# n

# nac-authentication-server-group (Deprecated)

To identify the group of authentication servers to be used for Network Admission Control posture validation, use the **nac-authentication-server-group** command in tunnel-group general-attributes configuration mode. To inherit the authentication server group from the default remote access group, access the alternative group policy from which to inherit it, then use the **no** form of this command.

**nac-authentication-server-group** *server-group*
**no nac-authentication-server-group**

## Syntax Description

| | |
|---|---|
| *server-group* | Name of the posture validation server group, as configured on the ASA using the **aaa-server host** command. The name must match the server-tag variable specified in that command. |

## Command Default

This command has no arguments or keywords.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Tunnel-group general-attributes configuration | • Yes | • — | • Yes | • — | — |

## Command History

| Release | Modification |
|---|---|
| 7.2(1) | This command was added. |
| 8.0(1) | This command was deprecated. The **authentication-server-group** command in nac-policy-nac-framework configuration mode replaced it. |

## Usage Guidelines

Configure at least one Access Control Server to support NAC. Use the **aaa-server** command to name the ACS group. Then use the **nac-authentication-server-group** command, using the same name for the server group.

## Examples

The following example identifies acs-group1 as the authentication server group to be used for NAC posture validation:

```
ciscoasa(config-group-policy)# nac-authentication-server-group acs-group1
ciscoasa(config-group-policy)
```

The following example inherits the authentication server group from the default remote access group.

```
ciscoasa(config-group-policy)# no nac-authentication-server-group
ciscoasa(config-group-policy)
```

**Related Commands**

| Command | Description |
|---|---|
| **aaa-server** | Creates a record of the AAA server or group and sets the host-specific AAA server attributes. |
| debug eap | Enables logging of EAP events to debug NAC messaging. |
| debug eou | Enables logging of EAP over UDP (EAPoUDP) events to debug NAC messaging. |
| debug nac | Enables logging of NAC events. |
| **nac** | Enables Network Admission Control on a group policy. |

# nac-policy (Deprecated)

**Note** The last supported release for this command was Version 9.1(1).

To create or access a Cisco Network Admission Control (NAC) policy, and specify its type, use the **nac-policy** command in global configuration mode. To remove the NAC policy from the configuration, use the **no** form of this command.

**nac-policy** *nac-policy-name* **nac-framework**
**no nac-policy** *nac-policy-name* **nac-framework**

## Syntax Description

| | |
|---|---|
| *nac-policy-name* | Name of the NAC policy. Enter a string of up to 64 characters to name the NAC policy. The **show running-config nac-policy** command displays the name and configuration of each NAC policy already present on the security appliance. |
| **nac-framework** | Specifies the use of a NAC framework to provide a network access policy for remote hosts. A Cisco Access Control Server must be present on the network to provide NAC Framework services for the ASA.<br><br>If you specify this type, the prompt indicates you are in config--nac-policy-nac-framework configuration mode. This mode lets you configure the NAC Framework policy. |

## Command Default

This command has no default settings.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Global Configuration | • Yes | • — | • Yes | • — | — |

## Command History

| Release | Modification |
|---|---|
| 8.0(2) | This command was added. |
| 9.1(2) | This command was deprecated. |

## Usage Guidelines

Use this command once for each NAC Appliance to be assigned to a group policy. Then use the **nac-settings** command to assign the NAC policy to each applicable group policy. Upon the setup of an IPsec or Cisco AnyConnect VPN tunnel, the ASA applies the NAC policy associated with the group policy in use.

You cannot use the **no nac-policy** *name* command to remove a NAC policy if it is already assigned to one or more group policies.

## Examples

The following command creates and accesses a NAC Framework policy named nac-framework1:

```
ciscoasa
(config)
# nac-policy nac-framework1 nac-framework
ciscoasa
(config-nac-policy-nac-framework)
```

The following command removes the NAC Framework policy named nac-framework1:

```
ciscoasa
(config)
# no nac-policy nac-framework1
ciscoasa
(config-nac-policy-nac-framework)
```

## Related Commands

| Command | Description |
|---|---|
| **show running-config nac-policy** | Displays the configuration of each NAC policy on the ASA. |
| **show nac-policy** | Displays NAC policy usage statistics on the ASA. |
| **clear nac-policy** | Resets the NAC policy usage statistics. |
| **nac-settings** | Assigns a NAC policy to a group policy. |
| **clear configure nac-policy** | Removes all NAC policies from the running configuration except for those that are assigned to group policies. |

# nac-settings (Deprecated)

**Note** The last supported release for this command was Version 9.1(1).

To assign a NAC policy to a group policy, use the **nac-settings** command in group-policy configuration mode, as follows:

**nac-settings** { **value** *nac-policy-name* | **none** }
**no nac-settings** { **value** *nac-policy-name* | **none** }

## Syntax Description

| | |
|---|---|
| *nac-policy-name* | NAC policy to be assigned to the group policy. The NAC policy you name must be present in the configuration of the ASA. The **show running-config nac-policy** command displays the name and configuration of each NAC policy. |
| **none** | Removes the *nac-policy-name* from the group policy and disables the use of a NAC policy for this group policy. The group policy does not inherit the nac-settings value from the default group policy. |
| **value** | Assigns the NAC policy to be named to the group policy. |

## Command Default

This command has no arguments or keywords.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Group-policy configuration | • Yes | • — | • Yes | • — | — |

## Command History

| Release | Modification |
|---|---|
| 8.0(2) | This command was added. |
| 9.1(2) | This command was deprecated. |

## Usage Guidelines

Use the **nac-policy** command to specify the name and type of the NAC policy, then use this command to assign it to a group policy.

The **show running-config nac-policy** command displays the name and configuration of each NAC policy.

The ASA automatically enables NAC for a group policy when you assign a NAC policy to it.

## Examples

The following command removes the *nac-policy-name* from the group policy. The group policy inherits the *nac-settings* value from the default group policy:

```
ciscoasa(config-group-policy)
# no nac-settings
ciscoasa(config-group-policy)
```

The following command removes the *nac-policy-name* from the group policy and disables the use of a NAC policy for this group policy. The group policy does not inherit the nac-settings value from the default group policy.

```
ciscoasa(config-group-policy)
# nac-settings none
ciscoasa(config-group-policy)
```

## Related Commands

| Command | Description |
|---|---|
| **nac-policy** | Creates and accesses a Cisco NAC policy, and specifies its type. |
| **show running-config nac-policy** | Displays the configuration of each NAC policy on the ASA. |
| **show nac-policy** | Displays NAC policy usage statistics on the ASA. |
| **show vpn-session_summary.db** | Displays the number IPsec, WebVPN, and NAC sessions. |
| **show vpn-session.db** | Displays information about VPN sessions, including NAC results. |

# name (dynamic-filter blacklist or whitelist)

To add a domain name to the Botnet Traffic Filter blacklist or whitelist, use the **name** command in dynamic-filter blacklist or whitelist configuration mode. To remove the name, use the **no** form of this command. The static database lets you augment the dynamic database with domain names or IP addresses that you want to whitelist or blacklist.

**name** *domain_name*
**no name** *domain_name*

## Syntax Description

| | |
|---|---|
| *domain_name* | Adds a name to the blacklist. You can enter this command multiple times for multiple entries. You can add up to 1000 blacklist entries. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Dynamic-filter blacklist or whitelist configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 8.2(1) | This command was added. |

## Usage Guidelines

After you enter the dynamic-filter whitelist or blacklist configuration mode, you can manually enter domain names or IP addresses (host or subnet) that you want to tag as good names in a whitelist or bad names in a blacklist using the **address** and **name** commands.

You can enter this command multiple times for multiple entries. You can add up to 1000 blacklist and 1000 whitelist entries.

When you add a domain name to the static database, the ASA waits 1 minute, and then sends a DNS request for that domain name and adds the domain name/IP address pairing to the *DNS host cache* . (This action is a background process, and does not affect your ability to continue configuring the ASA).

If you do not have a domain name server configured for the ASA, or it is unavailable, then you can alternatively enable DNS packet inspection with Botnet Traffic Filter snooping (see the **inspect dns dynamic-filter-snooping** command). With DNS snooping, when an infected host sends a DNS request for a name on the static database, the ASA looks inside the DNS packets for the domain name and associated IP address and adds the name and IP address to the DNS reverse lookup cache. See the **inspect dns dynamic-filter-snooping** command for information about the DNS reverse lookup cache.

Entries in the DNS host cache have a time to live (TTL) value provided by the DNS server. The largest TTL value allowed is 1 day (24 hours); if the DNS server provides a larger TTL, it is truncated to 1 day maximum.

For the DNS host cache, after an entry times out, the ASA periodically requests a refresh for the entry.

## Examples

The following example creates entries for the blacklist and whitelist:

```
ciscoasa(config)# dynamic-filter blacklist
ciscoasa(config-llist)# name bad1.example.com
ciscoasa(config-llist)# name bad2.example.com
ciscoasa(config-llist)# address 10.1.1.1 255.255.255.0
ciscoasa(config-llist)# dynamic-filter whitelist
ciscoasa(config-llist)# name good.example.com
ciscoasa(config-llist)# name great.example.com
ciscoasa(config-llist)# name awesome.example.com
ciscoasa(config-llist)# address 10.1.1.2
 255.255.255.255
```

## Related Commands

| Command | Description |
|---|---|
| **address** | Adds an IP address to the blacklist or whitelist. |
| clear configure dynamic-filter | Clears the running Botnet Traffic Filter configuration. |
| **clear dynamic-filter dns-snoop** | Clears Botnet Traffic Filter DNS snooping data. |
| **clear dynamic-filter reports** | Clears Botnet Traffic filter report data. |
| **clear dynamic-filter statistics** | Clears Botnet Traffic filter statistics. |
| dns domain-lookup | Enables the ASA to send DNS requests to a DNS server to perform a name lookup for supported commands. |
| dns server-group | Identifies a DNS server for the ASA. |
| **dynamic-filter blacklist** | Edits the Botnet Traffic Filter blacklist. |
| dynamic-filter database fetch | Manually retrieves the Botnet Traffic Filter dynamic database. |
| **dynamic-filter database find** | Searches the dynamic database for a domain name or IP address. |
| dynamic-filter database purge | Manually deletes the Botnet Traffic Filter dynamic database. |
| **dynamic-filter enable** | Enables the Botnet Traffic Filter for a class of traffic or for all traffic if you do not specify an access list. |
| **dynamic-filter updater-client enable** | Enables downloading of the dynamic database. |
| **dynamic-filter use-database** | Enables use of the dynamic database. |
| **dynamic-filter whitelist** | Edits the Botnet Traffic Filter whitelist. |
| **inspect dns dynamic-filter-snoop** | Enables DNS inspection with Botnet Traffic Filter snooping. |
| **name** | Adds a name to the blacklist or whitelist. |

| Command | Description |
|---|---|
| **show asp table dynamic-filter** | Shows the Botnet Traffic Filter rules that are installed in the accelerated security path. |
| **show dynamic-filter data** | Shows information about the dynamic database, including when the dynamic database was last downloaded, the version of the database, how many entries the database contains, and 10 sample entries. |
| **show dynamic-filter dns-snoop** | Shows the Botnet Traffic Filter DNS snooping summary, or with the **detail** keyword, the actual IP addresses and names. |
| **show dynamic-filter reports** | Generates reports of the top 10 Botnet sites, ports, and infected hosts. |
| **show dynamic-filter statistics** | Shows how many connections were monitored with the Botnet Traffic Filter, and how many of those connections match the whitelist, blacklist, and greylist. |
| **show dynamic-filter updater-client** | Shows information about the updater server, including the server IP address, the next time the ASA will connect with the server, and the database version last installed. |
| show running-config dynamic-filter | Shows the Botnet Traffic Filter running configuration. |

# name (global)

To associate a name with an IP address, use the **name** command in global configuration mode. To disable the use of the text names but not remove them from the configuration, use the **no** form of this command.

**name** *ip_address* [ *name* [ **description** *text* ] ]
**no name** *ip_address* [ *name* [ **description** *text* ] ]

## Syntax Description

| | |
|---|---|
| **description** | (Optional) Specifies a description for the ip address name. |
| *ip_address* | Specifies an IP address of the host that is named. |
| *name* | Specifies the name assigned to the IP address. Use characters a to z, A to Z, 0 to 9, a dash, and an underscore. The *name* must be 63 characters or less. Also, the *name* cannot start with a number. |
| text | Specifies the text for the description. |

## Command Default

No default behaviors or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Global Configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |
| 7.0(4) | This command was enhanced to include an optional description. |
| 8.3(1) | You can no longer use a named IP address in a **nat** command or an **access-list** command; you must use **object network** names instead. Although **network-object** commands in an object group accept **object network** names, you can still also use a named IP address identified by the **name** command. |

## Usage Guidelines

To enable the association of a name with an IP address, use the **names** command. You can associate only one name with an IP address.

You must first use the **names** command before you use the **name** command. Use the name command immediately after you use the names command and before you use the **write memory** command.

The **name** command lets you identify a host by a text name and map text strings to IP addresses. The **no name** command allows you to disable the use of the text names but does not remove them from the configuration. Use the **clear configure name** command to clear the list of names from the configuration.

To disable displaying **name** values, use the **no names** command.

Both the name and names commands are saved in the configuration.

The name command does not support assigning a name to a network mask. For example, this command would be rejected:

```
ciscoasa(config)# name 255.255.255.0 class-C-mask
```

**Note** None of the commands in which a mask is required can process a name as an accepted network mask.

## Examples

This example shows that the **names** command allows you to enable use of the **name** command. The **name** command substitutes **sa_inside** for references to 192.168.42.3 and **sa_outside** for 209.165.201.3. You can use these names with the **ip address** commands when assigning IP addresses to the network interfaces. The **no names** command disables the **name** command values from displaying. Subsequent use of the **names** command again restores the **name** command value display.

```
ciscoasa(config)# names
ciscoasa(config)# name 192.168.42.3 sa_inside
ciscoasa(config)# name 209.165.201.3 sa_outside
ciscoasa(config-if)# ip address inside sa_inside 255.255.255.0
ciscoasa(config-if)# ip address outside sa_outside 255.255.255.224
ciscoasa(config)# show ip address
System IP Addresses:
inside ip address sa_inside mask 255.255.255.0
outside ip address sa_outside mask 255.255.255.224
ciscoasa(config)# no names
ciscoasa(config)# show ip address
System IP Addresses:
inside ip address 192.168.42.3 mask 255.255.255.0
outside ip address 209.165.201.3 mask 255.255.255.224
ciscoasa(config)# names
ciscoasa(config)# show ip address
System IP Addresses:
inside ip address sa_inside mask 255.255.255.0
outside ip address sa_outside mask 255.255.255.224
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure name** | Clears the list of names from the configuration. |
| **names** | Enables the association of a name with an IP address. |
| **show running-config name** | Displays the names associated with an IP address. |

# nameif

To provide a name for an interface, use the **nameif** command in interface configuration mode. To remove the name, use the **no** form of this command. The interface name is used in all configuration commands on the ASA instead of the interface type and ID (such as gigabitethernet0/1), and is therefore required before traffic can pass through the interface.

**nameif** *name*
**no nameif**

## Syntax Description

*name* Sets a name up to 48 characters in length. The name is not case-sensitive. Do not use the names "Metrics_History" or "MH"; they cause ASDM to show the interface in a down state.

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Interface Configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was changed from a global configuration command to an interface configuration mode command. |

## Usage Guidelines

For subinterfaces, you must assign a VLAN with the **vlan** command before you enter the **nameif** command.

You can change the name by reentering this command with a new value. Do not enter the **no** form, because that command causes all commands that refer to that name to be deleted.

## Examples

The following example configures the names for two interfaces to be "inside" and "outside:"

```
ciscoasa(config)# interface gigabitethernet0/1
ciscoasa(config-if)# nameif inside
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0
ciscoasa(config-if)# no shutdown
ciscoasa(config-if)# interface gigabitethernet0/0
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# security-level 0
ciscoasa(config-if)# ip address 10.1.2.1 255.255.255.0
ciscoasa(config-if)# no shutdown
```

**Related Commands**

| Command | Description |
|---|---|
| **clear xlate** | Resets all translations for existing connections, causing the connections to be reset. |
| **interface** | Configures an interface and enters interface configuration mode. |
| **security-level** | Sets the security level for the interface. |
| **vlan** | Assigns a VLAN ID to a subinterface. |

# names

To enable the association of a name with an IP address, use the **names** command in global configuration mode. You can associate only one name with an IP address. To disable displaying **name** values, use the **no names** command.

**names**
**no names**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No default behaviors or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Global Configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |

## Usage Guidelines

To enable the association of a name with an IP address, use the **names** command. You can associate only one name with an IP address.

You must first use the **names** command before you use the **name** command. Use the name command immediately after you use the names command and before you use the **write memory** command.

To disable displaying **name** values, use the **no names** command.

Both the name and names commands are saved in the configuration.

## Examples

This example shows that the **names** command allows you to enable use of the **name** command. The **name** command substitutes **sa_inside** for references to 192.168.42.3 and **sa_outside** for 209.165.201.3. You can use these names with the **ip address** commands when assigning IP addresses to the network interfaces. The **no names** command disables the **name** command values from displaying. Subsequent use of the **names** command again restores the **name** command value display.

