Windows PowerShell Get-Help on Cmdlet 'New-NetIPsecMainModeRule'

PS:\>Get-HELP New-NetIPsecMainModeRule -Full

NAME

New-NetIPsecMainModeRule

SYNOPSIS

Creates an IPsec main mode rule that tells the computer which peers require IPsec security associations (SAs) for securing network traffic, and how to negotiate those

SAs.

SYNTAX

New-NetIPsecMainModeRule [-AsJob] [-CimSession <CimSession[]>] [-Confirm] [-Description <String>] -DisplayName <String> [-Enabled {True | False}] [-GPOSession

<String>] [-Group <String>] [-LocalAddress <String[]>] [-MainModeCryptoSet <String>] [-Name <String>] [-Phase1AuthSet <String>] [-Platform <String[]>] [-PolicyStore

<String>] [-Profile {Any | Domain | Private | Public | NotApplicable}] [-RemoteAddress <String[]>] [-ThrottleLimit <Int32>] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The New-NetIPsecMainModeRule cmdlet creates an IPsec main mode rule.

A main mode rule contains a set of local and remote end points to determine the peers to which it applies. When an application on the local computer attempts to

communicate with one of these specified remote hosts, the computer attempts to establish a security association (SA) with the remote server.

In order to set up this SA, the computers need to agree on how to authenticate with each other. The local computer will only agree to use one of the proposals from

the network IPsec phase 1 authorization set associated with the main mode rule. See the New-NetIPsecPhase1AuthSet cmdlet for more information. When the negotiation is

successful a main mode SA is created. See the Get-NetIPsecMainModeSA cmdlet for more information.

The computers also need to agree on common encryption, hashing, and key exchange methods. The local computer will only agree to use one of the cryptographic methods

contained in the IPsec main mode cryptographic set associated with the main mode rule. See the New-NetIPsecMainModeCryptoSet cmdlet for more information. When the

negotiation is successful a quick mode SA is created. See the Get-NetIPsecQuickModeSA cmdlet for more information.

A main mode rule offers four mandatory parameters that negotiated as part of the main mode security association (SA): - The computer authentication method: Kerberos

v5, certificate, or pre-shared key authentication that is provided by the NetIPsecPhase1AuthSet object. - The encryption algorithm that is provided by the

NetIPsecMainModeCryptoSet object. - The hashing algorithm that is provided by the NetIPsecMainModeCryptoSet object. - The Diffie-Hellman (DH) key exchange group to

be used for the base keying material that is provided by the NetIPsecMainModeCryptoSet object.

Each main mode rule must be created in the policy store of the associated IPsec rule. If a particular rule applies to multiple IPsec rules in different policy stores

(GPOs), then the rule must be duplicated for each of those stores (so that policies can be updated without linking issues). See the Copy-NetFirewallRule,

Copy-NetIPsecMainModeCryptoSet, Copy-NetIPsecMainModeRule, Copy-NetIPsecPhase1AuthSet, Copy-NetIPsecPhase2AuthSet, Copy-NetIPsecQuickModeCryptoSet, or

Copy-NetIPsecRule cmdlet for more information.

PARAMETERS

-AsJob [<SwitchParameter>]

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

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 (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.

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

-Description <String>

Specifies that matching firewall rules of the indicated description are created. Wildcard characters are accepted. This parameter provides information about the

firewall rule. This parameter specifies the localized, user-facing description of the IPsec rule.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-DisplayName <String>

Specifies that only matching firewall rules of the indicated display name are created. Wildcard characters are accepted. Specifies the localized, user-facing

name of the firewall rule being created. When creating a rule this parameter is required. This parameter value is locale-dependent. If the object is not modified,

this parameter value may change in certain circumstances. When writing scripts in multi-lingual environments, the Name parameter should be used instead, where the

default value is a randomly assigned value. This parameter cannot be set to All.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Enabled <Enabled>

Specifies that matching main mode rules of the indicated state are created. This parameter specifies that the rule object is administratively enabled or

administratively disabled. The acceptable values for this parameter are:

- True: Specifies the rule is currently enabled.

- False: Specifies the rule is currently disabled.

