



python



PowerShell

FPDF Library

PDF generator

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Sync-NetIPsecRule'

PS C:\Users\wahid> Get-Help Sync-NetIPsecRule

NAME

Sync-NetIPsecRule

SYNOPSIS

Gets the list of IP addresses to be added and deleted to an IPsec rule based on the differences detected between the existing rule IP addresses and the specified IP addresses.

SYNTAX

```
Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-All] [-AsJob]
[-CimSession <CimSession[]>] [-Confirm] [-DnsServers <String[]>] [-Domains
<String[]>] [-EndpointType {Endpoint1 | Endpoint2}] [-GPOSession <String>]
[-PolicyStore <String>] [-Servers <String[]>] [-ThrottleLimit <Int32>]
[-TracePolicyStore] [-WhatIf] [<CommonParameters>]
```

```
Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AllowSetKey
<Boolean[]>] [-AllowWatchKey <Boolean[]>] [-AsJob] [-CimSession
<CimSession[]>] [-Confirm] [-Description <String[]>] [-DisplayGroup
<String[]>] [-DnsServers <String[]>] [-Domains <String[]>] [-Enabled {True |
False}] [-EncryptedTunnelBypass <Boolean[]>] [-EndpointType {Endpoint1 |
```

Endpoint2}] [-ForwardPathLifetime <UInt32[]>] [-GPOSession <String>] [-Group <String[]>] [-InboundSecurity {None | Request | Require}] [-KeyModule {Default | IKEv1 | AuthIP | IKEv2}] [-Machine <String[]>] [-Mode {None | Tunnel | Transport}] [-OutboundSecurity {None | Request | Require}] [-Phase1AuthSet <String[]>] [-Phase2AuthSet <String[]>] [-PolicyStore <String>] [-PolicyStoreSource <String[]>] [-PolicyStoreSourceType {None | Local | GroupPolicy | Dynamic | Generated | Hardcoded}] [-PrimaryStatus {Unknown | OK | Inactive | Error}] [-QuickModeCryptoSet <String[]>] [-RemoteTunnelHostname <String[]>] [-RequireAuthorization <Boolean[]>] [-Servers <String[]>] [-Status <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore] [-User <String[]>] [-WhatIf] [<CommonParameters>]

Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AsJob] -AssociatedNetFirewallAddressFilter <CimInstance> [-CimSession <CimSession[]>] [-Confirm] [-DnsServers <String[]>] [-Domains <String[]>] [-EndpointType {Endpoint1 | Endpoint2}] [-GPOSession <String>] [-PolicyStore <String>] [-Servers <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore] [-WhatIf] [<CommonParameters>]

Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AsJob] -AssociatedNetFirewallInterfaceFilter <CimInstance> [-CimSession <CimSession[]>] [-Confirm] [-DnsServers <String[]>] [-Domains <String[]>] [-EndpointType {Endpoint1 | Endpoint2}] [-GPOSession <String>] [-PolicyStore <String>] [-Servers <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore] [-WhatIf] [<CommonParameters>]

Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AsJob] -AssociatedNetFirewallInterfaceTypeFilter <CimInstance> [-CimSession <CimSession[]>] [-Confirm] [-DnsServers <String[]>] [-Domains <String[]>] [-EndpointType {Endpoint1 | Endpoint2}] [-GPOSession <String>] [-PolicyStore <String>] [-Servers <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore] [-WhatIf] [<CommonParameters>]

Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AsJob]
-AssociatedNetFirewallPortFilter <CimInstance> [-CimSession <CimSession[]>]
[-Confirm] [-DnsServers <String[]>] [-Domains <String[]>] [-EndpointType
{Endpoint1 | Endpoint2}] [-GPOSession <String>] [-PolicyStore <String>]
[-Servers <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore] [-WhatIf]
[<CommonParameters>]

Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AsJob]
-AssociatedNetFirewallProfile <CimInstance> [-CimSession <CimSession[]>]
[-Confirm] [-DnsServers <String[]>] [-Domains <String[]>] [-EndpointType
{Endpoint1 | Endpoint2}] [-GPOSession <String>] [-PolicyStore <String>]
[-Servers <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore] [-WhatIf]
[<CommonParameters>]

Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AsJob]
-AssociatedNetIPsecPhase1AuthSet <CimInstance> [-CimSession <CimSession[]>]
[-Confirm] [-DnsServers <String[]>] [-Domains <String[]>] [-EndpointType
{Endpoint1 | Endpoint2}] [-GPOSession <String>] [-PolicyStore <String>]
[-Servers <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore] [-WhatIf]
[<CommonParameters>]

Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AsJob]
-AssociatedNetIPsecPhase2AuthSet <CimInstance> [-CimSession <CimSession[]>]
[-Confirm] [-DnsServers <String[]>] [-Domains <String[]>] [-EndpointType
{Endpoint1 | Endpoint2}] [-GPOSession <String>] [-PolicyStore <String>]
[-Servers <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore] [-WhatIf]
[<CommonParameters>]

Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AsJob]
-AssociatedNetIPsecQuickModeCryptoSet <CimInstance> [-CimSession
<CimSession[]>] [-Confirm] [-DnsServers <String[]>] [-Domains <String[]>]
[-EndpointType {Endpoint1 | Endpoint2}] [-GPOSession <String>] [-PolicyStore
<String>] [-Servers <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore]

[-WhatIf] [<CommonParameters>]

Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AsJob] [-CimSession <CimSession[]>] [-Confirm] -DisplayName <String[]> [-DnsServers <String[]>] [-Domains <String[]>] [-EndpointType {Endpoint1 | Endpoint2}] [-GPOSession <String>] [-PolicyStore <String>] [-Servers <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore] [-WhatIf] [<CommonParameters>]

Sync-NetIPsecRule [-IPsecRuleName] <String[]> [-AddressType {None | IPv4 | IPv6}] [-AsJob] [-CimSession <CimSession[]>] [-Confirm] [-DnsServers <String[]>] [-Domains <String[]>] [-EndpointType {Endpoint1 | Endpoint2}] [-GPOSession <String>] [-PolicyStore <String>] [-Servers <String[]>] [-ThrottleLimit <Int32>] [-TracePolicyStore] [-WhatIf] [<CommonParameters>]

Sync-NetIPsecRule [-AddressType {None | IPv4 | IPv6}] [-AsJob] [-CimSession <CimSession[]>] [-Confirm] [-DnsServers <String[]>] [-Domains <String[]>] [-EndpointType {Endpoint1 | Endpoint2}] -InputObject <CimInstance[]> [-Servers <String[]>] [-ThrottleLimit <Int32>] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The Sync-NetIPsecRule cmdlet detects changes in IPsec addresses retrieved from current IPsec addresses and input values, returns the addresses, and then updates the IPsec rule end points.

The first tunnel policy is defined by IP addresses that are derived from domain names and servers. Running this cmdlet resolves the IP addresses for the DirectAccess (DA) first tunnel and updates the Group Policy Objects (GPOs) appropriately. The specified DNS server, using the DnsServers parameter, will be used to resolve the domain name and server names.

A list of IP addresses is retrieved based on the derived values from input parameters like the Domains and Servers parameters. This cmdlet will output

delta collection objects and the associated actions: to Add or Delete the change in IP addresses, the actual list of changes detected, and a list of fully qualified domain names (FQDNs) that did not resolve. When there are multiple rules that match the same name, the cmdlet fails with an error.

This parameter updates on per-rule basis with greater flexibility in rule selection or querying. Rules can be obtained using parameter values including IPsecRuleName (default), DisplayName , rule properties, or by associated NetFirewall filters or NetIPsec objects. The resultant specified end point of the queried rule is immediately updated.

PARAMETERS

-AddressType <AddressVersion>

Specifies the type of addresses that will be used for making IP address comparisons. The acceptable values for this parameter are: None, IPv4, or IPv6. This parameter is mandatory.

-All [<SwitchParameter>]

Indicates that all of the IPsec rules within the specified policy store are synchronized.

-AllowSetKey <Boolean[]>

Indicates that matching IPsec rules of the indicated value are synchronized. This parameter specifies that the IPsec rule allows trusted intermediaries to override keying material. If this parameter is set to True, then the trusted intermediaries are allowed to manipulate the cryptographic keying material used with an IPsec security association (SA). It is possible that when this parameter is True at both ends, the computer will perform arbitration through SA negotiation so that one end sets the key while the other end watches the key. See the AllowWatchKey parameter for more information. The default value is False. This parameter is supported on Windows Server 2012.