```
ciscoasa(config)# names
ciscoasa(config)# name 192.168.42.3 sa_inside
ciscoasa(config)# name 209.165.201.3 sa_outside
ciscoasa(config-if)# ip address inside sa_inside 255.255.255.0
ciscoasa(config-if)# ip address outside sa_outside 255.255.255.224
ciscoasa(config)# show ip address
System IP Addresses:
```

```
inside ip address sa_inside mask 255.255.255.0
outside ip address sa_outside mask 255.255.255.224
ciscoasa(config)# no names
ciscoasa(config)# show ip address
System IP Addresses:
inside ip address 192.168.42.3 mask 255.255.255.0
outside ip address 209.165.201.3 mask 255.255.255.224
ciscoasa(config)# names
ciscoasa(config)# show ip address
System IP Addresses:
inside ip address sa_inside mask 255.255.255.0
outside ip address sa_outside mask 255.255.255.224
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure name** | Clears the list of names from the configuration. |
| **name** | Associates a name with an IP address. |
| **show running-config name** | Displays a list of names associated with IP addresses. |
| **show running-config names** | Displays the IP address-to-name conversions. |

# name-separator (pop3s, imap4s, smtps) (Deprecated)

**Note** The last supported release for this command was Version 9.5(1).

To specify a character as a delimiter between the e-mail and VPN username and password, use the **name-separator** command in the applicable e-mail proxy mode. To revert to the default, ":", use the **no** version of this command.

**name-separator** [ *symbol* ]
**no name-separator**

## Syntax Description

| | |
|---|---|
| symbol | (Optional) The character that separates the e-mail and VPN usernames and passwords. Choices are "@," (at) "\|" (pipe), ":"(colon), "#" (hash), "," (comma), and ";" (semi-colon). |

## Command Default

The default is ":" (colon).

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Pop3s | • Yes | • — | • Yes | • — | — |
| Imap4s | Yes | — | Yes | — | — |
| Smtps | Yes | — | Yes | — | — |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |
| 9.5(2) | This command was deprecated. |

## Usage Guidelines

The name separator must be different from the server separator.

## Examples

The following example shows how to set a hash (#) as the name separator for POP3S:

```
ciscoasa
(config)#
 pop3s
ciscoasa(config-pop3s)# name-separator #
```

**Related Commands**

| Command | Description |
|---|---|
| **server-separator** | Separates the e-mail and server names. |

# name-server

To identify one or more DNS servers so that the ASA can resolve hostnames to IP addresses, use the **name-server** command in dns server-group configuration mode. To remove a server or servers, use the **no** form of this command.

**Note** The ASA has limited support for using the DNS server, depending on the feature. For example, most commands require you to enter an IP address and can only use a name when you manually configure the **name** command to associate a name with an IP address and enable use of the names using the **names** command.

**name-server** *ip_address* [ *ip_address2* ] [ ... ] [ *ip_address6* ] [ *interface_name* ]
**no name-server** *ip_address* [ *ip_address2* ] [ ... ] [ *ip_address6* ] [ *interface_name* ]

## Syntax Description

| | |
|---|---|
| *interface_name* | (Optional) Specifies the interface name through which the ASA communicates with the server. If you do not specify the interface, the ASA checks the data routing table; if there are no matches, it then checks the management-only routing table. |
| *ip_address* | Specifies the DNS server IP address. You can specify up to six addresses as separate commands, or for convenience, up to six addresses in one command separated by spaces. If you enter multiple servers in one command, the ASA saves each server in a separate command in the configuration. The ASA tries each DNS server in order until it receives a response. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| dns server-group configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 7.1(1) | This command was added. |
| 9.5(1) | The *interface_name* argument was added. |

## Usage Guidelines

To enable DNS lookup on an interface, configure the **dns domain-lookup** command. If you do not enable DNS lookup, the DNS servers are not used.

By default, the ASA uses the **dns server-group DefaultDNS** server group for outgoing requests. You can change the default server group using the **dns-group** command. Other server groups can be associated with

specific domains. A DNS request that matches a domain associated with a DNS server group will use that group. For example, if you want traffic destined to inside eng.cisco.com servers to use an inside DNS server, you can map eng.cisco.com to an inside DNS group. All DNS requests that do not match a domain mapping will use the default DNS server group, which has no associated domains. For example, the DefaultDNS group can include a public DNS server available on the outside interface. Other DNS server groups can be configured for VPN tunnel groups. See the **tunnel-group** command for more information.

Some ASA features require use of a DNS server to access external servers by domain name; for example, the Botnet Traffic Filter feature requires a DNS server to access the dynamic database server and to resolve entries in the static database; and Cisco Smart Software Licensing needs DNS to resolve the License Authority address. Other features, such as the **ping** or **traceroute** command, let you enter a name that you want to ping or traceroute, and the ASA can resolve the name by communicating with a DNS server. Many SSL VPN and certificate commands also support names. You also must configure DNS servers to use fully qualified domain names (FQDN) network objects in access rules.

If you do not specify the interface for the **name-server**, the ASA checks the data routing table; if there are no matches, it then checks the management-only routing table. Note that if you have a default route through a data interface, all DNS traffic will match that route and never check the management-only routing table. In this scenario, always specify the interface if you need to access the server through a management interface.

## Examples

The following example adds three DNS servers to the group “DefaultDNS”:

```
ciscoasa(config)# dns server-group DefaultDNS
ciscoasa(config-dns-server-group)# name-server 10.1.1.1 10.2.3.4 192.168.5.5
```

The ASA saves the configuration as separate commands, as follows:

```
name-server 10.1.1.1
name-server 10.2.3.4
name-server 192.168.5.5
```

To add two additional servers, you can enter them as one command:

```
ciscoasa(config)# dns server-group
DefaultDNS
ciscoasa(config-dns-server-group)# name-server 10.5.1.1 10.8.3.8
```

To delete multiple servers you can enter them as multiple commands or as one command, as follows:

```
ciscoasa(config)# dns server-group DefaultDNS
ciscoasa(config-dns-server-group)# no
 name-server 10.5.1.1 10.8.3.8
```

## Related Commands

| Command | Description |
|---|---|
| **domain-name** | Sets the default domain name. |
| **retries** | Specifies the number of times to retry the list of DNS servers when the ASA does not receive a response. |
| **timeout** | Specifies the amount of time to wait before trying the next DNS server. |
| **show running-config dns server-group** | Shows one or all the existing dns-server-group configurations. |

# nat (global)

To configure twice NAT for IPv4, IPv6, or between IPv4 and IPv6 (NAT64), use the **nat** command in global configuration mode. To remove the twice NAT configuration, use the **no** form of this command.

For static NAT:

**nat** [ ( *real_ifc* , *mapped_ifc* ) ] [ *line* | { **after-auto** [ *line* ] } ] **source static** { *real_obj* | **any** } { *mapped_obj* | **interface** [ **ipv6** ] | **any** } ] [ **destination static** { *mapped_obj* | **interface** [ **ipv6** ] } { *real_obj* | **any** } ] [ **service** { *real_src_mapped_dest_svc_obj* | **any** } *mapped_src_real_dest_svc_obj* ] [ **net-to-net** ] [ **dns** ] [ **unidirectional** | [ **no-proxy-arp** ] [ **route-lookup** ] ] [ **inactive** ] [ **description** *desc*

**no nat** [ ( *real_ifc* , *mapped_ifc* ) ] [ *line* | { **after-auto** [ *line* ] } ] **source static** { *real_obj* | **any** } { *mapped_obj* | **interface** [ **ipv6** ] | **any** } ] [ **destination static** { *mapped_obj* | **interface** [ **ipv6** ] } { *real_obj* | **any** } ] [ **service** { *real_src_mapped_dest_svc_obj* | **any** } *mapped_src_real_dest_svc_obj* ] [ **net-to-net** ] [ **dns** ] [ **unidirectional** | [ **no-proxy-arp** ] [ **route-lookup** ] ] [ **inactive** ] [ **description** *desc*

For dynamic NAT:

**nat** [ ( *real_ifc* , *mapped_ifc* ) ] [ *line* | { **after-auto** [ *line* ] } ] **source dynamic** { *real_obj* | **any** } { *mapped_obj* | **interface** [ **ipv6** ] | **pat-pool** *mapped_obj* [ **round-robin** ] [ **extended** ] [ **flat** [ **include-reserve** ] ] [ **block-allocation** ] [ **interface** [ **ipv6** ] ] | **interface** [ **ipv6** ] } [ **destination static** { *mapped_obj* | **interface** [ **ipv6** ] } { *real_obj* | **any** } ] [ **service** { *mapped_dest_svc_obj* *real_dest_svc_obj* ] [ **dns** ] [ **unidirectional** ] [ **inactive** ] [ **description** *desc*

**no nat** [ ( *real_ifc* , *mapped_ifc* ) ] [ *line* | { **after-auto** [ *line* ] } ] **source dynamic** { *real_obj* | **any** } { *mapped_obj* | **interface** [ **ipv6** ] | **pat-pool** *mapped_obj* [ **round-robin** ] [ **extended** ] [ **flat** [ **include-reserve** ] ] [ **block-allocation** ] [ **interface** [ **ipv6** ] ] | **interface** [ **ipv6** ] } [ **destination static** { *mapped_obj* | **interface** [ **ipv6** ] } { *real_obj* | **any** } ] [ **service** { *mapped_dest_svc_obj* *real_dest_svc_obj* ] [ **dns** ] [ **unidirectional** ] [ **inactive** ] [ **description** *desc*

or

**no nat** { *line* **after-auto** *line* }

## Syntax Description

| | |
|---|---|
| **(***real_ifc***,***mapped_ifc***)** | (Optional) Specifies the real and mapped interfaces. If you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword **any** for one or both of the interfaces. For bridge group member interfaces (in transparent or routed mode), you must specify the real and mapped interfaces; you cannot use **any**.<br><br>Because twice NAT can translate both the source and destination addresses, these interfaces are better understood to be the source and destination interfaces. |
| **after-auto** | Inserts the rule at the end of section 3 of the NAT table, after the network object NAT rules. By default, twice NAT rules are added to section 1. You can insert a rule anywhere in section 3 using the *line* argument. |

| | |
|---|---|
| **any** | (Optional) Specifies a wildcard value. The main uses for **any** are:<br>• Interfaces—You can use **any** for one or both interfaces (**(any,outside)**, for example). If you do not specify the interfaces, then **any** is the default. However, **any** does not apply to bridge group member interfaces, and **any** is not available in transparent mode.<br>• Static NAT source real and mapped IP addresses—You can specify **source static any any** to enable identity NAT for all addresses.<br>• Dynamic NAT or PAT source real addresses—You can translate all addresses on the source interface by specifying **source dynamic any** *mapped_obj*<br>For static NAT, although **any** is also available for the real source port/mapped destination port, or for the source or destination real address (without **any** as the mapped address), these uses might result in unpredictable behavior.<br>**Note** The definition of "any" traffic (IPv4 vs. IPv6) depends on the rule. Before the ASA performs NAT on a packet, the packet must be IPv6-to-IPv6 or IPv4-to-IPv4; with this prerequisite, the ASA can determine the value of **any** in a NAT rule. For example, if you configure a rule from "any" to an IPv6 server, and that server was mapped from an IPv4 address, then **any** means "any IPv6 traffic." If you configure a rule from "any" to "any," and you map the source to the interface IPv4 address, then **any** means "any IPv4 traffic" because the mapped interface address implies that the destination is also IPv4. |
| **block-allocation** | Enables port block allocation. For carrier-grade or large-scale PAT, you can allocate a block of ports for each host, rather than have NAT allocate one port translation at a time. If you allocate a block of ports, subsequent connections from the host use new randomly-selected ports within the block. If necessary, additional blocks are allocated if the host has active connections for all ports in the original block. Port blocks are allocated in the 1024-65535 range only. Port block allocation is compatible with **round-robin**, but you cannot use the **extended** or **flat** [**include-reserve**] options. You also cannot use interface PAT fallback. |
| **description** *desc* | (Optional) Provides a description up to 200 characters. |
| destination | (Optional) Configures translation for the destination address. Although the main feature of twice NAT is the inclusion of the destination IP address, the destination address is optional. If you do specify the destination address, you can configure static translation for that address or just use identity NAT for it. You might want to configure twice NAT without a destination address to take advantage of some of the other qualities of twice NAT, including the use of network object groups for real addresses, or manually ordering of rules. For more information, see the CLI configuration guide. |

| | |
|---|---|
| **dns** | (Optional) Translates DNS replies. Be sure DNS inspection is enabled (**inspect dns**) (it is enabled by default). You cannot configure the **dns** keyword if you configure a **destination** address. Do not use this option with PAT rules. See the CLI configuration guide for more information. |
| **dynamic** | Configures dynamic NAT or PAT for the source addresses. The destination translation is always static. |
| **extended** | (Optional) Enables extended PAT for a PAT pool. Extended PAT uses 65535 ports per *service* , as opposed to per IP address, by including the destination address and port in the translation information. Normally, the destination port and address are not considered when creating PAT translations, so you are limited to 65535 ports per PAT address. For example, with extended PAT, you can create a translation of 10.1.1.1:1027 when going to 192.168.1.7:23 as well as a translation of 10.1.1.1:1027 when going to 192.168.1.7:80. |
| **flat** [**include-reserve**]<br><br>**include-reserve** | (Optional, pre-9.15) Enables use of the entire 1024 to 65535 port range when allocating ports. When choosing the mapped port number for a translation, the ASA uses the real source port number if it is available. However, without this option, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 1 to 511, 512 to 1023, and 1024 to 65535. To avoid running out of ports at the low ranges, configure this setting. To use the entire range of 1 to 65535, also specify the **include-reserve** keyword.<br><br>(9.15+) Starting with 9.15, flat is the default and unconfigurable behavior for a PAT pool. The **include-reserve** keyword is independent from the flat keyword, so you can still elect to include the reserved ports, 1-1023, in the PAT pool. |
| **inactive** | (Optional) To make this rule inactive without having to remove the command, use the **inactive** keyword. To reactivate it, reenter the whole command without the **inactive** keyword. |
| **interface** [**ipv6**] | (Optional) Uses the interface IP address as the mapped address. If you specify **ipv6**, then the IPv6 address of the interface is used.<br><br>For the dynamic NAT source mapped address, if you specify a mapped object or group followed by the **interface** keyword, then the IP address of the mapped interface is only used if all other mapped addresses are already allocated.<br><br>For dynamic PAT, you can specify **interface** alone for the source mapped address.<br><br>For static NAT with port translation (source or destination), be sure to also configure the **service** keyword.<br><br>For this option, you must configure a specific interface for the *mapped_ifc* .<br><br>This option is not available in transparent mode. In routed mode, you cannot use this option if the destination interface is a bridge group member. |