A disabled rule will not actively modify computer behavior, but the rule still exists on the computer so it can be re-enabled.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-GPOSession <String>

Specifies the network GPO from which to retrieve the rules to be created. 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

-Group <String>

Specifies that only matching firewall rules of the indicated group association are created. Wildcard characters are accepted. This parameter specifies the source

string for the DisplayGroup parameter. If the DisplayGroup parameter value is a localizable string, then this parameter

contains an indirect string. Rule groups

can be used to organize rules by influence and allows batch rule modifications. Using the Set-NetFirewallRule cmdlets, if the group name is specified for a set of

rules or sets, then all of the rules or sets in that group receive the same set of modifications. It is good practice to specify this parameter value with a

universal and world-ready indirect @FirewallAPI name. The DisplayGroup parameter cannot be specified upon object creation using the New-NetFirewallRule cmdlet,

but can be modified using dot-notation and the Set-NetFirewallRule cmdlet.

Required? false

Position? named

Default value None

Accept pipeline input? False

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

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

-MainModeCryptoSet <String>

Gets the IPsec main mode rules that are associated with the given main mode cryptographic set to be created. This parameter specifies, by name, the main mode

cryptographic set to be associated with the main mode rule. A NetIPsecMainModeCryptoSet object represents a main mode cryptographic conditions associated with a

main mode rule. This parameter sets the methods for main mode negotiation by describing the proposals for encryption. This is only associated with main mode

rules. See the Get-NetIPsecMainModeCryptoSet cmdlet for more information. Alternatively, the AssociatedNetIPsecMainModeCryptoSet parameter can be used for the

same purpose, but is used to pipe the input set into the rule.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Name <String>

Specifies that only matching main mode cryptographic sets of the indicated name are created. Wildcard characters are accepted. This parameter acts just like a

filename, in that only one rule with a given name may exist in a policy store at a time. During group policy processing and policy merge, rules that have the same

name but come from multiple stores being merged, will overwrite one another so that only one exists. This overwriting

behavior is desirable if the rules serve the

same purpose. For instance, all of the firewall rules have specific names, so if an administrator can copy these rules to a GPO, and the rules will override the

local versions on a local computer. GPOs can have precedence. So, if an administrator has a different or more specific rule the same name in a higher-precedence

GPO, then it overrides other rules that exist. The default value is a randomly assigned value. When you want to override the defaults for main mode encryption,

specify the customized parameters and set this parameter value, making this parameter the new default setting for encryption.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Phase1AuthSet <String>

Gets the main mode rules that are associated with the given phase 1 authentication set to be created. This parameter specifies, by Name, the Phase 1

authentication set to be associated with the main mode rule. A NetIPsecPhase1AuthSet object represents the phase 1 authentication conditions associated with an

IPsec or main mode rule. This parameter sets the methods for main mode negotiation by describing the proposals for computer authentication. See the

New-NetIPsecAuthProposal cmdlet of more information. Alternatively, the AssociatedNetIPsecPhase1AuthSet parameter can be used for the same purpose, but is used to

pipe the input set into the rule.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Platform <String[]>

Specifies which version of Windows the associated rule applies. The acceptable format for this parameter is a number in the Major.Minor format. The version

number of 6.0 corresponds to Vista (nextref_vista), 6.1 corresponds to Win7 (Windowsr 7 or firstref_longhorn), and 6.2 corresponds to Win8 (Windowsr 8 or Windows

Server 2012). If + is not specified, then only that version is associated. If + is specified, then that version and later are associated. Querying for rules

with this parameter with the Get-NetIPsecMainModeRule cmdlet cannot be performed.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-PolicyStore <String>

Specifies the policy store from which to retrieve the sets to be created. 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-NetIPsecMainModeCryptoSet 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-NetIPsecMainModeCryptoSet cmdlet or with this cmdlet.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Profile <Profile>

Specifies one or more profiles to which the rule is assigned. The rule is active on the local computer only when the specified profile is currently active. This

relationship is many-to-many and can be indirectly modified by the user, by changing the Profiles field on instances of firewall rules. Only one profile is

applied at a time. The acceptable values for this parameter are: Any, Domain, Private, Public, or NotApplicable. The default value is Any. Separate multiple

entries with a comma and do not include any spaces. Use the keyword Any to configure the profile as Private, Public, Domain in the ConfigurableServiceStore.

