <UInt16[]>] [[-UDPPorts] <UInt16[]>] [-AsJob] [-AssociatedEventSession
    <CimInstance>] [-CimSession <CimSession[]>] [-Confirm] [-PassThru]
    [-ThrottleLimit <Int32>] [-WhatIf] [<CommonParameters>]

    Set-NetEventWFPCaptureProvider [[-Level] <Byte>] [[-MatchAnyKeyword] <UInt64>]
    [[-MatchAllKeyword] <UInt64>] [[-CaptureLayerSet] {IPv4Inbound | IPv4Outbound
    | IPv6Inbound | IPv6Outbound}] [[-IPAddresses] <String[]>] [[-TCPPorts]
    <UInt16[]>] [[-UDPPorts] <UInt16[]>] [-AsJob] [-CimSession <CimSession[]>]
    [-Confirm] -InputObject <CimInstance[]> [-PassThru] [-ThrottleLimit <Int32>]
    [-WhatIf] [<CommonParameters>]
    Set-NetEventWFPCaptureProvider [[-SessionName] <String[]>] [[-Level] <Byte>]
    [[-MatchAnyKeyword] <UInt64>] [[-MatchAllKeyword] <UInt64>]
    [[-CaptureLayerSet] {IPv4Inbound | IPv4Outbound | IPv6Inbound | IPv6Outbound}]
    [[-IPAddresses] <String[]>] [[-TCPPorts] <UInt16[]>] [[-UDPPorts] <UInt16[]>]
    [-AsJob] [-CimSession <CimSession[]>] [-Confirm] [-PassThru] [-ThrottleLimit
    <Int32>] [-WhatIf] [<CommonParameters>]
```

DESCRIPTION

The Set-NetEventWFPCaptureProvider cmdlet modifies the configuration of a Windows Firewall Platform (WFP) capture provider. For more information about the NetEventWFPCaptureProvider , see the Add-NetEventWFPCaptureProvider cmdlet.

PARAMETERS

-AsJob [<SwitchParameter>]

Runs the cmdlet as a background job. Use this parameter to run commands that take a long time to complete.

-AssociatedEventSession <CimInstance>

Specifies the associated network event session, as a CIM object. To obtain the network event session, use the Get-NetEventSession cmdlet.

-CaptureLayerSet <WFPCaptureSet>

Specifies a WFP capture set, which designates the layers and directions to filter. The acceptable values for this parameter are:

- IPv4Inbound

- IPv4Outbound

- IPv6Inbound

- IPv6Outbound

You can logically OR the direction and IP layer pairs together. For instance, you could capture incoming loopback traffic from IPv6 to avoid seeing duplicate traffic received by the loopback interface.

-CimSession <CimSession[]>

Runs the cmdlet in a remote session or on a remote computer. Enter a computer name or a session object, such as the output of a New-CimSession (https://go.microsoft.com/fwlink/p/?LinkId=227967) or [Get-CimSession](https://go.microsoft.com/fwlink/p/?LinkId=227966)cmdlet. The default is the current session on the local computer.

-Confirm [<SwitchParameter>]

Prompts you for confirmation before running the cmdlet.

-IPAddresses <String[]>

Specifies an array of IP addresses. The provider filters for and logs network traffic that matches the IP addresses that this parameter specifies. The provider joins multiple addresses by using logical OR.

-InputObject <CimInstance[]>

Specifies the input object that is used in a pipeline command.

-Level <Byte>

Specifies the Event Tracing for Windows (ETW) event error levels that NetEventWFPCaptureProvider returns. Use a level of detail specifier as a filter the type of error events that are logged. The default value for this parameter is 0x4, for informational events. The acceptable values for this parameter are:

- 0x5. Verbose - 0x4. Informational - 0x3. Warning - 0x2. Error - 0x1.
Critical - 0x0. LogAlways

The provider must log the event if the value of the event is less than or equal to the value of this parameter. Lower level events up to and including the specified level are logged.

-MatchAllKeyword <UInt64>

Specifies a keyword bitmask that restricts the events that the provider logs.

-MatchAnyKeyword <UInt64>

Specifies keywords as a set of hexadecimal values. Keywords are flags that you can combine to generate hexadecimal values that enable the provider to write one or more events for which it is instrumented, if a match is found. Use a set of hexadecimal values for keywords instead of the keyword names, and apply a filter to write ETW events for keyword matches. For more information, see EnableTraceEx2 function (https://msdn.microsoft.com/en-us/library/windows/desktop/dd392305(v=vs.85))in the Microsoft Developer Network library.

-PassThru [<SwitchParameter>]

Returns an object representing the item with which you are working. By default, this cmdlet does not generate any output.

-SessionName <String[]>

Specifies an array of session names that are associated with the NetEventWFPCaptureProvider . This parameter has the same value as the Name parameter for the New-NetEventSession cmdlet.

-TCPPorts <UInt16[]>

Specifies an array of TCP ports. The provider filters and logs network traffic that matches the ports that this parameter specifies. The provider joins multiple port numbers with logical OR.

-ThrottleLimit <Int32>

Specifies the maximum number of concurrent operations that can be established to run the cmdlet. If this parameter is omitted or a value of `0` is entered, then Windows PowerShellr calculates an optimum throttle limit for the cmdlet based on the number of CIM cmdlets that are running on the computer. The throttle limit applies only to the current cmdlet, not to the session or to the computer.

-UDPPorts <UInt16[]>

Specifies an array of UDP ports. The provider filters and logs network traffic that matches the ports that this parameter specifies. The provider joins multiple port numbers with logical OR.

-WhatIf [<SwitchParameter>]

Shows what would happen if the cmdlet runs. The cmdlet is not run.

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

----------- Example 1: Modify a WFP capture provider -----------

```
PS C:\>New-NetEventSession -Name "WFPCapture" -CaptureMode RealtimeLocal
-LocalFilePath "C:\users\DavidChew\Documents\wfpdata.etl"
PS C:\> Add-NetEventWFPCaptureProvider -SessionName "WFPCapture"
PS C:\> Set-NetEventWFPCaptureProvider -SessionName "WFPCapture"
-CaptureLayerSet IPv4Inbound, IPv6Inbound -IPAddresses "127.0.0.1", "::1"
PS C:\> Start-NetEventSession -Name "WFPCapture"
PS C:\> ping 127.0.0.1
PS C:\> ping ::1
PS C:\> Stop-NetEventSession -Name "WFPCapture"
PS C:\> Remove-NetEventSession -Name "WFPCapture"
```

The first command creates a network event session by using the New-NetEventSession cmdlet. This command also assigns the name WFPCapture to the session.

The second command uses the current cmdlet to create a WFP capture provider for the session named WFPCapture.

The third command configures the provider for capture layer directional filtering and loopback IP addresses.

The fourth command starts the session named WFPCapture.

The fifth and commands use the ping utility to test the local host address in both IPv4 and IPv6. The provider should capture both of these local test connections.

The final two commands stop the session named WFPCapture and remove it.

REMARKS

To see the examples, type: "get-help Set-NetEventWFPCaptureProvider -examples".

For more information, type: "get-help Set-NetEventWFPCaptureProvider -detailed".

For technical information, type: "get-help Set-NetEventWFPCaptureProvider -full".

For online help, type: "get-help Set-NetEventWFPCaptureProvider -online"