| | |
|---|---|
| *line* | (Optional) Inserts a rule anywhere in section 1 of the NAT table. By default, the NAT rule is added to the end of section 1 (see the CLI configuration guide for more information). If you want to add the rule into section 3 instead (after the network object NAT rules), then use the **after-auto** *line* option. |
| *mapped_dest_svc_obj* | (Optional) For dynamic NAT/PAT, specifies the mapped destination port (the destination translation is always static). See the **service** keyword for more information. |
| *mapped_object* | Identifies the mapped network object or object group (**object network** or **object-group network**).<br><br>For dynamic NAT, you typically configure a larger group of addresses to be mapped to a smaller group.<br><br>**Note** The mapped object or group cannot contain a subnet.You can share this mapped IP address across different dynamic NAT rules, if desired.You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address.<br><br>For dynamic PAT, configure a group of addresses to be mapped to a single address. You can either translate the real addresses to a single mapped address of your choosing, or you can translate them to the mapped interface address. If you want to use the interface address, do not configure a network object for the mapped address; instead use the **interface** keyword.<br><br>For static NAT, the mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired. For more information, see the CLI configuration guide. |
| *mapped_src_real_dest_svc_obj* | (Optional) For static NAT, specifies the either the mapped source port, the real destination port, or both together. See the **service** keyword for more information. |
| **net-to-net** | (Optional) For static NAT 46, specify **net-to-net** to translate the first IPv4 address to the first IPv6 address, the second to the second, and so on. Without this option, the IPv4-embedded method is used. For a one-to-one translation, you must use this keyword. |
| **no-proxy-arp** | (Optional) For static NAT, disables proxy ARP for incoming packets to the mapped IP addresses. |
| **pat-pool** *mapped_obj* | (Optional) Enables a PAT pool of addresses; all addresses in the object are used as PAT addresses. For dynamic NAT, you can configure the PAT pool as a fallback method. You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address. |
| *real_dest_svc_obj* | (Optional) For dynamic NAT/PAT, specifies the real destination port (the destination translation is always static). See the **service** keyword for more information. |

| | |
|---|---|
| *real_ifc* | (Optional) Specifies the name of the interface where packets may originate. For source option. For the source option, the origin_ifc is the real interface. For the destination option, the real_ifc is the mapped interface. |
| *real_object* | Identifies the real network object or object group (**object network** or **object-group network**). You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address. |
| *real_src_mapped_dest_svc_obj* | (Optional) For static NAT, specifies the either the real source port, the mapped destination port, or both together. See the **service** keyword for more information. |
| **round-robin** | (Optional) Enables round-robin address allocation for a PAT pool. By default, all ports for a PAT address will be allocated before the next PAT address is used. The round-robin method assigns an address/port from each PAT address in the pool before returning to use the first address again, and then the second address, and so on. |
| **route-lookup** | (Optional) For identity NAT in routed mode, determines the egress interface using a route lookup instead of using the interface specified in the NAT command. If you do not specify interfaces in the NAT command, a route lookup is used by default. |
| **service** | (Optional) Specifies the port translation.<br>• Dynamic NAT and PAT—Dynamic NAT and PAT do not support (additional) port translation. However, because the *destination* translation is always static, you can perform port translation for the destination port. A service object (**object service**) can contain both a source and destination port, but only the destination port is used in this case. If you specify the source port, it will be ignored.<br>• Static NAT with port translation—You should specify *either* the source *or* the destination port for both service objects. You should only specify *both* the source and destination ports if your application uses a fixed source port (such as some DNS servers); but fixed source ports are rare.<br>For source port translation, the objects must specify the source service. The order of the service objects in the command in this case is **service** *real_port mapped_port* . For destination port translation, the objects must specify the destination service. The order of the service objects in this case is **service** *mapped_port real_port* . In the rare case where you specify both the source and destination ports in the object, the first service object contains the real source port/mapped destination port; the second service object contains the mapped source port/real destination port. See the "Usage Guidelines" section for more information about "source" and "destination" terminology.<br>For identity port translation, simply use the same service object for both the real and mapped ports (source and/or destination ports, depending on your configuration). The "not equal" (**neq**) operator is not supported.<br>NAT only supports TCP or UDP. When translating a port, be sure the protocols in the real and mapped service objects are identical (both TCP or both UDP). |

| | |
|---|---|
| **source** | Configures translation for the source address. |
| **static** | Configures static NAT or static NAT with port translation. |
| **unidirectional** | (Optional) For static NAT, makes the translarion unidirection from the source to the destination; the destination addresses cannot initiate traffic to the source addresses. This option might be useful for testing purposes. |

## Command Default

- By default, the rule is added to the end of section 1 of the NAT table.
- The default value of *real_ifc* and *mapped_ifc* is **any**, which applies the rule to all interfaces.
- (8.3(1), 8.3(2), and 8.4(1)) The default behavior for identity NAT has proxy ARP disabled. You cannot configure this setting. (8.4(2) and later) The default behavior for identity NAT has proxy ARP enabled, matching other static NAT rules. You can disable proxy ARP if desired.
- If you specify an optional interface, then the ASA uses the NAT configuration to determine the egress interface. (8.3(1) through 8.4(1)) The only exception is for identity NAT, which always uses a route lookup, regardless of the NAT configuration. (8.4(2) and later) For identity NAT, the default behavior is to use the NAT configuration, but you have the option to always use a route lookup instead.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Global Configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 8.3(1) | This command was added. |
| 8.3(2) | When migrating from a pre-8.3 NAT exemption configuration, the keyword **unidirectional** is added for the resulting static identity NAT rule. |
| 8.4(2)/8.5(1) | The **no-proxy-arp**, **route-lookup**, **pat-pool**, and **round-robin** keywords were added.<br><br>The default behavior for identity NAT was changed to have proxy ARP enabled, matching other static NAT rules.<br><br>For pre-8.3 configurations, the migration of NAT exempt rules (the **nat 0 access-list** command) to 8.4(2) and later now includes the following keywords to disable proxy ARP and to use a route lookup: **no-proxy-arp** and **route-lookup**. The **unidirectional** keyword that was used for migrating to 8.3(2) and 8.4(1) is no longer used for migration. When upgrading to 8.4(2) from 8.3(1), 8.3(2), and 8.4(1), all identity NAT configurations will now include the **no-proxy-arp** and **route-lookup** keywords, to maintain existing functionality. The **unidirectional** keyword is removed. |

| Release | Modification |
|---|---|
| 8.4(3) | The **extended**, **flat**, and **include-reserve** keywords were added.<br>When using a PAT pool with round robin allocation, if a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available.<br>*This feature is not available in 8.5(1).* |
| 9.0(1) | NAT now supports IPv6 traffic, as well as translating between IPv4 and IPv6. Translating between IPv4 and IPv6 is not supported in transparent mode. We added the **interface ipv6** option and the **net-to-net** option. |
| 9.5(1) | The **block-allocation** keyword was added. |
| 9.15(1) | The **flat** keyword was removed, and the **include-reserve** keyword is no longer a sub-parameter of flat. All PAT pools now use a flat port range, 1024-65535, and you can optionally include the reserved ports, 1-1023. |
| 9.17(1) | You can specify an FQDN network object as the translated (mapped) destination. |

## Usage Guidelines

### Usage Guideline

Twice NAT lets you identify both the source and destination address in a single rule. Specifying both the source and destination addresses lets you specify that a source address should be translated to A when going to destination X, but be translated to B when going to destination Y, for example.

**Note** For static NAT, the rule is bidirectional, so be aware that "source" and "destination" are used in commands and descriptions throughout this guide even though a given connection might originate at the "destination" address. For example, if you configure static NAT with port translation, and specify the source address as a Telnet server, and you want all traffic going to that Telnet server to have the port translated from 2323 to 23, then in the command, you must specify the *source* ports to be translated (real: 23, mapped: 2323). You specify the source ports because you specified the Telnet server address as the **source** address.

The destination address is optional. If you specify the destination address, you can either map it to itself (identity NAT), or you can map it to a different address. The destination mapping is always a static mapping.

Twice NAT also lets you use service objects for static NAT with port translation; network object NAT only accepts inline definition.

For detailed information about the differences between twice NAT and network object NAT, see the CLI configuration guide.

Twice NAT rules are added to section 1 of the NAT rules table, or if specified, section 3. For more information about NAT ordering, see the CLI configuration guide.

### Mapped Address Guidelines

The mapped IP address pool cannot include:

- The mapped interface IP address. If you specify **any** interface for the rule, then all interface IP addresses are disallowed. For interface PAT (routed mode only), use the **interface** keyword instead of the IP address.
- (Transparent mode) The management IP address.

- (Dynamic NAT) The standby interface IP address when VPN is enabled.
- Existing VPN pool addresses.

**Prerequisites**

- For both the real and mapped addresses, configure network objects or network object groups (the **object network** or **object-group network** command). Network object groups are particularly useful for creating a mapped address pool with discontinuous IP address ranges or multiple hosts or subnets. You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address.
- For static NAT with port translation, configure TCP or UDP service objects (the **object service** command).

Objects and object groups used in NAT cannot be undefined; they must include IP addresses.

**Clearing Translation Sessions**

If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using clear xlate command. However, clearing the translation table disconnects all of the current connections.

**PAT Pool Guidelines**

- DNS rewrite is not applicable for PAT because multiple PAT rules are applicable for each A-record, and the PAT rule to use is ambiguous.
- (Pre-9.15) If available, the real source port number is used for the mapped port. However, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool that can be used. (8.4(3) and later, not including 8.5(1) or 8.6(1)) If you have a lot of traffic that uses the lower port ranges, you can now specify a flat range of ports to be used instead of the three unequal-sized tiers: either 1024 to 65535, or 1 to 65535.
- (9.15+) Ports are mapped to an available port in the 1024 to 65535 range. You can optionally include the reserved ports, those below 1024, to make the entire port range available for translations.

When operating in a cluster, blocks of 512 ports per address are allocated to the members of the cluster, and mappings are made within these port blocks. If you also enable block allocation, the ports are distributed according to the block allocation size, whose default is also 512.

- If you enable block allocation for a PAT pool, port blocks are allocated in the 1024-65535 range only. Thus, if an application requires a low port number (1-1023), it might not work. For example, an application requesting port 22 (SSH) will get a mapped port within the range of 1024-65535 and within the block allocated to the host.
- If you use an object group for the dynamic NAT mapped IP addresses, and the group includes host addresses, then enabling the PAT pool changes the use of those host addresses from PAT fallback to dynamic NAT.
- (8.4(3) and later, not including 8.5(1) or 8.6(1)) If you use the same PAT pool object in two separate rules, then be sure to specify the same options for each rule. For example, if one rule specifies extended PAT and a flat range, then the other rule must also specify extended PAT and a flat range.

**Exten ded PAT for a PAT Pool Guidelines**

- Many application inspections do not support extended PAT. See the configuration guide for a complete list of unsupported inspections.
- If you enable extended PAT for a dynamic PAT rule, then you cannot also use an address in the PAT pool as the PAT address in a separate static NAT-with-port-translation rule. For example, if the PAT pool includes 10.1.1.1, then you cannot create a static NAT-with-port-translation rule using 10.1.1.1 as the PAT address.
- If you use a PAT pool and specify an interface for fallback, you cannot specify extended PAT.
- For VoIP deployments that use ICE or TURN, do not use extended PAT. ICE and TURN rely on the PAT binding to be the same for all destinations.

**Round robin for a PAT Pool Guidelines**

- (8.4(3) and later, not including 8.5(1) or 8.6(1)) If a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available. **Note**: This "stickiness" does not survive a failover. If the ASA fails over, then subsequent connections from a host may not use the initial IP address.
- (8.4(2), 8.5(1), and 8.6(1)) If a host has an existing connection, then subsequent connections from that host will likely use *different* PAT addresses for each connection because of the round robin allocation. In this case, you may have problems when accessing two websites that exchange information about the host, for example an e-commerce site and a payment site. When these sites see two different IP addresses for what is supposed to be a single host, the transaction may fail.

**NAT and IPv6**

You can use NAT to translate between IPv6 networks, and also to translate between IPv4 and IPv6 networks (routed mode only). We recommend the following best practices. Note that you cannot perform NAT64/46 when the interfaces are members of the same bridge group.

- NAT66 (IPv6-to-IPv6)—We recommend using static NAT. Although you can use dynamic NAT or PAT, IPv6 addresses are in such large supply, you do not have to use dynamic NAT. If you do not want to allow returning traffic, you can make the static NAT rule unidirectional (twice NAT only).
- NAT46 (IPv4-to-IPv6)—We recommend using static NAT. Because the IPv6 address space is so much larger than the IPv4 address space, you can easily accommodate a static translation. If you do not want to allow returning traffic, you can make the static NAT rule unidirectional (twice NAT only). When translating to an IPv6 subnet (/96 or lower), the resulting mapped address is an IPv4-embedded IPv6 address, where the 32-bits of the IPv4 address is embedded after the IPv6 prefix. For example, if the IPv6 prefix is a /96 prefix, then the IPv4 address is appended in the last 32-bits of the address. For example, if you map 192.168.1.0/24 to 201b::0/96, then 192.168.1.4 will be mapped to 201b::0.192.168.1.4 (shown with mixed notation). If the prefix is smaller, such as /64, then the IPv4 address is appended after the prefix, and a suffix of 0s is appended after the IPv4 address.
- NAT64 (IPv6-to-IPv4)—You may not have enough IPv4 addresses to accommodate the number of IPv6 addresses. We recommend using a dynamic PAT pool to provide a large number of IPv4 translations.