-AllowWatchKey <Boolean[]>

Indicates that matching IPsec rules of the indicated value are synchronized. This parameter specifies that the IPsec rule allows trusted intermediaries to provide notification of changes in keying material. If this parameter is set to True, then the trusted intermediaries are allowed to retrieve the cryptographic keying material associated with an IPsec security association (SA), and to subscribe for notification of changes. The default value is False. This parameter is supported on Windows Server 2012.

-AsJob [<SwitchParameter>]

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

-AssociatedNetFirewallAddressFilter <CimInstance>

Gets only the IPsec rules that are associated with the given address filter to be synchronized. A NetFirewallAddressFilter object represents the address conditions associated with a rule. See the Get-NetFirewallAddressFilter cmdlet for more information.

-AssociatedNetFirewallInterfaceFilter <CimInstance>

Gets the IPsec rules that are associated with the given interface filter to be synchronized. A NetFirewallInterfaceFilter object represents the interface conditions associated with a rule. See the Get-NetFirewallInterfaceFilter cmdlet for more information.

-AssociatedNetFirewallInterfaceTypeFilter <CimInstance>

Gets the IPsec rules that are associated with the given interface type filter to be synchronized. A NetFirewallInterfaceTypeFilter object represents the interface conditions associated with a rule. See the Get-NetFirewallInterfaceTypeFilter cmdlet for more information.

-AssociatedNetFirewallPortFilter <CimInstance>

Gets the IPsec rules that are associated with the given port filter to be synchronized. A NetFirewallPortFilter object represents the port conditions associated with a rule. See the Get-NetFirewallPortFilter cmdlet for more information.

-AssociatedNetFirewallProfile <CimInstance>

Gets the IPsec rules that are associated with the given firewall profile type to be synchronized. A NetFirewallProfile object represents the profile conditions associated with a rule. See the Get-NetFirewallProfile cmdlet for more information.

-AssociatedNetIPsecPhase1AuthSet <CimInstance>

Gets the IPsec rules that are associated with the given phase 1 authentication set to be synchronized. A NetIPsecPhase1AuthSet object represents the phase 1 authorization set 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 Get-NetIPsecPhase1AuthSet cmdlet for more information. Alternatively, the Phase1AuthSet parameter can be used for the same purpose, but does not allow the authentication set to be piped into the cmdlet and the set must be specified with the IPsecRuleName parameter.

-AssociatedNetIPsecPhase2AuthSet <CimInstance>

Gets the IPsec rules that are associated, via the pipeline, with the input phase 2 authentication set to be synchronized. A NetIPsecPhase1AuthSet object represents the phase 2 authorization set conditions associated with a rule. See the Get-NetIPsecPhase2AuthSet cmdlet for more information. Alternatively, the Phase2AuthSet parameter can be used for the same purpose, but does not allow the authentication set to be piped into the cmdlet and the set must be specified with the IPsecRuleName parameter.

-AssociatedNetIPsecQuickModeCryptoSet <CimInstance>

Gets the IPsec rules that are associated, via the pipeline, with the input quick mode cryptographic set to be synchronized. A NetIPsecQuickModeCryptoSet object represents a quick mode cryptographic set that contains cryptographic proposals. This parameter specifies parameters for the quick mode negotiation as well as dictating the cryptographic proposals that should be proposed during the exchange. This is only associated with IPsec rules. See the Get-NetIPsecQuickModeCryptoSet cmdlet for more information. Alternatively, the QuickModeCryptoSet parameter can be used for the same purpose, but does not allow the cryptographic set to be piped into the cmdlet and the set must be specified with the IPsecRuleName parameter.

-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.

-Description <String[]>

Specifies that matching IPsec rules of the indicated description are synchronized. Wildcard characters are accepted. This parameter provides information about the IPsec rule. This parameter specifies a localized, user-facing description of the object.

-DisplayGroup <String[]>

Specifies that only matching firewall rules of the indicated group association are synchronized. Wildcard characters are accepted. The Group parameter specifies the source string for this parameter. If the value for this parameter is a localizable string, then the Group parameter contains

an indirect string. Rule groups can be used to organize rules by influence and allows batch rule modifications. Using the Set-NetIPsecRule cmdlet, 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 the Group parameter value with a universal and world-ready indirect @FirewallAPI name. This parameter cannot be specified upon object creation using the New-NetIPsecRule cmdlet, but can be modified using dot-notation and the Set-NetIPsecRule cmdlet.