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

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

None

OUTPUTS

Microsoft.Management.Infrastructure.CimInstance#root\StandardCimv2\NetIPsec

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.

Microsoft.Management.Infrastructure.CimInstance#root\StandardCimv2\AssociatedNetIPsecMainModeCryptoSet

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:\>$proposal1 = (New-NetIPsecMainModeCryptoProposal -Encryption DES3 -Hash MD5 -KeyExchange DH1)

PS C:\>$proposal2 = (New-NetIPsecMainModeCryptoProposal -Encryption AES192 -Hash MD5 -KeyExchange DH14)

PS C:\>$proposal3 = (New-NetIPsecMainModeCryptoProposal -Encryption DES3 -Hash MD5 -KeyExchange DH19)

PS C:\>$mmCryptoSet = New-NetIPsecMainModeCryptoSet -DisplayName "Main Mode Crypto Set" -Proposal $proposal1,$proposal2,$proposal3

PS C:\>New-NetIPsecMainModeRule -DisplayName "Custom Main Mode Rule" -MainModeCryptoSet $mmCryptoSet.Name

This example creates a main mode rule linked to a cryptographic set that contains three cryptographic proposals.

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

PS C:\>$cert1Proposal = New-NetIPsecAuthProposal -Machine -Cert -Authority "C=US,O=MSFT,CN=Microsoft Root Authority" -AuthorityType Root

PS C:\>$cert2Proposal = New-NetIPsecAuthProposal -Machine -Cert -Authority "C=US,O=MYORG,CN='My Organizations Root Certificate'" -AuthorityType Root

PS C:\>$certAuthSet = New-NetIPsecPhase1AuthSet -DisplayName "Computer Certificate Auth Set" -Proposal $Cert1Proposal,$cert2Proposal

PS C:\>New-NetIPsecMainModeRule -DisplayName "Main Mode Authenticate with Certificates Rule" -Phase1AuthSet $certAuthSet.Name

This example creates a main mode rule that requires that incoming connections are authenticated by using either of two computer certificates.

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

PS C:\>$proposal1 = New-NetIPsecAuthProposal -Machine -Cert -Authority "C=US,O=MSFT,CN=Microsoft Root Authority" -AuthorityType Root

PS C:\>$poAuthSet = New-NetIPsecPhase1AuthSet -DisplayName "Computer Certificate Auth Set" -Proposal $proposal1

PS C:\>$proposal2 = New-NetIPsecMainModeCryptoProposal -Encryption DES3 -Hash MD5 -KeyExchange DH1

PS C:\>$mmCryptoSet = New-NetIPsecMainModeCryptoSet -DisplayName "dhgroup2:3des-sha256,3des-sha384" -Proposal $proposal2

PS C:\>New-NetIPsecMainModeRule -DisplayName "Alternate Main Mode Rule" -LocalAddress Any -RemoteAddress 192.168.0.5 -Phase1AuthSet $poAuthSet.Name -MainModeCryptoSet
$mmCryptoSet.Name

This example creates a main mode rule that specifies using alternate authentication and security methods for clients that communicate with the server at the IP
address 192.168.0.5 only.

RELATED LINKS

Online Version:

https://learn.microsoft.com/powershell/module/netsecurity/new-netipsecmainmoderule?view=windowsserver2022-ps&wt.mc_id=ps-gethelp

Get-NetFirewallAddressFilter

Get-NetIPsecMainModeCryptoSet

Get-NetIPsecMainModeRule

Get-NetIPsecMainModeSA

Get-NetIPsecQuickModeSA

New-NetFirewallRule

New-NetIPsecMainModeCryptoSet

New-NetIPsecPhase1AuthSet

Open-NetGPO

Save-NetGPO

Set-NetFirewallRule

Set-NetIPsecMainModeCryptoSet

New-NetIPsecAuthProposal

New-NetIPsecMainModeCryptoProposal