## Examples

The following example includes a host on the 10.1.2.0/24 network that accesses two different servers. When the host accesses the server at 209.165.201.11, the real address is translated to 209.165.202.129:*port* . When the host accesses the server at 209.165.200.225, the real address is translated to 209.165.202.130:*port* .

```
ciscoasa(config)# object network myInsideNetwork
ciscoasa(config-network-object)# subnet 10.1.2.0 255.255.255.0
ciscoasa(config)# object network DMZnetwork1
ciscoasa(config-network-object)# subnet 209.165.201.0 255.255.255.224
ciscoasa(config)# object network PATaddress1
ciscoasa(config-network-object)# host 209.165.202.129
ciscoasa(config)# nat (inside,dmz) source dynamic myInsideNetwork PATaddress1 destination
static DMZnetwork1 DMZnetwork1
ciscoasa(config)# object network DMZnetwork2
ciscoasa(config-network-object)# subnet 209.165.200.224 255.255.255.224
ciscoasa(config)# object network PATaddress2
ciscoasa(config-network-object)# host 209.165.202.130
ciscoasa(config)# nat (inside,dmz) source dynamic myInsideNetwork PATaddress2 destination
static DMZnetwork2 DMZnetwork2
```

The following example shows the use of source and destination ports. The host on the 10.1.2.0/24 network accesses a single host for both web services and Telnet services. When the host accesses the server for Telnet services, the real address is translated to 209.165.202.129:*port* . When the host accesses the same server for web services, the real address is translated to 209.165.202.130:*port* .

```
ciscoasa(config)# object network myInsideNetwork
ciscoasa(config-network-object)# subnet 10.1.2.0 255.255.255.0
ciscoasa(config)# object network TelnetWebServer
ciscoasa(config-network-object)# host 209.165.201.11
ciscoasa(config)# object network PATaddress1
ciscoasa(config-network-object)# host 209.165.202.129
ciscoasa(config)# object service TelnetObj
ciscoasa(config-network-object)# service
 tcp
 destination eq telnet
ciscoasa(config)# nat (inside,outside) source dynamic myInsideNetwork PATaddress1 destination
 static TelnetWebServer TelnetWebServer service TelnetObj TelnetObj
ciscoasa(config)# object network PATaddress2
ciscoasa(config-network-object)# host 209.165.202.130
ciscoasa(config)# object service HTTPObj
ciscoasa(config-network-object)# service
 tcp
 destination eq http
ciscoasa(config)# nat (inside,outside) source dynamic myInsideNetwork PATaddress2 destination
 static TelnetWebServer TelnetWebServer service HTTPObj HTTPObj
```

The following example shows the use of static interface NAT with port translation. Hosts on the outside access an FTP server on the inside by connecting to the outside interface IP address with destination port 65000 through 65004. The traffic is untranslated to the internal FTP server at 192.168.10.100:6500 through :65004. Note that you specify the source port range in the service object (and not the destination port) because you want to translate the source address and port as identified in the command; the destination port is “any.” Because static NAT is bidirectional, “source” and “destination” refers primarily to the command keywords; the actual source and destination address and port in a packet depends on which host sent the packet. In this example, connections are originated from outside to inside, so the “source” address and port of the FTP server is actually the destination address and port in the originating packet.

```
ciscoasa(config)# object service FTP_PASV_PORT_RANGE
ciscoasa(config-service-obvject)# service tcp source range 65000 65004
ciscoasa(config)# object network HOST_FTP_SERVER
ciscoasa(config-network-obvject)# host 192.168.10.100
ciscoasa(config)# nat (inside,outside) source static HOST_FTP_SERVER interface service
FTP_PASV_PORT_RANGE FTP_PASV_PORT_RANGE
```

The following example configures dynamic NAT for an IPv6 inside network 2001:DB8:AAAA::/96 when accessing servers on the IPv4 209.165.201.1/27 network as well as servers on the 203.0.113.0/24 network:

```
ciscoasa(config)# object network INSIDE_NW
ciscoasa(config-network-object)# subnet 2001:DB8:AAAA::/96
ciscoasa(config)# object network MAPPED_1
ciscoasa(config-network-object)# range 209.165.200.225 209.165.200.254
ciscoasa(config)# object network MAPPED_2
ciscoasa(config-network-object)# range 209.165.202.129 209.165.200.158
ciscoasa(config)# object network SERVERS_1
ciscoasa(config-network-object)# subnet 209.165.201.0 255.255.255.224
ciscoasa(config)# object network SERVERS_2
ciscoasa(config-network-object)# subnet 203.0.113.0 255.255.255.0
ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW MAPPED_1 destination static
 SERVERS_1 SERVERS_1
ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW MAPPED_2 destination static
 SERVERS_2 SERVERS_2
```

The following example configures interface PAT for inside network 192.168.1.0/24 when accessing outside IPv6 Telnet server 2001:DB8::23, and Dynamic PAT using a PAT pool when accessing any server on the 2001:DB8:AAAA::/96 network.

```
ciscoasa(config)# object network INSIDE_NW
ciscoasa(config-network-object)# subnet 192.168.1.0 255.255.255.0
ciscoasa(config)# object network PAT_POOL
ciscoasa(config-network-object)# range 2001:DB8:AAAA::1 2001:DB8:AAAA::200
ciscoasa(config)# object network TELNET_SVR
ciscoasa(config-network-object)# host 2001:DB8::23
ciscoasa(config)# object service TELNET
ciscoasa(config-service-object)# service tcp destination eq 23
ciscoasa(config)# object network SERVERS
ciscoasa(config-network-object)# subnet 2001:DB8:AAAA::/96
ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW interface ipv6 destination
 static TELNET_SVR TELNET_SVR service TELNET TELNET
ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW pat-pool PAT_POOL destination
 static SERVERS SERVERS
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure nat** | Removes the NAT configuration (both twice NAT and network object NAT). |
| show nat | Displays NAT policy statistics. |
| show nat pool | Displays information about NAT pools. |
| **show running-config nat** | Shows the NAT configuration. |
| show xlate | Displays NAT session (xlate) information. |
| **xlate block-allocation** | Configures the PAT port block allocation characteristics. |

# nat (object)

To configure NAT for a network object, use the **nat** command in object network configuration mode. To remove the NAT configuration, use the **no** form of this command.

For dynamic NAT and PAT:

**nat** [ ( *real_ifc* , *mapped_ifc* ) ] **dynamic** { *mapped_inline_host_ip* [ **interface** [ **ipv6** ] ] | [ *mapped_obj* ] [ **pat-pool** *mapped_obj* [ **round-robin** ] [ **extended** ] [ **flat** [ **include-reserve** ] ] [ **block-allocation** ] ] [ **interface** [ **ipv6** ] ] } [ **dns** ]

**no nat** [ ( *real_ifc* , *mapped_ifc* ) ] **dynamic** { *mapped_inline_host_ip* [ **interface** [ **ipv6** ] ] | [ *mapped_obj* ] [ **pat-pool** *mapped_obj* [ **round-robin** ] [ **extended** ] [ **flat** [ **include-reserve** ] ] [ **block-allocation** ] ] [ **interface** [ **ipv6** ] ] } [ **dns** ]

For static NAT and static NAT with port translation:

**nat** [ ( *real_ifc* , *mapped_ifc* ) ] **static** { *mapped_inline_host_ip* | *mapped_obj* | **interface** [ **ipv6** ] } [ **net-to-net** ] [ **dns** | **service** { **tcp** | **udp** | **sctp** } *real_port mapped_port* ] [ **no-proxy-arp** ] [ **route-lookup** ]

**no nat** [ ( *real_ifc* , *mapped_ifc* ) ] **static** { *mapped_inline_host_ip* | *mapped_obj* | **interface** [ **ipv6** ] } [ **net-to-net** ] [ **dns** | **service** { **tcp** | **udp** | **sctp** } *real_port mapped_port* ] [ **no-proxy-arp** ] [ **route-lookup** ]

## Syntax Description

| | |
|---|---|
| **(***real_ifc***,***mapped_ifc***)** | (Optional) For static NAT, specifies the real and mapped interfaces. If you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword **any** for one or both of the interfaces. Be sure to include the parentheses in your command. For bridge group member interfaces (in transparent or routed mode), you must specify the real and mapped interfaces; you cannot use **any**. |
| **block-allocation** | Enables port block allocation. For carrier-grade or large-scale PAT, you can allocate a block of ports for each host, rather than have NAT allocate one port translation at a time. If you allocate a block of ports, subsequent connections from the host use new randomly-selected ports within the block. If necessary, additional blocks are allocated if the host has active connections for all ports in the original block. Port blocks are allocated in the 1024-65535 range only. Port block allocation is compatible with **round-robin**, but you cannot use the **extended** or **flat** [**include-reserve**] options. You also cannot use interface PAT fallback. |
| **dns** | (Optional) Translates DNS replies. Be sure DNS inspection (**inspect dns**) is enabled (it is enabled by default). This option is not available if you specify the **service** keyword (for static NAT). Do not use this option with PAT rules. For more information, see the CLI configuration guide. |
| **dynamic** | Configures dynamic NAT or PAT. |

| | |
|---|---|
| **extended** | (Optional) Enables extended PAT for a PAT pool. Extended PAT uses 65535 ports per *service* , as opposed to per IP address, by including the destination address and port in the translation information. Normally, the destination port and address are not considered when creating PAT translations, so you are limited to 65535 ports per PAT address. For example, with extended PAT, you can create a translation of 10.1.1.1:1027 when going to 192.168.1.7:23 as well as a translation of 10.1.1.1:1027 when going to 192.168.1.7:80. |
| **flat** [**include-reserve**]<br><br>**include-reserve** | (Optional, pre-9.15) Enables use of the entire 1024 to 65535 port range when allocating ports. When choosing the mapped port number for a translation, the ASA uses the real source port number if it is available. However, without this option, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 1 to 511, 512 to 1023, and 1024 to 65535. To avoid running out of ports at the low ranges, configure this setting. To use the entire range of 1 to 65535, also specify the **include-reserve** keyword.<br><br>(9.15+) Starting with 9.15, flat is the default and unconfigurable behavior for a PAT pool. The **include-reserve** keyword is independent from the flat keyword, so you can still elect to include the reserved ports, 1-1023, in the PAT pool. |
| **interface** [**ipv6**] | (Optional) For dynamic NAT, if you specify a mapped IP address, object, or group followed by the **interface** keyword, then the IP address of the mapped interface is only used if all of the other mapped addresses are already allocated.<br><br>For dynamic PAT, if you specify the **interface** keyword instead of a mapped IP address, object, or group, then you use the interface IP address for the mapped IP address. You must use this keyword when you want to use the interface IP address; you cannot enter it inline or as an object.<br><br>If you specify **ipv6**, then the IPv6 address of the interface is used.<br><br>For static NAT with port translation, you can specify the **interface** keyword if you also configure the **service** keyword.<br><br>For this option, you must configure a specific interface for the *mapped_ifc* .<br><br>You cannot specify **interface** in transparent mode. In routed mode, you cannot use this option if the destination interface is a bridge group member. |
| *mapped_inline_host_ip* | If you specify **dynamic**, then using a host IP address configures dynamic PAT. If you specify **static**, the netmask or range for the mapped network is the same as that of the real network. For example, if the real network is a host, then this address will be treated as a host address. In the case of a range or subnet, then the mapped addresses include the same number of addresses as the real range or subnet. For example, if the real address is defined as a range from 10.1.1.1 through 10.1.1.6, and you specify 172.20.1.1 as the mapped address, then the mapped range will include 172.20.1.1 through 172.20.1.6. If you want a many-to-one mapping, which we do not recommend, use a host network object instead of an inline address. |

| | |
|---|---|
| *mapped_obj* | Specifies the mapped IP address(es) as a network object (**object network**) or object group (**object-group network**). You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address.<br><br>For dynamic NAT, the object or group cannot contain a subnet. You can share this mapped object across different dynamic NAT rules, if desired. See the "Mapped Address Guidelines" for information about disallowed mapped IP addresses.<br><br>For static NAT, typically you configure the same number of mapped addresses as real addresses for a one-to-one mapping. You can, however, have a mismatched number of addresses. For more information, see the CLI configuration guide. |
| *mapped_port* | (Optional) Specifies the mapped TCP/UDP/SCTP port. You can specify ports by either a literal name or a number in the range of 0 to 65535. |
| **net-to-net** | (Optional) For NAT 46, specify **net-to-net** to translate the first IPv4 address to the first IPv6 address, the second to the second, and so on. Without this option, the IPv4-embedded method is used. For a one-to-one translation, you must use this keyword. |
| **no-proxy-arp** | (Optional) For static NAT, disables proxy ARP for incoming packets to the mapped IP addresses. |
| **pat-pool** *mapped_obj* | (Optional) Enables a PAT pool of addresses; all addresses in the object are used as PAT addresses. For dynamic NAT, you can configure the PAT pool as a fallback method. You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address. |
| *real_port* | (Optional) For static NAT, specifies the real TCP/UDP/SCTP port. You can specify ports by either a literal name or a number in the range of 0 to 65535. |
| **round-robin** | (Optional) Enables round-robin address allocation for a PAT pool. By default, all ports for a PAT address will be allocated before the next PAT address is used. The round-robin method assigns an address/port from each PAT address in the pool before returning to use the first address again, and then the second address, and so on. |
| **route-lookup** | (Optional) For identity NAT in routed mode, determines the egress interface using a route lookup instead of using the interface specified in the NAT command. If you do not specify interfaces in the NAT command, a route lookup is used by default. |
| **service** {**tcp** \| **udp** \| **sctp**} | (Optional) For static NAT with port translation, specifies the protocol for port translation: TCP, UDP, SCTP. |
| **static** | Configures static NAT or static NAT with port translation. |

## Command Default

- The default value of *real_ifc* and *mapped_ifc* is **any**, which applies the rule to all interfaces.
- (8.3(1), 8.3(2), and 8.4(1)) The default behavior for identity NAT has proxy ARP disabled. You cannot configure this setting. (8.4(2) and later) The default behavior for identity NAT has proxy ARP enabled, matching other static NAT rules. You can disable proxy ARP if desired.
- If you specify an optional interface, then the ASA uses the NAT configuration to determine the egress interface. (8.3(1) through 8.4(1)) The only exception is for identity NAT, which always uses a route

lookup, regardless of the NAT configuration. (8.4(2) and later) For identity NAT, the default behavior is to use the NAT configuration, but you have the option to always use a route lookup instead.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Object network configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 8.3(1) | This command was added. |
| 8.4(2)/8.5(1) | The **no-proxy-arp**, **route-lookup**, **pat-pool**, and **round-robin** keywords were added.<br>The default behavior for identity NAT was changed to have proxy ARP enabled, matching other static NAT rules.<br>When upgrading to 8.4(2) from 8.3(1), 8.3(2), and 8.4(1), all identity NAT configurations will now include the **no-proxy-arp** and **route-lookup** keywords, to maintain existing functionality. |
| 8.4(3) | The **extended**, **flat**, and **include-reserve** keywords were added.<br>When using a PAT pool with round robin allocation, if a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available.<br>*This feature is not available in 8.5(1).* |
| 9.0(1) | NAT now supports IPv6 traffic, as well as translating between IPv4 and IPv6. Translating between IPv4 and IPv6 is not supported in transparent mode. We added the **interface ipv6** option and the **net-to-net** option. |
| 9.5(1) | The **block-allocation** keyword was added. |
| 9.5(2) | The **service sctp** keyword was added. |
| 9.15(1) | The **flat** keyword was removed, and the **include-reserve** keyword is no longer a sub-parameter of flat. All PAT pools now use a flat port range, 1024-65535, and you can optionally include the reserved ports, 1-1023. |

## Usage Guidelines

When a packet enters the ASA, both the source and destination IP addresses are checked against the network object NAT rules. The source and destination address in the packet can be translated by separate rules if separate matches are made. These rules are not tied to each other; different combinations of rules can be used depending on the traffic.

Because the rules are never paired, you cannot specify that a source address should be translated to A when going to destination X, but be translated to B when going to destination Y. Use twice NAT for that kind of functionality (twice NAT lets you identify the source and destination address in a single rule).

For detailed information about the differences between twice NAT and network object NAT, see the CLI configuration guide.

Network object NAT rules are added to section 2 of the NAT rules table. For more information about NAT ordering, see the CLI configuration guide.

Depending on the configuration, you can configure the mapped address inline if desired or you can create a network object or network object group for the mapped address (the **object network** or **object-group network** command). Network object groups are particularly useful for creating a mapped address pool with discontinuous IP address ranges or multiple hosts or subnets. You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address.

Objects and object groups used in NAT cannot be undefined; they must include IP addresses.

You can only define a single NAT rule for a given object; if you want to configure multiple NAT rules, you need to create multiple objects that specify the same IP address, for example, **object network obj-10.10.10.1-01**, **object network obj-10.10.10.1-02**, and so on.

**Mapped Address Guidelines**

The mapped IP address pool cannot include:

- The mapped interface IP address. If you specify **any** interface for the rule, then all interface IP addresses are disallowed. For interface PAT (routed mode only), use the **interface** keyword instead of the IP address.
- (Transparent mode) The management IP address.
- (Dynamic NAT) The standby interface IP address when VPN is enabled.
- Existing VPN pool addresses.

**Clearing Translation Sessions**

If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using clear xlate command. However, clearing the translation table disconnects all of the current connections.

**PAT Pool Guidelines**

- DNS rewrite is not applicable for PAT because multiple PAT rules are applicable for each A-record, and the PAT rule to use is ambiguous.
- (Pre-9.15) If available, the real source port number is used for the mapped port. However, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool that can be used. (8.4(3) and later, not including 8.5(1) or 8.6(1)) If you have a lot of traffic that uses the lower port ranges, you can now specify a flat range of ports to be used instead of the three unequal-sized tiers: either 1024 to 65535, or 1 to 65535.
- (9.15+) Ports are mapped to an available port in the 1024 to 65535 range. You can optionally include the reserved ports, those below 1024, to make the entire port range available for translations.

When operating in a cluster, blocks of 512 ports per address are allocated to the members of the cluster, and mappings are made within these port blocks. If you also enable block allocation, the ports are distributed according to the block allocation size, whose default is also 512.

- If you enable block allocation for a PAT pool, port blocks are allocated in the 1024-65535 range only. Thus, if an application requires a low port number (1-1023), it might not work. For example, an application

requesting port 22 (SSH) will get a mapped port within the range of 1024-65535 and within the block allocated to the host.

If you use an object group for the dynamic NAT mapped IP addresses, and the group includes host addresses, then enabling the PAT pool changes the use of those host addresses from PAT fallback to dynamic NAT.

- (8.4(3) and later, not including 8.5(1) or 8.6(1)) If you use the same PAT pool object in two separate rules, then be sure to specify the same options for each rule. For example, if one rule specifies extended PAT and a flat range, then the other rule must also specify extended PAT and a flat range.

**Extended PAT for a PAT Pool Guidelines**

- Many application inspections do not support extended PAT. See the configuration guide for a complete list of unsupported inspections.
- If you enable extended PAT for a dynamic PAT rule, then you cannot also use an address in the PAT pool as the PAT address in a separate static NAT-with-port-translation rule. For example, if the PAT pool includes 10.1.1.1, then you cannot create a static NAT-with-port-translation rule using 10.1.1.1 as the PAT address.
- If you use a PAT pool and specify an interface for fallback, you cannot specify extended PAT.
- For VoIP deployments that use ICE or TURN, do not use extended PAT. ICE and TURN rely on the PAT binding to be the same for all destinations.

**Round robin for a PAT Pool Guidelines**

- (8.4(3) and later, not including 8.5(1) or 8.6(1)) If a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available. **Note**: This "stickiness" does not survive a failover. If the ASA fails over, then subsequent connections from a host may not use the initial IP address.
- (8.4(2), 8.5(1), and 8.6(1)) If a host has an existing connection, then subsequent connections from that host will likely use *different* PAT addresses for each connection because of the round robin allocation. In this case, you may have problems when accessing two websites that exchange information about the host, for example an e-commerce site and a payment site. When these sites see two different IP addresses for what is supposed to be a single host, the transaction may fail.
- Round robin, especially when combined with extended PAT, can consume a large amount of memory.

**NAT and IPv6**

You can use NAT to translate between IPv6 networks, and also to translate between IPv4 and IPv6 networks (routed mode only). We recommend the following best practices. Note that you cannot perform NAT64/46 when the interfaces are members of the same bridge group.

- NAT66 (IPv6-to-IPv6)—We recommend using static NAT. Although you can use dynamic NAT or PAT, IPv6 addresses are in such large supply, you do not have to use dynamic NAT. If you do not want to allow returning traffic, you can make the static NAT rule unidirectional (twice NAT only).
- NAT46 (IPv4-to-IPv6)—We recommend using static NAT. Because the IPv6 address space is so much larger than the IPv4 address space, you can easily accommodate a static translation. If you do not want to allow returning traffic, you can make the static NAT rule unidirectional (twice NAT only). When translating to an IPv6 subnet (/96 or lower), the resulting mapped address is an IPv4-embedded IPv6 address, where the 32-bits of the IPv4 address is embedded after the IPv6 prefix. For example, if the IPv6 prefix is a /96 prefix, then the IPv4 address is appended in the last 32-bits of the address. For

example, if you map 192.168.1.0/24 to 201b::0/96, then 192.168.1.4 will be mapped to 201b::0.192.168.1.4 (shown with mixed notation). If the prefix is smaller, such as /64, then the IPv4 address is appended after the prefix, and a suffix of 0s is appended after the IPv4 address.

- NAT64 (IPv6-to-IPv4)—You may not have enough IPv4 addresses to accommodate the number of IPv6 addresses. We recommend using a dynamic PAT pool to provide a large number of IPv4 translations.

## Examples

### Dynamic NAT Examples

The following example configures dynamic NAT that hides 192.168.2.0 network behind a range of outside addresses 2.2.2.1-2.2.2.10:

