MyWebUniversity
python
PowerShell Get-Help cmdlet
PS C:\> Get-Help
FPDF Library
PDF generator
FPDF
Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Set-NetFirewallAddressFilter'

PS:\>Get-HELP Set-NetFirewallAddressFilter -Full

NAME

Set-NetFirewallAddressFilter

SYNOPSIS

Modifies address filter objects, thereby modifying the local and remote address conditions of the firewall, IPsec, and main mode rules.

SYNTAX

Set-NetFirewallAddressFilter [-AsJob] [-CimSession <CimSession[]>] [-Confirm] [-GPOSession <String>] [-LocalAddress <String[]>] [-PassThru] [-PolicyStore <String>]

[-RemoteAddress <String[]>] [-ThrottleLimit <Int32>] [-WhatIf] [<CommonParameters>]

Set-NetFirewallAddressFilter [-AsJob] [-CimSession <CimSession[]>] [-Confirm] -InputObject <CimInstance[]> [-LocalAddress <String[]>] [-PassThru] [-RemoteAddress

<String[]>] [-ThrottleLimit <Int32>] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The Set-NetFirewallAddressFilter cmdlet modifies the local and remote addresses associated with the input rules

See the Get-NetFirewallAddressFilter cmdlet for more information on address filters.

To modify rule address conditions, two methods can be used starting with the address filters returned by the Get-NetFirewallAddressFilter cmdlet and optional

additional querying. The address filter objects can be piped to the Get-NetFirewallRule , Get-NetIPsecRule , or Get-NetIPsecMainModeRule cmdlet, which returns the

rule objects associated with the filters. These rules are then piped to the Set-NetFirewallRule , Set-NetIPsecRule , or Set-NetIPsecMainModeRule cmdlet, where the

address properties can be configured. Alternatively, piping the address filter objects directly to this cmdlet allows the LocalAddress and RemoteAddress parameters

of the rules to be specified.

PARAMETERS

-AsJob [<SwitchParameter>]

Runs the cmdlet as a background job. Use this parameter to run commands that take a long time to complete. The cmdlet immediately returns an object that

represents the job and then displays the command prompt. You can continue to work in the session while the job completes. To manage the job, use the `*-Job`

cmdlets. To get the job results, use the Receive-Job (https://go.microsoft.com/fwlink/?LinkID=113372)cmdlet. For more information about Windows PowerShell

background jobs, see about_Jobs (https://go.microsoft.com/fwlink/?LinkID=113251).

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-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

(/powershell/module/cimcmdlets/new-cimsession) or [Get-CimSession](https://go.microsoft.com/fwlink/p/?LinkId=227966)cmdlet. The default is the current session on the local computer.

```
        Required?                    false
        Position?                    named
        Default value                None
        Accept pipeline input?       False
        Accept wildcard characters?  false

    -Confirm [<SwitchParameter>]
        Prompts you for confirmation before running the cmdlet.

        Required?                    false
        Position?                    named
        Default value                False
        Accept pipeline input?       False
        Accept wildcard characters?  false

    -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.

```
        Required?                    false
        Position?                    named
        Default value                None
        Accept pipeline input?       False
```

Accept wildcard characters? false

-InputObject <CimInstance[]>

Specifies the input to this cmdlet. You can use this parameter, or you can pipe the input to this cmdlet.

Required?	true
Position?	named
Default value	None
Accept pipeline input?	True (ByValue)
Accept wildcard characters?	false

-LocalAddress <String[]>

Specifies that network packets with matching IP addresses match this rule. This parameter value is the first end point of an IPsec rule and specifies the

computers that are subject to the requirements of this rule. This parameter value is an IPv4 or IPv6 address, hostname, subnet, range, or the following keyword:

Any. The acceptable formats for this parameter are: - Single IPv4 Address: 1.2.3.4

- Single IPv6 Address: fe80::1

- IPv4 Subnet (by network bit count): 1.2.3.4/24

- IPv6 Subnet (by network bit count): fe80::1/48

- IPv4 Subnet (by network mask): 1.2.3.4/255.255.255.0

- IPv4 Range: 1.2.3.4 through 1.2.3.7

- IPv6 Range: fe80::1 through fe80::9

> [!NOTE] > Querying for rules with this parameter can only be performed using filter objects. See the Get-NetFirewallAddressFilter cmdlet for more information.

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-PassThru [<SwitchParameter>]

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

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-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-NetFirewallRule 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-NetFirewallRule or with the New-NetFirewallRule cmdlet.

Required?   false

Position?   named

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-RemoteAddress <String[]>

Specifies that network packets with matching IP addresses match this rule. This parameter value is the second end point of an IPsec rule and specifies the computers that are subject to the requirements of this rule. This parameter value is an IPv4 or IPv6 address, hostname, subnet, range, or the following keyword:

Any. The acceptable formats for this parameter are: - Single IPv4 Address: 1.2.3.4

- Single IPv6 Address: fe80::1

- IPv4 Subnet (by network bit count): 1.2.3.4/24

- IPv6 Subnet (by network bit count): fe80::1/48

- IPv4 Subnet (by network mask): 1.2.3.4/255.255.255.0

- IPv4 Range: 1.2.3.4 through 1.2.3.7

- IPv6 Range: fe80::1 through fe80::9

> [!NOTE] > Querying for rules with this parameter can only be performed using filter objects. See the Get-NetFirewallAddressFilter cmdlet for more information.

