PowerShell Get-Help on command 'Set-NetFirewallInterfaceTypeFilter'

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

NAME

Set-NetFirewallInterfaceTypeFilter

SYNOPSIS

Modifies interface type filter objects, thereby modifying the interface type conditions of the firewall or IPsec rules.

SYNTAX

```
Set-NetFirewallInterfaceTypeFilter [-AsJob] [-CimSession <CimSession[]>]
[-Confirm] [-GPOSession <String>] [-InterfaceType {Any | Wired | Wireless |
RemoteAccess}] [-PassThru] [-PolicyStore <String>] [-ThrottleLimit <Int32>]
[-WhatIf] [<CommonParameters>]

Set-NetFirewallInterfaceTypeFilter [-AsJob] [-CimSession <CimSession[]>]
[-Confirm] -InputObject <CimInstance[]> [-InterfaceType {Any | Wired |
Wireless | RemoteAccess}] [-PassThru] [-ThrottleLimit <Int32>] [-WhatIf]
[<CommonParameters>]
```

DESCRIPTION

The Set-NetFirewallInterfaceTypeFilter cmdlet modifies the interface types associated with the input firewall or IPsec rules.

See theGet-NetFirewallInterfaceTypeFilter cmdlet for more information about the interface type filters.

To modify the interface type conditions, two methods can be used starting with the interface type filters returned by the Get-NetFirewallInterfaceTypeFilter cmdlet and optional additional querying.

- The network firewall interface type filter objects are piped into the Get-NetFirewallRule or Get-NetIPsecRule cmdlet. The Get-NetFirewallRule or Get-NetIPsecRule cmdlet returns the rules associated with the filters and pipes the rules into the Set-NetFirewallRule or Set-NetIPsecRule cmdlet, which configures the interface properties. - Alternatively, the network firewall interface type filter objects are piped directly to this cmdlet, which modifies the InterfaceType parameter value of the rules.

PARAMETERS

-AsJob [<SwitchParameter>]

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

-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.

-GPOSession <String>

Targets the network GPO from which to retrieve the rules to be modified. This parameter is used in the same way as the PolicyStore parameter. When modifying GPOs in Windows PowerShellr, each change to a GPO requires the entire GPO to be loaded, modified, and saved back. On a busy Domain Controller (DC), this can be a slow and resource-heavy operation. A GPO Session loads a domain GPO onto the local computer and makes all changes in a batch, before saving it back. This reduces the load on the DC and speeds up the Windows PowerShell cmdlets. To load a GPO Session, use the Open-NetGPO cmdlet. To save a GPO Session, use the Save-NetGPO cmdlet.

-InputObject <CimInstance[]>

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

-InterfaceType <InterfaceType>

Specifies that only network connections made through the indicated interface types are subject to the requirements of this rule. This parameter specifies different authentication requirements for each of the three main network types. The acceptable values for this parameter are: Any, Wired, Wireless, or RemoteAccess. The default value is Any.

-PassThru [<SwitchParameter>]

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

-PolicyStore <String>

Targets the policy store from which to retrieve the rules to be modified. A policy store is a container for firewall and IPsec policy. The acceptable values for this parameter are:

- PersistentStore: Sometimes called static rules, this store contains the persistent policy for the local computer. This policy is not from GPOs, and has been created manually or programmatically (during application installation) on the computer. Rules created in this store are attached to the ActiveStore and activated on the computer immediately. - ActiveStore: This store contains the currently active policy, which is the sum of all policy stores that apply to the computer. This is the resultant set of policy (RSOP) for the local computer (the sum of all GPOs that apply to the computer), and the local stores (the PersistentStore, the static Windows service hardening (WSH), and the configurable WSH). ---- GPOs are also policy stores. Computer GPOs can be specified as follows. ------ `-PolicyStore hostname`.

---- Active Directory GPOs can be specified as follows.

------ `-PolicyStore domain.fqdn.com\GPO_Friendly_Namedomain.fqdn.comGPO_Friendly_Name`.

------ Such as the following.

-------- `-PolicyStore localhost`

-------- `-PolicyStore corp.contoso.com\FirewallPolicy`

---- Active Directory GPOs can be created using the New-GPO cmdlet or the Group Policy Management Console. - RSOP: This read-only store contains the sum of all GPOs applied to the local computer.

- SystemDefaults: This read-only store contains the default state of firewall rules that ship with Windows Serverr 2012.

- StaticServiceStore: This read-only store contains all the service restrictions that ship with Windows Server 2012.

Optional and product-dependent features are considered part of Windows

Server 2012 for the purposes of WFAS. - ConfigurableServiceStore: This read-write store contains all the service restrictions that are added for third-party services. In addition, network isolation rules that are created for Windows Store application containers will appear in this policy store.

The default value is PersistentStore. The Set-NetIPsecRule cmdlet cannot be used to add an object to a policy store. An object can only be added to a policy store at creation time with the Copy-NetIPsecRule cmdlet or with the New-NetIPsecRule cmdlet.

-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.

-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 --------------------------

```
PS C:\>$nfwInterfaceTypeFilter = ( Get-FirewallRule -DisplayName "Contoso
Messenger" | Get-NetFirewallInterfaceTypeFilter )
```

```
PS C:\>Set-NetFirewallInterfaceTypeFilter -InterfaceType Any -InputObject
$nfwInterfaceTypeFilter
```

This cmdlet can be run using only the pipeline.

```
PS C:\>Get-FirewallRule -DisplayName "Contoso Messenger" |
Get-NetFirewallInterfaceTypeFilter | Set-NetFirewallInterfaceTypeFilter
-InterfaceType Any
```

This cmdlet can be run without the pipeline.

```
PS C:\>Set-NetFirewallRule -DisplayName "Contoso Messenger" -InterfaceType Any
```

This example modifies the InterfaceType parameter value for a particular firewall rule.

-------------------------- EXAMPLE 2 --------------------------

```
PS C:\>$nfwInterfaceTypeFilter = ( Get-NetFirewallInterfaceTypeFilter
-InterfaceType Wired )
```

```
PS C:\>Set-NetFirewallInterfaceTypeFilter -InterfaceType Any -InputObject
$nfwInterfaceTypeFilter
```

This cmdlet can be run using only the pipeline.

```
PS C:\>Get-NetFirewallInterfaceTypeFilter -InterfaceType Wired |
Set-NetFirewallInterfaceTypeFilter -InterfaceType Any
```

This example modifies the interface type for all of the rules previously associated with the Wired interface type.

-------------------------- EXAMPLE 3 --------------------------

```
PS C:\>$nfwInterfaceTypeFilter = ( Get-NetIPsecRule -Group DirectAccess |
Get-NetFirewallInterfaceTypeFilter )
```

```
PS C:\>$nfwInterfaceTypeFilterWired = Where-Object -Property {
$_.InterfaceType -Eq "Wired" } -InputObject $nfwInterfaceTypeFilter
```

```
PS C:\>Set-NetFirewallInterfaceTypeFilter -InterfaceType RemoteAccess
-InputObject $nfwInterfaceTypeFilterWired
```

This cmdlet can be run using only the pipeline.

```
PS C:\>Get-NetIPsecRule -Group DirectAccess |
Get-NetFirewallInterfaceTypeFilter | Where-Object -Property { $_.InterfaceType
-Eq "Wired" } | Set-NetFirewallInterfaceTypeFilter -InterfaceType RemoteAccess
```

This example modifies the interface type associated with all of the IPsec rules in a specified group.

REMARKS

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

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

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

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