```
ciscoasa(config)# object network my-range-obj
ciscoasa(config-network-object)# range 2.2.2.1 2.2.2.10
ciscoasa(config)# object network my-inside-net
ciscoasa(config-network-object)# subnet 192.168.2.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic my-range-obj
```

The following example configures dynamic NAT with dynamic PAT backup. Hosts on inside network 10.76.11.0 are mapped first to the nat-range1 pool (10.10.10.10-10.10.10.20). After all addresses in the nat-range1 pool are allocated, dynamic PAT is performed using the pat-ip1 address (10.10.10.21). In the unlikely event that the PAT translations are also use up, dynamic PAT is performed using the outside interface address.

```
ciscoasa(config)# object network nat-range1
ciscoasa(config-network-object)# range 10.10.10.10 10.10.10.20
ciscoasa(config-network-object)# object network pat-ip1
ciscoasa(config-network-object)# host 10.10.10.21
ciscoasa(config-network-object)# object-group network nat-pat-grp
ciscoasa(config-network-object)# network-object object nat-range1
ciscoasa(config-network-object)# network-object object pat-ip1
ciscoasa(config-network-object)# object network my_net_obj5
ciscoasa(config-network-object)# subnet 10.76.11.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic nat-pat-grp interface
```

The following example configures dynamic NAT with dynamic PAT backup to translate IPv6 hosts to IPv4. Hosts on inside network 2001:DB8::/96 are mapped first to the IPv4_NAT_RANGE pool (209.165.201.1 to 209.165.201.30). After all addresses in the IPv4_NAT_RANGE pool are allocated, dynamic PAT is performed using the IPv4_PAT address (209.165.201.31). In the event that the PAT translations are also used up, dynamic PAT is performed using the outside interface address.

```
ciscoasa(config)# object network IPv4_NAT_RANGE
ciscoasa(config-network-object)# range 209.165.201.1 209.165.201.30
ciscoasa(config-network-object)# object network IPv4_PAT
ciscoasa(config-network-object)# host 209.165.201.31
ciscoasa(config-network-object)# object-group network IPv4_GROUP
ciscoasa(config-network-object)# network-object object IPv4_NAT_RANGE
ciscoasa(config-network-object)# network-object object IPv4_PAT
ciscoasa(config-network-object)# object network my_net_obj5
ciscoasa(config-network-object)# subnet 2001:DB8::/96
ciscoasa(config-network-object)# nat (inside,outside) dynamic IPv4_GROUP interface
```

### Dynamic PAT Example

The following example configures dynamic PAT that hides the 192.168.2.0 network behind address 2.2.2.2:

```
ciscoasa(config)# object network my-inside-net
ciscoasa(config-network-object)# subnet 192.168.2.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic 2.2.2.2
```

The following example configures dynamic PAT that hides the 192.168.2.0 network behind the outside interface address:

```
ciscoasa(config)# object network my-inside-net
ciscoasa(config-network-object)# subnet 192.168.2.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic interface
```

The following example configures dynamic PAT with a PAT pool to translate the inside IPv6 network to an outside IPv4 network:

```
ciscoasa(config)# object network IPv4_POOL
ciscoasa(config-network-object)# range 203.0.113.1 203.0.113.254
ciscoasa(config)# object network IPv6_INSIDE
ciscoasa(config-network-object)# subnet 2001:DB8::/96
ciscoasa(config-network-object)# nat (inside,outside) dynamic pat-pool IPv4_POOL
```

### Static NAT Examples

The following example configures static NAT for the real host 1.1.1.1 on the inside to 2.2.2.2 on the outside with DNS rewrite enabled.

```
ciscoasa(config)# object network my-host-obj1
ciscoasa(config-network-object)# host 1.1.1.1
ciscoasa(config-network-object)# nat (inside,outside) static 2.2.2.2 dns
```

The following example configures static NAT for the real host 1.1.1.1 on the inside to 2.2.2.2 on the outside using a mapped object.

```
ciscoasa(config)# object network my-mapped-obj
ciscoasa(config-network-object)# host 2.2.2.2
ciscoasa(config-network-object)# object network my-host-obj1
ciscoasa(config-network-object)# host 1.1.1.1
ciscoasa(config-network-object)# nat (inside,outside) static my-mapped-obj
```

The following example configures static NAT with port translation for 1.1.1.1 at TCP port 21 to the outside interface at port 2121.

```
ciscoasa(config)# object network my-ftp-server
ciscoasa(config-network-object)# host 1.1.1.1
ciscoasa(config-network-object)# nat (inside,outside) static interface service tcp 21 2121
```

The following example maps an inside IPv4 network to an outside IPv6 network.

```
ciscoasa(config)# object network inside_v4_v6
```

```
ciscoasa(config-network-object)# subnet 10.1.1.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) static 2001:DB8::/96
```

The following example maps an inside IPv6 network to an outside IPv6 network.

```
ciscoasa(config)# object network inside_v6
ciscoasa(config-network-object)# subnet 2001:DB8:AAAA::/96
ciscoasa(config-network-object)# nat (inside,outside) static 2001:DB8:BBBB::/96
```

### Identity NAT Examples

The following example maps a host address to itself using an inline mapped address:

```
ciscoasa(config)# object network my-host-obj1
ciscoasa(config-network-object)# host 10.1.1.1
ciscoasa(config-network-object)# nat (inside,outside) static 10.1.1.1
```

The following example maps a host address to itself using a network object:

```
ciscoasa(config)# object network my-host-obj1-identity
ciscoasa(config-network-object)# host 10.1.1.1
ciscoasa(config-network-object)# object network my-host-obj1
ciscoasa(config-network-object)# host 10.1.1.1
ciscoasa(config-network-object)# nat (inside,outside) static my-host-obj1-identity
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure nat** | Removes the NAT configuration (both twice NAT and network object NAT). |
| show nat | Displays NAT policy statistics. |
| show nat pool | Displays information about NAT pools. |
| **show running-config nat** | Displays the NAT configuration. |
| show xlate | Displays xlate information. |
| **xlate block-allocation** | Configures the PAT port block allocation characteristics. |

# nat (vpn load-balancing)

To set the IP address to which NAT translates the IP address of this device, use the **nat** command in VPN load-balancing configuration mode. To disable this NAT translation, use the **no** form of this command.

**nat** *ip-address*
**no nat** [ *ip-adddress* ]

## Syntax Description

| | |
|---|---|
| *ip-address* | The IP address to which you want this NAT to translate the IP address of this device. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| VPN load-balancing configuration | • Yes | • — | • Yes | • — | — |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |

## Usage Guidelines

You must first use the **vpn load-balancing** command to enter VPN load-balancing mode.

In the **no nat** form of the command, if you specify the optional *ip-address* value, the IP address must match the existing NAT IP address in the running configuration.

## Examples

The following is an example of a VPN load-balancing command sequence that includes a **nat** command that sets the NAT-translated address to 192.168.10.10:

```
ciscoasa(config)# interface GigabitEthernet 0/1
ciscoasa(config-if)# ip address 209.165.202.159 255.255.255.0
ciscoasa(config)# nameif test
ciscoasa(config)# interface GigabitEthernet 0/2
ciscoasa(config-if)# ip address 209.165.201.30 255.255.255.0
ciscoasa(config)# nameif foo
ciscoasa(config)# vpn load-balancing
ciscoasa(config-load-balancing)# nat 192.168.10.10
ciscoasa(config-load-balancing)# priority 9
ciscoasa(config-load-balancing)# interface lbpublic test
ciscoasa(config-load-balancing)# interface lbprivate foo
ciscoasa(config-load-balancing)# cluster ip address 209.165.202.224
ciscoasa(config-load-balancing)# cluster port 9023
ciscoasa(config-load-balancing)# participate
ciscoasa(config-load-balancing)# participate
```

**Related Commands**

| Command | Description |
|---|---|
| **vpn load-balancing** | Enter VPN load-balancing mode. |

# nat-assigned-to-public-ip

To automatically translate a VPN peer's local IP address back to the peer's real IP address, use the **nat-assigned-to-public-ip** command in tunnel-group general-attributes configuration mode. To disable the NAT rules, use the **no** form of this command.

**nat-assigned-to-public-ip** *interface*
**no nat-assigned-to-public-ip** *interface*

## Syntax Description

| | |
|---|---|
| *interface* | Specifies the interface where you want to apply NAT. |

## Command Default

This command is disabled by default.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Tunnel-group general-attributes configuration | • Yes | • Yes | • Yes | — | — |

## Command History

| Release | Modification |
|---|---|
| 8.4(3) | This command was added. |

## Usage Guidelines

In rare situations, you might want to use a VPN peer's real IP address on the inside network instead of an assigned local IP address. Normally with VPN, the peer is given an assigned local IP address to access the inside network. However, you might want to translate the local IP address back to the peer's real public IP address if, for example, your inside servers and network security is based on the peer's real IP address.

You can enable this feature on one interface per tunnel group. Object NAT rules are dynamically added and deleted when the VPN session is established or disconnected. You can view the rules using the **show nat** command.

### Data Flow

The following steps describe the packet flow through the ASA when this feature is enabled:

**1.**The VPN peer sends a packet to the ASA.

The outer source/destination consists of the peer public IP address/ASA IP address. The encrypted inner source/destination consists of the VPN-assigned IP address/inside server address.

**2.**The ASA decrypts the packet (removing the outer source/destination).

**3.**The ASA performs a route lookup for the inside server, and sends the packet to the inside interface.

**4.**The automatically created VPN NAT policy translates the VPN-assigned source IP address to the peer public IP address.

**5.**The ASA sends the translated packet to the server.

**6.**The server responds to the packet, and sends it to the peer's public IP address.

**7.**The ASA receives the response, and untranslates the destination IP address to the VPN-assigned IP address.

**8.**The ASA forwards the untranslated packet to the outside interface where it is encrypted, and an outer source/destination is added consisting of the ASA IP address/peer public IP address.

**9.**The ASA sends the packet back to the peer.

**10.**The peer decrypts and processes the data.

**Limitations**

Because of routing issues, we do not recommend using this feature unless you know you need this feature; contact Cisco TAC to confirm feature compatibility with your network. See the following limitations:

- Only supports Cisco IPsec and Secure Client.
- Return traffic to the public IP addresses must be routed back to the ASA so the NAT policy and VPN policy can be applied.
- If you enable reverse route injection (see the **set reverse-route** command), only the VPN-assigned IP address is advertised.
- Does not support load-balancing (because of routing issues).
- Does not support roaming (public IP changing).

## Examples

The following example enables NAT to the public IP for the "vpnclient" tunnel group:

```
ciscoasa# ip local pool client 10.1.226.4-10.1.226.254
ciscoasa# tunnel-group vpnclient type remote-access
ciscoasa# tunnel-group vpnclient general-attributes
ciscoasa(config-tunnel-general)# address-pool client
ciscoasa(config-tunnel-general)# nat-assigned-to-public-ip inside
```

The following is sample output from the **show nat detail** command showing an automatic NAT rule from peer 209.165.201.10 with assigned IP 10.1.226.174:

```
ciscoasa# show nat detail
 Auto NAT Policies (Section 2)
 1 (outside) to (inside) source static _vpn_nat_10.1.226.174 209.165.201.10
     translate_hits = 0, untranslate_hits = 0
     Source - Origin: 10.1.226.174/32, Translated: 209.165.201.10/32
```

## Related Commands

| Command | Description |
|---|---|
| **show nat** | Shows current xlates. |
| **tunnel-group general-attributes** | Sets general attributes for a tunnel group. |
| **debug menu webvpn 99** | For AnyConnect SSL sessions, the VPN NAT interface is stored in the session. |

| Command | Description |
|---|---|
| **debug menu ike 2** *peer_ip* | For Cisco IPsec client sessions, the VPN NAT interface is stored in the SA. |
| debug nat 3 | Shows debug messages for NAT. |

# nat-rewrite

To enable NAT rewrite for IP addresses embedded in the A-record of a DNS response, use the **nat-rewrite** command in parameters configuration mode. To disable this feature, use the **no** form of this command.

**nat-rewrite**
**no nat-rewrite**

## Syntax Description

This command has no arguments or keywords.