-DisplayName <String[]>

Specifies that only matching IPsec rules of the indicated display name are synchronized. Wildcard characters are accepted. Specifies the localized, user-facing name of the IPsec 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 IPsecRuleName parameter should be used instead, where the default value is a randomly assigned value. This parameter cannot be set to All.

-DnsServers <String[]>

Specifies a list of DNS server IP addresses that will be used for name resolution. This parameter accepts one or more DNS server IP addresses. If this parameter is not specified, then this cmdlet uses the default DNS servers.

-Domains <String[]>

Gets all of the IP addresses that are associated with the list of domains by specifying an array of fully qualified domain names (FQDN).

-Enabled <Enabled[]>

Specifies that matching IPsec rules of the indicated state are synchronized. 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 management construct still exists on the computer so it can be re-enabled.

-EncryptedTunnelBypass <Boolean[]>

Indicates that matching IPsec rules of the specified value are synchronized. This parameter specifies the encapsulation state for network traffic sent to a tunnel end point that is already IPsec protected. If this parameter is set to True, then the network traffic sent to a tunnel end point that is already IPsec protected does not have to be encapsulated again. This option can improve network performance in the case where network traffic that is already end-to-end protected by other IPsec rules. The default value is False. This parameter is only supported on firstref_server_7 and Windows Server 2012.

-EndpointType <EndpointType>

Specifies that the local or remote endpoint should be modified by adding or removing IP addresses. The acceptable values for this parameter are: Endpoint1 or Endpoint2. Endpoint1 or Endpoint2 corresponds to the local address or remote address for the IPsec rule.

-ForwardPathLifetime <UInt32[]>

Specifies that matching IPsec rules of the specified path lifetime value are synchronized. This parameter specifies the session key lifetime for an IPsec rule, in minutes. The acceptable values for this parameter are: 78 through 172799. The default value is 0 minutes. When managing a GPO, the default setting is NotConfigured. This parameter is supported on Windows Server 2012.

-GPOSession <String>

Specifies the network GPO from which to retrieve the rules to be synchronized. This parameter is used in the same way as the PolicyStore parameter. When modifying GPOs in Windows PowerShell, 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.

-Group <String[]>

Specifies that only matching IPsec rules of the indicated group association are synchronized. 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. The rule groups can be used to organize rules by influence and allows batch rule modifications. Using the Set-NetIPsecRule 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 a 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-NetIPsecRule cmdlet, but can be modified using dot-notation and the Set-NetIPsecRule cmdlet.

-IPsecRuleName <String[]>

Specifies that only matching IPsec rules of the indicated name are synchronized. Wildcard characters are accepted. This parameter acts just like a file name, 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 with the same name in a higher-precedence GPO, then it overrides other rules that exist. The default value is a randomly assigned value. When the defaults for main mode encryption need to be overridden, specify the customized parameters and set this parameter, making it the new default setting for encryption.

-InboundSecurity <SecurityPolicy[]>

Specifies that matching IPsec rules of the indicated security policy are synchronized. This parameter determines the degree of enforcement for security on inbound traffic. The acceptable values for this parameter are:

- None: No authentication is requested or required for connections that match the rule. It specifies that the local computer does not attempt authentication for any network connections that match this rule. This option is typically used to grant IPsec exemptions for network connections that do not need to be protected by IPsec, but would otherwise match other rules that could cause the connection to be dropped. - Request:

Authentication is requested for connections that match the rule. The local computer attempts to authenticate any inbound network connections that match this rule, but allows the connection if the authentication attempt is not successful. - Require: Authentication is required for connections that match the rule. If the authentication is not successful, then the

inbound network traffic is discarded. The default value is None. When the OutboundSecurity parameter is also specified, the following configurations are valid: InboundSecurity \ OutboundSecurity = None\None, Request\None, Request\Request, Require\Request, or Require\Require.

-InputObject <CimInstance[]>

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

-KeyModule <KeyModule[]>

Specifies that matching IPsec rules of the indicated key module are synchronized. This parameter specifies which keying modules to negotiate.