```
    Required?                    false
    Position?                    named
    Default value                None
    Accept pipeline input?       False
    Accept wildcard characters?  false
```

-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.

```
    Required?                    false
    Position?                    named
    Default value                None
```

Accept pipeline input?   False

Accept wildcard characters? false

-WhatIf [<SwitchParameter>]

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

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

<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).

INPUTS

Microsoft.Management.Infrastructure.CimInstance#root\StandardCimv2\MSFT_NetAddressFilter[]

The `Microsoft.Management.Infrastructure.CimInstance` object is a wrapper class that displays Windows Management Instrumentation (WMI) objects. The path after the

pound sign (`#`) provides the namespace and class name for the underlying WMI object.

OUTPUTS

Microsoft.Management.Infrastructure.CimInstance#root\StandardCimv2\MSFT_NetAddressFilter[]

The `Microsoft.Management.Infrastructure.CimInstance` object is a wrapper class that displays Windows Management Instrumentation (WMI) objects. The path after the

pound sign (`#`) provides the namespace and class name for the underlying WMI object.

NOTES

-------------------------- EXAMPLE 1 --------------------------

```
PS C:\>Set-NetIPsecRule -DisplayName "Tunnel Rule" -LocalAddress Any


# Alternatively, this task can be done with the following cmdlets.
PS C:\>$nfwAddressFilter = ( Get-NetIPsecRule -DisplayName "Tunnel Rule" | Get-NetFirewallAddressFilter )
PS C:\>Set-NetFirewallAddressFilter -InputObject $nfwAddressFilter -LocalAddress Any


# Alternatively, this task can be done with the following cmdlet.
PS C:\>Get-NetIPsecRule -DisplayName "Tunnel Rule" | Get-NetFirewallAddressFilter | Set-NetFirewallAddressFilter -LocalAddress Any
```

This example changes the first end point of a particular IPsec rule.

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

```
PS C:\>$nfwAddressFilter = ( Get-NetFirewallRule -DisplayGroup "Core Networking" | Get-NetFirewallAddressFilter )
PS C:\>$nfwAddressFilterLS6 = ( Where-Object -InputObject $nfwAddressFilter -Property { $_.RemoteAddress -Eq "LocalSubnet6" } )
PS C:\>Set-NetFirewallAddressFilter -InputObject $nfwAddressFilterLS6 -RemoteAddress LocalSubnet4


# Alternatively, this task can be done with the following cmdlet.
PS C:\>Get-NetFirewallRule -DisplayGroup "Core Networking" | Get-NetFirewallAddressFilter | Where-Object -Property { $_.RemoteAddress -Eq "LocalSubnet6" } |
Get-NetFirewallRule | Set-NetFirewallRule -RemoteAddress LocalSubnet4
```

This example gets the filter objects associated with the firewall rules with a particular remote, or second, end point belonging to the Core Networking group and
modifies the second endpoint of those rules.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/netsecurity/set-netfirewalladdressfilter?view=windowsserver2022-ps&wt.mc_id=ps-gethelp

Where-Object https://go.microsoft.com/fwlink/p/?LinkId=113423

Copy-NetIPsecRule

Get-NetFirewallAddressFilter

Get-NetFirewallRule

Get-NetIPsecMainModeRule

Get-NetIPsecRule

New-NetFirewallRule

New-NetIPsecRule

Open-NetGPO

Save-NetGPO

Set-NetFirewallRule

Get-NetIPsecMainModeRule

Set-NetIPsecRule