## Command Default

NAT rewrite is enabled by default. This feature can be enabled when **inspect dns** is configured even if a **policy-map type inspect dns** is not defined. To disable, **no nat-rewrite** must explicitly be stated in the policy map configuration. If **inspect dns** is not configured, NAT rewrite is not performed.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Parameters configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 7.2(1) | This command was added. |

## Usage Guidelines

This feature performs NAT translation of A-type Resource Record (RR) in a DNS response.

## Examples

The following example shows how to enable NAT rewrite in a DNS inspection policy map:

```
ciscoasa(config)# policy-map type inspect dns preset_dns_map
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# nat-rewrite
```

## Related Commands

| Command | Description |
|---|---|
| **class** | Identifies a class map name in the policy map. |
| **class-map type inspect** | Creates an inspection class map to match traffic specific to an application. |
| **policy-map** | Creates a Layer 3/4 policy map. |
| **show running-config policy-map** | Display all current policy map configurations. |

# nbns-server

To configure an NBNS server, use the **nbns-server** command in tunnel-group webvpn-attributes configuration mode. To remove the NBNS server from the configuration, use the **no** form of this command.

The ASA queries NBNS servers to map NetBIOS names to IP addresses. WebVPN requires NetBIOS to access or share files on remote systems.

**nbns-server** { *ipaddr* | *hostname* } [ **master** ] [ **timeout** *timeout* ] [ **retry** *retries* ]
**no nbns-server**

## Syntax Description

| | |
|---|---|
| *hostname* | Specifies the hostname for the NBNS server. |
| *ipaddr* | Specifies the IP address for the NBNS server. |
| **master** | Indicates that this is a master browser, rather than a WINS server. |
| **retry** | Indicates that a retry value follows. |
| *retries* | Specifies the number of times to retry queries to NBNS servers. The ASA recycles through the list of servers the number of times you specify here before sending an error message. The default value is 2; the range is 1 through 10. |
| **timeout** | Indicates that a timeout value follows. |
| *timeout* | Specifies the amount of time the ASA waits before sending the query again, to the same server if there is only one, or another server if there are multiple NBNS servers. The default timeout is 2 seconds; the range is 1 to 30 seconds. |

## Command Default

No NBNS server is configured by default.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Tunnel-group webvpn-attributes configuration | • Yes | • — | • Yes | • — | — |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |
| 7.1(1) | Moved from webvpn mode to tunnel-group webvpn configuration mode. |

## Usage Guidelines

In Release 7.1(1), if you enter this command in webvpn configuration mode, it is transformed to the same command in tunnel-group webvpn-attributes configuration mode.

Maximum of 3 server entries. The first server you configure is the primary server, and the others are backups, for redundancy.

Use the **no** option to remove the matching entry from the configuration.

## Examples

The following example shows how to configure the tunnel-group "test" with an NBNS server that is a master browser with an IP address of 10.10.10.19, a timeout value of 10 seconds, and 8 retries. It also shows how to configure an NBNS WINS server with an IP address of 10.10.10.24, a timeout value of 15 seconds, and 8 retries.

```
ciscoasa
(config)#
 tunnel-group test type webvpn
ciscoasa
(config)#
 tunnel-group test webvpn-attributes
ciscoasa(config-tunnel-webvpn)# nbns-server 10.10.10.19 master timeout 10 retry 8
ciscoasa(config-tunnel-webvpn)# nbns-server 10.10.10.24 timeout 15 retry 8
ciscoasa(config-tunnel-webvpn)#
```

## Related Commands

| Command | Description |
|---|---|
| clear configure group-policy | Removes the configuration for a particular group policy or for all group policies. |
| show running-config group-policy | Displays the running configuration for a particular group policy or for all group policies. |
| tunnel-group webvpn-attributes | Specifies the WebVPN attributes for the named tunnel-group. |

# neighbor (router eigrp)

To define an EIGRP neighbor router with which to exchange routing information, use the **neighbor** command in router eigrp configuration mode. To remove a neighbor entry, use the **no** form of this command.

**neighbor** *ip_address* **interface** *name*
**no neighbor** *ip_address* **interface** *name*

## Syntax Description

| | |
|---|---|
| **interface** *name* | The interface name, as specified by the **nameif** command, through which the neighbor can be reached. |
| *ip_address* | IPv4 or IPv6 address of the neighbor router with which routing information is exchanged. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Router eigrp configuration | — | • | • Yes | — | — |

## Command History

| Release | Modification |
|---|---|
| 8.0(2) | This command was added. |
| 9.20(1) | Support for IPv6 was added. |

## Usage Guidelines

You can use multiple neighbor statements to establish peering sessions with specific EIGRP neighbors. The interface through which EIGRP exchanges routing updates must be specified in the neighbor statement. The interfaces through which two EIGRP neighbors exchange routing updates must be configured with IP addresses from the same network.

**Note** Configuring the **passive-interface** command for an interface suppresses all incoming and outgoing routing updates and hello messages on that interface. EIGRP neighbor adjacencies cannot be established or maintained over an interface that is configured as passive.

EIGRP hello messages are sent as unicast messages to neighbors defined using the **neighbor** command.

## Examples

The following example configures EIGRP peering sessions with the 192.168.1.1 and 192.168.2.2 neighbors:

```
ciscoasa(config)# router eigrp 100
ciscoasa(config-router)# network 192.168.0.0
ciscoasa(config-router)# neighbor 192.168.1.1 interface outside
ciscoasa(config-router)# neighbor 192.168.2.2 interface branch_office
```

## Examples

The following example configures EIGRP peering sessions with the fe80::250:56ff:feb9:b41b and fe80::250:56ff:fe9f:13f4 neighbors:

```
ciscoasa(config)# rtr eigrp 100
ciscoasa(config-rtr)# neighbor fe80::250:56ff:feb9:b41b interface gig1
ciscoasa(config-rtr)# neighbor fe80::250:56ff:fe9f:13f4 interface branch_office
```

## Related Commands

| Command | Description |
|---|---|
| **debug eigrp neighbors** | Displays debug information for EIGRP neighbor messages. |
| **show eigrp neighbors** | Displays the EIGRP neighbor table. |

# neighbor (router ospf)

To define a static neighbor on a point-to-point, non-broadcast network, use the **neighbor** command in router ospf configuration mode. To remove the statically defined neighbor from the configuration, use the **no** form of this command.

**neighbor** *ip_address* [ **interface** *name* ]
**no neighbor** *ip_address* [ **interface** *name* ]

## Syntax Description

| | |
|---|---|
| **interface** *name* | (Optional) Specifies the interface name, as specified by the **nameif** command, through which the neighbor can be reached. |
| *ip_address* | Specifies the IP address of the neighbor router. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Router ospf configuration | • Yes | • — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |
| 9.0(1) | Support for multiple context mode was added. |

## Usage Guidelines

The **neighbor** command is used to advertise OSPF routes over VPN tunnels. One neighbor entry must be included for each known non-broadcast network neighbor. The neighbor address must be on the primary address of the interface.

The **interface** option needs to be specified when the neighbor is not on the same network as any of the directly connected interfaces of the system. Additionally, a static route must be created to reach the neighbor.

## Examples

The following example defines a neighbor router with an address of 192.168.1.1:

```
ciscoasa(config-router)# neighbor 192.168.1.1
```

## Related Commands

| Command | Description |
|---|---|
| **router ospf** | Enters router configuration mode. |

| Command | Description |
| --- | --- |
| **show running-config router** | Displays the commands in the global router configuration. |

# neighbor activate

To enable the exchange of information with a Border Gateway Protocol (BGP) neighbor, use the neighbor activate command in address-family configuration mode. To disable the exchange of an address with a BGP neighbor, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **activate**
**no neighbor** { *ip_address* | *ipv6-address* } **activate**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the BGP router. |
| ipv6-address | IPv6 address of the BGP router |

## Command Default

Address exchange with BGP neighbors is enabled by default for the IPv4 address family. You cannot enable address exchange for any other address families.

**Note** Address exchange for the IPv4 address family is enabled by default for each BGP routing session defined by the neighbor remote-as command; unless you configure the no bgp default ipv4-activate command before configuring the neighbor remote-as command, or you disable address exchange with a specific neighbor by using the no neighbor activate command.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | • — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument and support for IPv6 address family were added. |

## Usage Guidelines

You can use this command to advertise address information in the form of an IP prefix. The address prefix information is known as Network Layer Reachability Information (NLRI) in BGP.

## Examples

The following example enables address exchange for IPv4 address family unicast for the BGP neighbor 172.16.1.1:

```
ciscoasa(config)# router bgp 50000
ciscoasa(config-router)# address-family ipv4
```

```
ciscoasa(config-router-af)# neighbor 172.16.1.1 remote-as 4
ciscoasa(config-router-af)# neighbor 172.16.1.1 activate
```

The following example shows how to enable address exchange for address family IPv6 for all neighbors in the BGP peer group named group2 and for the BGP neighbor 7000::2:

```
Router(config)# address-family ipv6
Router(config-router-af)# neighbor group2 activate
Router(config-router-af)# neighbor 7000::2 activate
```

## Related Commands

| Command | Description |
|---|---|
| **neighbor remote-as** | Adds an entry to the BGP or multi-protocol BGP neighbor table. |

# neighbor advertise-map

To advertise the routes in the BGP table matching the configured route-map, use the neighbor advertise-map command in router configuration mode. To disable route advertisement, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **advertise-map** *map-name* { **exist-map** *map-name* | **non-exist-map** *map-name* } [ **check-all-paths** ]
**no neighbor** { *ip_address* | *ipv6-address* } **advertise-map** *map-name* { **exist-map** *map-name* | **non-exist-map** *map-name* } [ **check-all-paths** ]

## Syntax Description

| | |
|---|---|
| *ipv4_address* | Specifies the IPv4 address of the router that should receive conditional advertisements. |
| ipv6_address | Specifies the IPv6 address of the router that should receive conditional advertisements. |
| advertise-map map-name | Specifies the name of the route map that will be advertised if the conditions of the exist map or non-exist map are met. |
| exist-map map-name | Specifies the name of the exist-map that is compared with the routes in the BGP table to determine whether the advertise-map route is advertised or not. |
| **non-exist-map map-name** | Specifies the name of the non-exist-map that is compared with the routes in the BGP table to determine whether the advertise-map route is advertised or not. |
| check-all-paths | (Optional) Enables checking of all paths by the exist-map with a prefix in the BGP table. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | • — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.3(1) | This command was added. |

## Usage Guidelines

Use the neighbor advertise-map command to conditionally advertise selected routes. The routes (prefixes) that will be conditionally advertised are defined in two route maps: an advertise map and either an exist map or non-exist map.

The route map associated with the exist map or non-exist map specifies the prefix that the BGP speaker will track.

The route map associated with the advertise map specifies the prefix that will be advertised to the specified neighbor when the condition is met.

If an exist map is configured, the condition is met when the prefix exists in both the advertise map and the exist map.

If a non-exist map is configured, the condition is met when the prefix exists in the advertise map, but does not exist in the non-exist map.

If the condition is not met, the route is withdrawn and conditional advertisement does not occur. All routes that may be dynamically advertised or not advertised need to exist in the BGP routing table for conditional advertisement to occur.

## Examples

The following router configuration example configures BGP to check all :

```
ciscoasa(config)# router bgp 5000
ciscoasa(config-router)# address-family ipv4 unicast
ciscoasa(config-router-af)# neighbor 10.2.1.1 advertise-map MAP1 exist-map MAP2
ciscoasa(config-router-af)# neighbor 172.16.1.1 activate
```

The following address family configuration example configures BGP to conditionally advertise a prefix to the 10.1.1.1 neighbor using a non-exist map. If the prefix exists in MAP3 but not MAP4, the condition is met and the prefix is advertised.

```
ciscoasa(config)# router bgp 5000
ciscoasa(config-router)# address-family ipv4 unicast
ciscoasa(config-router-af)# neighbor 10.1.1.1 advertise-map MAP3 non-exist-map MAP4
```

The following peer group configuration example configures BGP to check all paths against the prefix to the BGP neighbor:

```
ciscoasa(config)# router bgp 5
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# redistribute static
ciscoasa(config-router-af)# neighbor route1 send-community both
ciscoasa(config-router-af)# neighbor route1 advertise-map MAP1 exist-map MAP2 check-all-paths
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters the address family configuration mode. |

# neighbor advertisement-interval

To set the minimum route advertisement interval (MRAI) between the sending of BGP routing updates, use the neighbor advertisement-interval command in address-family configuration mode. To restore the default value, use the no form of this command

**neighbor** { *ip_address* | *ipv6-address* } **advertisement-interval** *seconds*
**no neighbor** { *ip_address* | *ipv6-address* } **advertisement-interval** *seconds*

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router |
| seconds | Minimum time interval between sending BGP routing updates.<br>Valid values are between 0 and 600. |

## Command Default

eBGP sessions not in a VRF: 30 seconds

eBGP sessions in a VRF: 0 seconds

iBGP sessions: 0 seconds.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | • — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

When the MRAI is equal to 0 seconds, BGP routing updates are sent as soon as the BGP routing table changes.

## Examples

The following example sets the minimum time between sending BGP routing updates to 10 seconds:

```
ciscoasa(config-router-af)# neighbor 172.16.1.1 advertisement-interval 10
```

The following example sets the minimum time between sending BGPv6 routing updates to 100 seconds:

```
asa(config-router-af)# neighbor 2001::1 advertisement-interval 100
```

## Related Commands

| Command | Description |
|---|---|
| **neighbor remote-as** | Adds an entry to the BGP or multi-protocol BGP neighbor table. |
| neighbor activate | Enables information exchange with a BGP neighbor. |

# neighbor default-originate

To allow a BGP speaker (the local router) to send the default route 0.0.0.0 to a neighbor for use as a default route, use the neighbor default-originate command in address-family configuration mode. To send no route as a default, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **default-originate** [ **route-map** *route-map name* ]
**no neighbor** { *ip_address* | *ipv6-address* } **default-originate** [ **route-map** *route-map name* ]

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| route-map route-map name | (Optional) Name of the route map. The route map allows route 0.0.0.0 to be injected conditionally. |

## Command Default

No default route is sent to the neighbor.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | • — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

This command does not require the presence of 0.0.0.0 in the local router. When used with a route map, the default route 0.0.0.0 is injected if the route map contains a match ip address clause and there is a route that matches the IP access list exactly. The route map can also contain other match clauses.

You can use standard or extended access lists with the neighbor default-originate command.

## Examples

In the following example, the local router injects route 0.0.0.0 to the neighbor 72.16.2.3 unconditionally:

```
ciscoasa(config-router-af)# neighbor 172.16.2.3 default-originate
In the following example, the local router injects route 0.0.0.0 to the neighbor 2001::1:
asa(config-router-af)#neighbor 2001::1 default-originate route-map default-map
```

**Related Commands**

| Command | Description |
|---|---|
| **neighbor remote-as** | Adds an entry to the BGP or multi-protocol BGP neighbor table. |
| neighbor activate | Enables information exchange with a BGP neighbor. |

# neighbor description

To associate a description with a neighbor, use the neighbor description command in address-family configuration mode. To remove the description, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **description** *text*
**no neighbor** { *ip_address* | *ipv6-address* } **description** *text*

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| text | Text (up to 80 characters in length) that describes the neighbor. |

## Command Default

There is no description of the neighbor.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | • — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Examples

In the following example, the description of the neighbor is “peer with example.com”:

```
ciscoasa(config-router-af)# neighbor 172.16.2.3 description peer with example.com
```

In the following example, the description of the IPv6 neighbor is “peer with example.com”:

```
ciscoasa(config-router-af)#neighbor 2001::1 description peer with example.com
```

## Related Commands

| Command | Description |
|---|---|
| **neighbor remote-as** | Adds an entry to the BGP or multi-protocol BGP neighbor table. |
| neighbor activate | Enables information exchange with a BGP neighbor. |

# neighbor disable-connected-check

To disable connection verification to establish an eBGP peering session with a single-hop peer that uses a loopback interface, use the neighbor disable-connected-check command in address-family configuration mode. To enable connection verification for eBGP peering sessions, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **disable-connected-check**
**no neighbor** { *ip_address* | *ipv6-address* } **disable-connected-check**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |

## Command Default

A BGP routing process will verify the connection of single-hop eBGP peering session (TTL=254) to determine if the eBGP peer is directly connected to the same network segment by default. If the peer is not directly connected to same network segment, connection verification will prevent the peering session from being established.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | • — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

The neighbor disable-connected-check command is used to disable the connection verification process for eBGP peering sessions that are reachable by a single hop but are configured on a loopback interface or otherwise configured with a non-directly connected IP address.

This command is required only when the neighbor ebgp-multihop command is configured with a TTL value of 1. The address of the single-hop eBGP peer must be reachable. The neighbor update-source command must be configured to allow the BGP routing process to use the loopback interface for the peering session.

## Examples

In the following example, a single-hop eBGP peering session is configured between two BGP peers that are reachable on the same network segment through a local loopback interfaces on each router:

BGP Peer 1

```
ciscoasa(config)# interface loopback1
ciscoasa(config-if)# ip address 10.0.0.100 255.255.255
ciscoasa(config-if)# exit
ciscoasa(config)# router bgp 64512
ciscoasa(config-router)# neighbor 192.168.0.200 remote-as 65534
ciscoasa(config-router)# neighbor 192.168.0.200 ebgp-multihop 1
ciscoasa(config-router)# neighbor 192.168.0.200 update-source loopback2
ciscoasa(config-router)# neighbor 192.168.0.200 disable-connected-check
BGP Peer 2
ciscoasa(config)# interface loopback2
ciscoasa(config-if)# ip address 192.168.0.200 255.255.255
ciscoasa(config-if)# exit
ciscoasa(config)# router bgp 65534
ciscoasa(config-router)# neighbor 10.0.0.100 remote-as 64512
ciscoasa(config-router)# neighbor 10.0.0.100 ebgp-multihop 1
ciscoasa(config-router)# neighbor 10.0.0.100 update-source loopback1
ciscoasa(config-router)# neighbor 10.0.0.100 disable-connected-check
BGPv6 Peer
ciscoasa(config-router)# neighbor 2001::1 disable-connected-check
```

## Related Commands

| Command | Description |
|---|---|
| **neighbor remote-as** | Adds an entry to the BGP or multi-protocol BGP neighbor table. |
| neighbor ebgp-multihop | Accepts or initiates BGP connections to external peers residing on networks that are not directly connected. |

# neighbor distribute-list

To distribute BGP neighbor information as specified in an access list, use the neighbor distribute-list command in address-family configuration mode. To remove an entry, use the no form of this command.

**neighbor** *ip_address* **distribute-list** { **access-list-name** } { **in** | **out** }
**no neighbor** *ip_address* **distribute-list** { **access-list-name** } { **in** | **out** }

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| access-list-name | Name of a standard access list. |
| in | Access list is applied to incoming advertisements to that neighbor |
| out | Access list is applied to outgoing advertisements to that neighbor |

## Command Default

No BGP neighbor is specified.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | • — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |

## Usage Guidelines

Using a distribute list is one of several ways to filter advertisements. Advertisements can also be filtered by using the following methods:

- Autonomous system path filters can be configured with the ip as-path access-list and neighbor filter-list commands.
- The access-list (IP standard) commands can be used to configure standard access lists for the filtering of advertisement
- The route-map (IP) command can be used to filter advertisements. Route maps may be configured with autonomous system filters, prefix filters, access lists and distribute lists.

Standard access lists may be used to filter routing updates. However, in the case of route filtering when using classless inter-domain routing (CIDR), standard access lists do not provide the level of granularity that is necessary to configure advanced filtering of network addresses and masks.

## Examples

In the following example, BGP neighbor information in the standard access-list distribute-list-acl is applied to incoming advertisements to the neighbor 172.16.4.1.

```
ciscoasa(config)#router bgp 109
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# network 10.108.0.0
ciscoasa(config-router-af) neighbor 172.16.4.1 distribute-list distribute-list-acl in
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables information exchange with a BGP neighbor. |
| network | Specifies the networks to be advertised by BGP. |
| access-list permit | Specifies packets to forward. |
| access-list deny | Species packets to deny. |

# neighbor ebgp-multihop

To accept and attempt BGP connections to external peers residing on networks that are not directly connected, use the neighbor ebgp-multihop command in address-family configuration mode. To return to the default, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **ebgp-multihop** [ **ttl** ]
**no neighbor** { *ip_address* | *ipv6-address* } **ebgp-multihop**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| ttl | (Optional) Time to live.<br>Valid values are between 1 and 255 hops. |

## Command Default

Only directly connected neighbors are allowed.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | • — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

This feature should be used only under the guidance of Cisco technical support staff. To prevent the creation of loops through oscillating routes, the multihop will not be established if the only route to the multihop peer is the default route (0.0.0.0).

## Examples

The following example allows connections to or from neighbor 10.108.1.1, which resides on a network that is not directly connected:

```
ciscoasa(config)# router bgp 109
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af) neighbor 10.108.1.1 ebgp-multihop
```

The following example allows connections to or from neighbor 2001::1, which resides on a network that is not directly connected:

```
ciscoasa(config)# router bgp 3
ciscoasa(config-router)# address-family ipv6
ciscoasa(config-router-af) neighbor 12001::1 ebgp-multihop
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables information exchange with a BGP neighbor. |

# neighbor fall-over bfd (router bgp)

To configure BFD support for BGP so that BGP is registered to receive forwarding path detection failure messages from BFD, use the **fall-over** option when configuring the neighbor.

**neighbor** *ip_address* | *ipv6_address* **fall-over bfd**

## Syntax Description

| | |
|---|---|
| *ip_address/ipv6_address* | IP/IPv6 address of the neighbor router A.B.C.D/ X:X:X:X::X format. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Router BFD configuration | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.6(2) | This command was added. |

## Usage Guidelines

When configuring BFD support for BGP for multi-hop, ensure that the BFD map is already created for the source destination pair.

## Examples

The following example configures BFD support for the 172.16.10.2 and 1001::2 neighbors:

```
ciscoasa(config)# router bgp 100
ciscoasa(config-router)# address-family ipv4 unicast
ciscoasa(config-router-af)# neighbor 172.16.10.2 fall-over bfd
ciscoasa(config-router)# address-family ipv6 unicast
ciscoasa(config-router-af)# neighbor 1001::2 fall-over bfd
```

## Related Commands

| Command | Description |
|---|---|
| **authentication** | Configures authentication in a BFD template for single-hop and multi-hop sessions. |
| bfd echo | Enables BFD echo mode on the interface, |
| **bfd interval** | Configures the baseline BFD parameters on the interface. |
| bfd map | Configures a BFD map that associates addresses with multi-hop templates. |

| Command | Description |
|---|---|
| bfd slow-timers | Configures the BFD slow timers value. |
| bfd template | Binds a single-hop BFD template to an interface. |
| bfd-template single-hop \| multi-hop | Configures the BFD template and enters BFD configuration mode. |
| clear bfd counters | Clears the BFD counters. |
| echo | Configures echo in the BFD single-hop template. |
| show bfd drops | Displays the numbered of dropped packets in BFD. |
| show bfd map | Displays the configured BFD maps. |
| show bfd neighbors | Displays a line-by-line listing of existing BFD adjacencies. |
| show bfd summary | Displays summary information for BFD. |

# neighbor filter-list

To set up a BGP filter, use the neighbor filter-list command in address-family configuration mode. To disable this function, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **filter-list** *access-list-name* { **in** | **out** }
**no neighbor** { *ip_address* | *ipv6-address* } **filter-list** *access-list-name* { **in** | **out** }

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| access-list-name | Name of an autonomous system path access list. You define this access list with the as-path access-list command. |
| in | Access list is applied to incoming routes. |
| out | Access list is applied to outgoing routes. |

## Command Default

No BGP filter is used.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

This command establishes filters on both inbound and outbound BGP routes.

**Note** Do not apply both a neighbor distribute-list and a neighbor prefix-list command to a neighbor in any given direction (inbound or outbound). These two commands are mutually exclusive, and only one command ( neighbor distribute-list or neighbor prefix-list) can be applied to each inbound or outbound direction.

## Examples

In the following address-family configuration mode example, the BGP neighbor with IP address 172.16.1.1 is not sent advertisements about any path through or from the adjacent autonomous system 123:

```
ciscoasa(config)# as-path access-list as-path-acl deny _123_
ciscoasa(config)# as-path access-list as-path-acl deny ^123$
ciscoasa(config)#router bgp 109
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# network 10.108.0.0
ciscoasa(config-router-af)# neighbor 192.168.6.6 remote-as 123
ciscoasa(config-router-af)# neighbor 172.16.1.1 remote-as 47
ciscoasa(config-router-af)# neighbor 172.16.1.1 filter-list as-path-acl out
```

In the following address-family configuration mode example, the BGPv6 neighbor with IP address 2001::1 is not sent advertisements about any path through or from the adjacent autonomous system:

```
ciscoasa(config-router-af)# neighbor 2001::1 filter-list as-path-acl out
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables information exchange with a BGP neighbor. |
| neighbor remote-as | Adds an entry to the BGP or multi-protocol BGP neighbor table. |
| network | Specifies the network to be advertised by the BGP routing process. |

# neighbor ha-mode graceful-restart

To enable or disable the Border Gateway Protocol (BGP) graceful restart capability for a BGP neighbor, use the neighbor ha-mode graceful-restart command in the address-family configuration mode. To remove from the configuration the BGP graceful restart capability for a neighbor, use the no form of this command.

**neighbor** *ip_address* **ha-mode graceful-restart** [ **disable** ]
**no neighbor** *ip_address* **ha-mode graceful-restart**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor. |
| disable | (Optional) Disables BGP graceful restart capability for a neighbor. |

## Command Default

BGP graceful restart capability is disabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.3(1) | This command was added. |

## Usage Guidelines

The neighbor ha-mode graceful-restart command is used to enable or disable the graceful restart capability for an individual BGP neighbor. Use the disable keyword to disable the graceful restart capability when graceful restart has been previously enabled for the BGP peer.

The graceful restart capability is negotiated between nonstop forwarding (NSF)-capable and NSF-aware peers in OPEN messages during session establishment. If the graceful restart capability is enabled after a BGP session has been established, the session will need to be restarted with a soft or hard reset.

The graceful restart capability is supported by NSF-capable and NSF-aware ASA. An ASA that is NSF-capable can perform a stateful switchover (SSO) operation (graceful restart) and can assist restarting peers by holding routing table information during the SSO operation. An ASA that is NSF-aware functions like a router that is NSF-capable but cannot perform an SSO operation.

**Note** To enable the BGP graceful restart capability globally for all BGP neighbors, use the bgp graceful-restart command. When the BGP graceful restart capability is configured for an individual neighbor, each method of configuring graceful restart has the same priority, and the last configuration instance is applied to the neighbor.

Use the show bgp neighbors command to verify the BGP graceful restart configuration for BGP neighbors.

## Examples

The following example enables the BGP graceful restart capability for the BGP neighbor, 172.21.1.2:

```
Ciscoasa(config)# router bgp 45000
Ciscoasa(config-router)# bgp log-neighbor-changes
Ciscoasa(config-router)# address-family ipv4 unicast
Ciscoasa(config-router-af)# neighbor 172.21.1.2 remote-as 45000
Ciscoasa(config-router-af)# neighbor 172.21.1.2 activate
Ciscoasa(config-router-af)# neighbor 172.21.1.2 ha-mode graceful-restart
```

## Related Commands

| Command | Description |
|---|---|
| **bgp graceful-restart** | Enables or disables he BGP graceful restart capability globally for all BGP neighbors. |
| show bgp neighbors | Displays information about the TCP and BGP connections to neighbors. |

# neighbor local-as

To customize the AS_PATH attribute for routes received from an external Border Gateway Protocol (eBGP) neighbor, use the neighbor local-as command in address-family configuration mode. To disable AS_PATH attribute customization, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **local-as** [ *autonomous-system-number* [ **no-prepend** [ **replace-as** [ **dual-as** ] ] ]
**no neighbor** { *ip_address* | *ipv6-address* } **local-as**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| autonomous-system-number | (Optional) Number of an autonomous system to prepend to the AS_PATH attribute. The range of values for this argument is any valid autonomous system number from 1 to 65535.<br>**Note** With this argument, you cannot specify the autonomous system number from the local BGP routing process or from the network of the remote peer.<br>For more details about autonomous system number formats, see the router bgp command. |
| no-prepend | (Optional) Does not prepend the local autonomous system number to any routes received from the eBGP neighbor. |
| replace-as | (Optional) Replaces the real autonomous system number with the local autonomous system number in the eBGP updates. The autonomous system number from the local BGP routing process is not prepended. |
| dual-as | (Optional) Configures the eBGP neighbor to establish a peering session using the real autonomous system number (from the local BGP routing process) or by using the autonomous system number configured with the autonomous-system-number argument (local-as). |

## Command Default

The autonomous system number from the local BGP routing process is prepended to all external routes by default.

## Command Modes

The following table shows the modes in which you can enter the command:

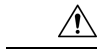
| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

The neighbor local-as command is used to customize the AS_PATH attribute by adding and removing autonomous system numbers for routes received from eBGP neighbors. The configuration of this command allows a router to appear to external peers as a member of another autonomous system for the purpose of autonomous system number migration. This feature simplifies the process of changing the autonomous system number in a BGP network by allowing the network operator to migrate customers to new configurations during normal service windows without disrupting existing peering arrangements.



**Caution** BGP prepends the autonomous system number from each BGP network that a route traverses to maintain network reachability information and to prevent routing loops. This command should be configured only for autonomous system migration, and should be de-configured after the transition has been completed. This procedure should be attempted only by an experienced network operator. Routing loops can be created through improper configuration.

This command can be used for only true eBGP peering sessions. This command does not work for two peers in different sub-autonomous systems of a confederation.

To ensure a smooth transition, we recommend that all BGP speakers within an autonomous system that is identified using a 4-byte autonomous system number, be upgraded to support 4-byte autonomous system numbers.

## Examples

### Local-AS Example

The following example establishes peering between Router 1 and Router 2 through autonomous system 300, using the local-as feature:

**Router 1 (Local router)**

```
ciscoasa(config)# router bgp 100
ciscoasa(config-router)# address-family ipv4 unicast
ciscoasa(config-router-af)# neighbor 172.16.1.1 remote-as 200
ciscoasa(config-router-af)# neighbor 172.16.1.1 local-as 300
Router 2 (Remote router)
ciscoasa(config)# router bgp 200
ciscoasa(config-router)# address-family ipv4 unicast
ciscoasa(config-router-af)# neighbor 10.0.0.1 remote-as 300
```

### No-prepend keyword configuration Example

The following example configures BGP to not prepend autonomous system 500 to routes received from the 192.168.1.1 neighbor:

```
ciscoasa(config)# router bgp 400
ciscoasa(config-router)# address-family ipv4
```

```
ciscoasa(config-router-af)# network 192.168.0.0
ciscoasa(config-router-af)# neighbor 192.168.1.1 local-as 500 no-prepend
```

### Replace-as keyword configuration Example

The following example strips private autonomous system 64512 from outbound routing updates for the 172.20.1.1 neighbor and replaces it with autonomous system 600:

```
ciscoasa(config)# router bgp 64512

ciscoasa(config-router)# address-family ipv4

ciscoasa(config-router-af)# neighbor 172.20.1.1 local-as 600 no-prepend replace-as

ciscoasa(config-router-af)# neighbor 172.20.1.1 remove-private-as
```

### Dual-as keyword configuration Example

The following examples show the configurations for two provider networks and one customer network. Router 1 belongs to autonomous system 100, and Router 2 belongs to autonomous system 200. Autonomous system 200 is being merged into autonomous system 100. This transition needs to occur without interrupting service to Router 3 in autonomous system 300 (customer network). The neighbor local-as command is configured on router 1 to allow Router 3 to maintain peering with autonomous system 200 during this transition. After the transition is complete, the configuration on Router 3 can be updated to peer with autonomous system 100 during a normal maintenance window or during other scheduled downtime.