The acceptable values for this parameter are: Default, AuthIP, IKEv1, or IKEv2.

- Default: Equivalent to both IKEv1 and AuthIP. Required in order for the rule to be applied to computers running Windows versions prior to nextref_server_7. ---- There are authentication and cryptographic methods that are only compatible with certain keying modules. This is a very advanced setting intended only for specific interoperability scenarios. Overriding this parameter value may result in traffic being sent in plain-text if the authorization and cryptographic settings are not supported by the keying modules there. - AuthIP: Supported with phase 2 authentication.

- IKEv1: Supported with pre-shared key (PSK), Certificates, and Kerberos.

- IKEv2: Not supported with Kerberos, PSK, or NTLM.

Windows versions prior to Windows Server 2012 only support the Default configuration.

-Machine <String[]>

Specifies that matching IPsec rules of the indicated computer accounts are synchronized. This parameter specifies that only network packets that are authenticated as incoming from or outgoing to a computer identified in the list of computer accounts (SID) match this rule. This parameter value is specified as an SDDL string.

-Mode <IPsecMode[]>

Specifies that matching IPsec rules of the indicated mode are synchronized. This parameter specifies the type of IPsec mode connection that the IPsec rule defines. The acceptable values for this parameter are: None, Transport, or Tunnel. The default value is Transport.

-OutboundSecurity <SecurityPolicy[]>

Specifies that matching IPsec rules of the indicated security policy are synchronized. This parameter determines the degree of enforcement for security on outbound traffic. The acceptable values for this parameter are:

- None: No authentication is requested or required for connections that match the rule. It specifies that the local computer does not attempt authentication for any network connections that match this rule. This option is typically used to grant IPsec exemptions for network connections that do not need to be protected by IPsec, but would otherwise match other rules that could cause the connection to be dropped. - Request:

Authentication is requested for connections that match the rule. The local computer attempts to authenticate any outbound network connections that match this rule, but allows the connection if the authentication attempt fails. - Require: Authentication is required for connections that match the rule. If the authentication is not successful, then the outbound network traffic is discarded. The default value is None. When the

InboundSecurity parameter is also specified, the following configurations are valid: InboundSecurity / OutboundSecurity = None\None, Request\None, Request\Request, Require\Request, or Require\Require.

-Phase1AuthSet <String[]>

Gets the IPsec rules that are associated with the given phase 1 authentication set to be synchronized. A NetIPsecPhase1AuthSet object represents the phase 1 authorization set 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 Get-NetIPsecPhase1AuthSet cmdlet for more information. Alternatively, this parameter can be used for the same purpose, but does not allow the authentication set to be piped into the cmdlet and the set must be specified with the IPsecRuleName parameter.

`-Phase2AuthSet <String[]>`

Gets the IPsec rules that are associated with the given phase 2 authentication set to be synchronized. A NetIPsecPhase2AuthSet object represents the phase 2 authorization set 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 Get-NetIPsecPhase2AuthSet cmdlet for more information. Alternatively, this parameter can be used for the same purpose, but does not allow the authentication set to be piped into the cmdlet and the set must be specified with the IPsecRuleName parameter.

`-PolicyStore <String>`

Specifies the policy store from which to retrieve the rules to be synchronized. 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 Server 2012.

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

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.

-PolicyStoreSource <String[]>

Specifies that IPsec rules matching the indicated policy store source are synchronized. This parameter contains a path to the policy store where the rule originated if the object is retrieved from the ActiveStore with the TracePolicyStoreSource option set. This parameter value is automatically generated and should not be modified. The monitoring output from this parameter is not completely compatible with the PolicyStore parameter. This parameter value cannot always be passed into the PolicyStore parameter. Domain GPOs are one example in which this parameter contains only the GPO name, not the domain name.

-PolicyStoreSourceType <PolicyStoreType[]>

Specifies that IPsec rules that match the indicated policy store source type are synchronized. This parameter describes the type of policy store where the rule originated if the object is retrieved from the ActiveStore with the TracePolicyStoreSource option set. This parameter value is automatically generated and should not be modified. The acceptable values for this parameter are:

- Local: The object originates from the local store.
- GroupPolicy: The object originates from a GPO.
- Dynamic: The object originates from the local runtime state.

This policy store name is not valid for use in the cmdlets, but may appear when monitoring active policy. - Generated: The object was generated automatically. This policy store name is not valid for use in the cmdlets, but may appear when monitoring active policy. - Hardcoded: The object was hard-coded. This policy store name is not valid for use in the cmdlets, but may appear when monitoring active policy.

-PrimaryStatus <PrimaryStatus[]>

Specifies that IPsec rules that match the indicated primary status are synchronized. This parameter specifies the overall status of the rule. -

OK: Specifies that the rule will work as specified.

- Degraded: Specifies that one or more parts of the rule will not be enforced.

- Error: Specifies that the computer is unable to use the rule at all.

See the Status and StatusCode fields of the object for more detailed status information.

`-QuickModeCryptoSet <String[]>`

Specifies that matching IPsec rules of the specified quick mode cryptographic set are synchronized. This parameter specifies, by name, the quick mode cryptographic that is associated with the IPsec rule. A NetIPsecMainModeCryptoSet object represents quick mode cryptographic conditions associated with an IPsec rule. This parameter sets the methods for quick mode negotiation by describing the proposals for encryption. See the New-NetIPsecQuickModeCryptoSet cmdlet for more information. Alternatively, the AssociatedNetIPsecQuickModeCryptoSet parameter can be used for the same purpose, but is used to pipe the input set into the rule. When specifying cryptographic sets, the IPsecRuleName parameter value of the cryptographic set. The object cannot be directly passed to this cmdlet.

`-RemoteTunnelHostname <String[]>`

Specifies that matching IPsec rules of the specified second end point tunnel host name are synchronized. Specifies a fully qualified DNS name that resolves to a list of remote tunnel end points. This parameter is supported on Windows Server 2012. This parameter can only be used with multiple remote tunnel end points. Specifying this parameter prevents a non-asymmetric tunnel mode IPsec rule from being created. Rule creation

will fail when a single remote tunnel end point and this parameter are specified, or when remote tunnel end point is Any and this parameter is specified.

-RequireAuthorization <Boolean[]>

Indicates that matching IPsec rules of the specified value are synchronized. Specifies the given value for an IPsec rule. If this parameter is set to True, then enforcement of authorization is allowed for end points. This parameter is supported on nextref_server_7 and Windows Server 2012.

-Servers <String[]>

Gets all of the IP addresses that are associated with the list of domains by specifying an array of fully qualified domain names (FQDN). Specifies the list of servers (using FQDNs) associated with the IPsec rule.

-Status <String[]>

Specifies that IPsec rules that match the indicated status are synchronized. This parameter describes the status message for the specified status code value. The status code is a numerical value that indicates any syntax, parsing, or runtime errors in the rule or set. This parameter value should not be modified.

-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 PowerShell 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.

-TracePolicyStore [<SwitchParameter>]

Specifies that the name of the source GPO is queried and set to the

PolicyStoreSource parameter value.

-User <String[]>

Specifies that matching IPsec rules of the indicated user accounts are synchronized. This parameter specifies that only network packets that are authenticated as incoming from or outgoing to a user identified in the list of user accounts match this rule. This parameter value is specified as an SDDL string.

-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:\>$serverPolicyStore = domain.contoso.com/server_GPO
```

```
PS C:\>$serverRuleName = "Any-Traffic-Win8DA-Rule"
```

```
PS C:\>$domains = "corp.contoso.com", "corp.contoso2.com"
```

```
PS C:\>$servers = "server2.corp.contoso.com"
```

```
PS C:\>$primaryDns64 = 1.2.2.1
```

```
PS C:\>Sync-NetIPsecRule -PolicyStore $serverPolicyStore -IPsecRuleName  
$serverRuleName -EndpointType endpoint1 -Domains $domains -Servers $servers  
-DnsServers $primaryDns64 -AddressType IPv6 -Confirm
```

This example gets the list of IP addresses that need to be added and deleted to an IPsec rule based on the differences detected between the existing rule IP addresses and the IP addresses derived from the input parameters, and then makes the updates. Specify the Confirm parameter to see which rules are being updated.

REMARKS

To see the examples, type: "get-help Sync-NetIPsecRule -examples".

For more information, type: "get-help Sync-NetIPsecRule -detailed".

For technical information, type: "get-help Sync-NetIPsecRule -full".

For online help, type: "get-help Sync-NetIPsecRule -online"