**Router 1 Configuration (Local Provider Network)**

```
ciscoasa(config)# router bgp 100
ciscoasa(config-router)# address-family pv4
ciscoasa(config-router-af)# no synchronization
ciscoasa(config-router-af)# bgp router-id 100.0.0.11
ciscoasa(config-router-af)# neighbor 10.3.3.33 remote-as 300
ciscoasa(config-router-af)# neighbor 10.3.3.33 local-as 200 no-prepend replace-as dual-as
```

**Router 2 Configuration (Remote Provider Network)**

```
ciscoasa(config)# router bgp 200

ciscoasa(config-router)# address-family pv4
ciscoasa(config-router-af)# bgp router-id 100.0.0.11
 ciscoasa(config-router-af)# neighbor 10.3.3.33 remote-as 300
```

**Router 3 Configuration (Remote Customer Network)**

```
ciscoasa(config)# router bgp 300
ciscoasa(config-router)# address-family pv4
ciscoasa(config-router-af)# bgp router-id 100.0.0.3
ciscoasa(config-router-af)# neighbor 10.3.3.11 remote-as 200
```

To complete the migration after the two autonomous systems have merged, the peering session is updated on Router 3:

```
ciscoasa(config-router-af)# neighbor 10.3.3.11 remote-as 100
```

BGPv6 configuration

```
ciscoasa(config-router-af)# neighbor 2001::1 local-as 500 no-prepend
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| bgp router-id | Configure a fixed router ID for the local Border Gateway Protocol (BGP) routing process. |
| neighbor activate | Enables information exchange with a BGP neighbor. |
| neighbor remote-as | Adds an entry to the BGP or multi-protocol BGP neighbor table. |
| network | Specifies the network to be advertised by the BGP routing process. |
| synchronization | Enables the synchronization between BGP and your Interior Gateway Protocol (IGP) system |

# neighbor maximum-prefix

To control how many prefixes can be received from a neighbor, use the neighbor maximum-prefix command in address-family configuration mode. To disable this function, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **maximum-prefix** *maximum* [ *threshold* ] [ **restart** *restart-interval* ] [ **warning-only** ]
**no neighbor** { *ip_address* | *ipv6-address* } **maximum-prefix** *maximum*

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| maximum | Maximum number of prefixes allowed from this neighbor. |
| threshold | (Optional) Integer specifying at what percentage of maximum the router starts to generate a warning message. The range is from 1 to 100; the default is 75 (percent). |
| restart | (Optional) Configures the router that is running BGP to automatically reestablish a peering session that has been disabled because the maximum-prefix limit has been exceeded. The restart timer is configured with the restart-interval argument. |
| restart-interval | (Optional) Time interval (in minutes) that a peering session is reestablished. The range is from 1 to 65535 minutes. |
| warning-only | (Optional) Allows the router to generate a log message when the maximum is exceeded, instead of terminating the peering. |

## Command Default

This command is disabled by default. There is no limit on the number of prefixes.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

This command allows you to configure a maximum number of prefixes that a BGP router is allowed to receive from a peer. It adds another mechanism (in addition to distribute lists, filter lists, and route maps) to control prefixes received from a peer.

When the number of received prefixes exceeds the maximum number configured, the router terminates the peering (by default). However, if the warning-only keyword is configured, the router instead only sends a log message, but continues peering with the sender. If the peer is terminated, the peer stays down until the clear bgp command is issued.

## Examples

The following example sets the maximum number of prefixes allowed from the neighbor at 192.168.6.6 to 1000:

```
ciscoasa(config)# router bgp 109
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# network 10.108.0.0
ciscoasa(config-router-af)# neighbor 192.168.6.6 maximum-prefix 1000
```

The following example sets the maximum number of prefixes allowed from the neighbor at 2001::1 to 1000:

```
ciscoasa(config-router-af)# neighbor 2001::1 maximum-prefix 1000
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables information exchange with a BGP neighbor. |
| network | Specifies the network to be advertised by the BGP routing process. |

# neighbor next-hop-self

To configure the router as the next hop for a BGP-speaking neighbor, use the neighbor next-hop-self command in address-family configuration mode. To disable this feature, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **next-hop-self**
**no neighbor** { *ip_address* | *ipv6-address* } **next-hop-self**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| warning-only | (Optional) Allows the router to generate a log message when the maximum is exceeded, instead of terminating the peering. |

## Command Default

This command is disabled by default.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

This command is useful in unmeshed networks (such as Frame Relay or X.25) where BGP neighbors may not have direct access to all other neighbors on the same IP subnet.

## Examples

The following example forces all updates destined for 10.108.1.1 to advertise this router as the next hop:

```
ciscoasa(config)# router bgp 109
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# neighbor 10.108.1.1 next-hop-self
```

The following example forces all updates destined for 2001::1 to advertise this router as the next hop:

```
ciscoasa(config-router-af)#neighbor 2001::1 next-hop-selfs
```

**Related Commands**

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables information exchange with a BGP neighbor. |

# neighbor password

To enable message digest5 (MD5) authentication on a TCP connection between two BGP peers, use the neighbor password command in address-family configuration mode. To disable this function, use the no form of this command

**neighbor** { *ip_address* | *ipv6-address* } **password** [ **0-7** ] *string*
**no neighbor** { *ip_address* | *ipv6-address* } **password**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| string | Case-sensitive password of up to 25 characters in length.<br>The first character cannot be a number. The string can contain any alphanumeric characters, including spaces. You cannot specify a password in the format number-space-anything. The space after the number can cause authentication to fail. |
| 0-7 | (Optional) Encryption type. 0-6 is without encryption. 7 is used for encryption. |

## Command Default

MD5 is not authenticated on a TCP connection between two BGP peers.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

You can configure MD5 authentication between two BGP peers, meaning that each segment sent on the TCP connection between the peers is verified. MD5 authentication must be configured with the same password on both BGP peers; otherwise, the connection between them will not be made. Configuring MD5 authentication causes the ASA software to generate and check the MD5 digest of every segment sent on the TCP connection.

When configuring you can provide a case-sensitive password of up to 25 characters regardless of whether the service password-encryption command is enabled. If the length of password is more than 25 characters, an error message is displayed and the password is not accepted. The string can contain any alphanumeric characters, including spaces. A password cannot be configured in the number-space-anything format. The space after the

number can cause authentication to fail. You can also use any combination of the following symbolic characters along with alphanumeric characters:

` ~ ! @ # $ % ^ & * ( ) - _ = + | \ } ] { [ " ` : ; / > < . , ?

**Caution** If the authentication string is configured incorrectly, the BGP peering session will not be established. We recommend that you enter the authentication string carefully and verify that the peering session is established after authentication is configured.

If a router has a password configured for a neighbor, but the neighbor router does not, a message such as the following will appear on the console while the routers attempt to establish a BGP session between them:

%TCP-6-BADAUTH: No MD5 digest from [peer's IP address]:11003 to [local router's IP address]:179

Similarly, if the two routers have different passwords configured, a message such as the following will appear on the screen:

%TCP-6-BADAUTH: Invalid MD5 digest from [peer's IP address]:11004 to [local router's IP address]:179

**Configuring an MD5 Password in an Established BGP Session**

If you configure or change the password or key used for MD5 authentication between two BGP peers, the local router will not tear down the existing session after you configure the password. The local router will attempt to maintain the peering session using the new password until the BGP hold-down timer expires. The default time period is 180 seconds. If the password is not entered or changed on the remote router before the hold-down timer expires, the session will time out.

**Note** Configuring a new timer value for the hold-down timer will only take effect after the session has been reset. So, it is not possible to change the configuration of the hold-down timer to avoid resetting the BGP session.

## Examples

The following example configures MD5 authentication for the peering session with the 10.108.1.1 neighbor. The same password must be configured on the remote peer before the hold-down timer expires:

```
ciscoasa(config)# router bgp 109
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# neighbor 10.108.1.1 password bla4u00=2nkq
```

The following example configures a password for more than 25 characters when the service password-encryption command is disabled.

```
ciscoasa(config)# router bgp 200
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# bgp router-id 2.2.2.2
ciscoasa(config-router-af)# neighbor remote-as 3
 ciscoasa(config-router-af)# neighbor 209.165.200.225 password 12345678912345678912345678901

    % BGP: Password length must be less than or equal to 25.
ciscoasa(config-router-af)# do show run | i password
 no service password-encryption
  neighbor 209.165.200.225 password 123456789123456789123456
```

In the following example an error message occurs when you configure a password for more than 25 characters when the service password-encryption command is enabled.

```
Router(config)# service password-encryption
Router(config)# router bgp 200
Router(config-router)# bgp router-id 2.2.2.2
Router(config-router)# neighbor 209.165.200.225 remote-as 3
Router(config-router)# neighbor 209.165.200.225 password 12345678912345678912345678901

% BGP: Password length must be less than or equal to 25.
 Router(config-router)# do show run | i password  service password-encryption
 neighbor 209.165.200.225 password 123456789123456789123456
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables information exchange with a BGP neighbor. |
| bgp router-id | Configure a fixed router ID for the local Border Gateway Protocol (BGP) routing process. |
| neighbor remote-as | Add an entry to the BGP or multiprotocol BGP neighbor table. |

# neighbor prefix-list

To prevent distribution of Border Gateway Protocol (BGP) neighbor information as specified in a prefix list, use the neighbor prefix-list command in address-family configuration mode. To remove a filter list, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **prefix-list** *prefix-list-name* { **in** | **out** }
**no neighbor** { *ip_address* | *ipv6-address* } **prefix-list** *prefix-list-name* { **in** | **out** }

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-adress | IPv6 address of the neighbor router. |
| prefix-list-name | Name of a prefix list. |
| in | Filter list is applied to incoming advertisements from that neighbor. |
| out | Filter list is applied to outgoing advertisements to that neighbor. |

## Command Default

All external and advertised address prefixes are distributed to BGP neighbors.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

Using prefix lists is one of three ways to filter BGP advertisements. You can also use AS-path filters, defined with the ip as-path access-list global configuration command and used in the neighbor filter-list command to filter BGP advertisements. The third way to filter BGP advertisements uses access or prefix lists with the neighbor distribute-list command.

**Note** Do not apply both a neighbor distribute-list and a neighbor prefix-list command to a neighbor in any given direction (inbound or outbound). These two commands are mutually exclusive, and only one command (neighbor distribute-list or neighbor prefix-list) can be applied to each inbound or outbound direction..

## Examples

The following address-family configuration mode example applies the prefix list named abc to incoming advertisements from neighbor 10.23.4.1:

```
ciscoasa(config)# router bgp 65200
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# network 192.168.1.2
ciscoasa(config-router-af)# neighbor 10.23.4.1 prefix-list abc in
```

The following address family router configuration mode example applies the prefix list named CustomerA to outgoing advertisements to neighbor 10.23.4.3:

```
ciscoasa(config)# router bgp 64800
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# network 192.168.3.6
ciscoasa(config-router-af)# neighbor 10.23.4.3 prefix-list CustomerA out
The following address family router configuration mode example applies the prefix list named
 CustomerA to outgoing advertisements to neighbor 2001::1:
ciscoasa(config-router-af)#neighbor 2001::1 prefix-list CustomerA out
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables information exchange with a BGP neighbor. |
| network | Specifies the network to be advertised by the BGP routing process. |

# neighbor remote-as

To add an entry to the BGP or multiprotocol BGP neighbor table, use the neighbor remote-as command in the address-family configuration mode. To remove an entry from the table, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **remote-as** *autonomous-system-number*
**no neighbor** { *ip_address* | *ipv6-address* } **remote-as** *autonomous-system-number*

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| autonomous-system-number | Number of an autonomous system to which the neighbor belongs in the range from 1 to 65535.<br>For more details about autonomous system number formats, see the router bgp command.<br>When used with the alternate-as keyword, up to five autonomous system numbers may be entered. |

## Command Default

There are no BGP or multiprotocol BGP neighbor peers.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

Specifying a neighbor with an autonomous system number that matches the autonomous system number specified in the router bgp global configuration command identifies the neighbor as internal to the local autonomous system. Otherwise, the neighbor is considered external.

By default, neighbors that are defined using the neighbor remote-as command in router configuration mode exchange only unicast address prefixes.

Use the alternate-as keyword is used to specify up to five alternate autonomous systems in which a dynamic BGP neighbor may be identified. BGP dynamic neighbor support allows BGP peering to a group of remote neighbors that are defined by a range of IP addresses. BGP dynamic neighbors are configured using a range

of IP addresses and BGP peer groups. After a subnet range is configured and associated with a BGP peer group using the bgp listen command and a TCP session is initiated for an IP address in the subnet range, a new BGP neighbor is dynamically created as a member of that group. The new BGP neighbor will inherit any configuration or templates for the group.

Cisco implementation of 4-byte autonomous system numbers uses asplain—65538 for example—as the default regular expression match and output display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain format and the asdot format as described in RFC 5396. To change the default regular expression match and output display of 4-byte autonomous system numbers to asdot format, use the bgp asnotation dot command followed by the clear bgp * command to perform a hard reset of all current BGP sessions.

## Examples

The following example specifies that a router at the address 10.108.1.2 is an internal BGP (iBGP) neighbor in autonomous system number 65200:

```
ciscoasa(config)# router bgp 65200
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# network 10.108.0.0
ciscoasa(config-router-af)# neighbor 10.108.1.2 remote-as 65200
```

The following example assigns a BGP router to autonomous system 65400, and two networks are listed as originating in the autonomous system. Then the addresses of three remote routers (and their autonomous systems) are listed. The router being configured will share information about networks 10.108.0.0 and 192.168.7.0 with the neighbor routers. The first router is a remote router in a different autonomous system from the router on which this configuration is entered (an eBGP neighbor); the second neighbor remote-as command shows an internal BGP neighbor (with the same autonomous system number) at address 10.108.234.2; and the last neighbor remote-as command specifies a neighbor on a different network from the router on which this configuration is entered (also an eBGP neighbor).

```
ciscoasa(config)# router bgp 65400
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# network 10.108.0.0
ciscoasa(config-router-af)# network 192.168.7.0
ciscoasa(config-router-af)# neighbor 10.108.200.1 remote-as 65200
ciscoasa(config-router-af)# neighbor 10.108.234.2 remote-as 65400
ciscoasa(config-router-af)# neighbor 172.29.64.19 remote-as 65300
```

The following example configures neighbor 10.108.1.1 in autonomous system 65001 to exchange only unicast routes:

```
ciscoasa(config)# router bgp 65001
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# neighbor 10.108.1.1 remote-as 65001
ciscoasa(config-router-af)# neighbor 172.31 1.2 remote-as 65001
ciscoasa(config-router-af)# neighbor 172.16.2.2 remote-as 65002
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| network | Specifies the network to be advertised by the BGP routing process. |

| Command | Description |
| --- | --- |
| neighbor remove private-as | Removes private autonomous system numbers from the eBGP outbound routing updates. |

# neighbor remove-private-as

To remove private autonomous system numbers from the eBGP outbound routing updates, use the neighbor remove-private-as command in address-family configuration mode. To disable this function, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **remove-private-as** [ **all** [ **replace-as** ] ]
**no neighbor** { *ip_address* | *ipv6-address* } **remove-private-as** [ **all** [ **replace-as** ] ]

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| all | (Optional) Removes all private AS numbers from the AS path in outgoing updates. |
| replace-as | (Optional) As long as the all keyword is specified, the replace-as keyword causes all private AS numbers in the AS path to be replaced with the router's local AS number. |

## Command Default

No private AS numbers are removed from the AS path.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

This command is available for external BGP (eBGP) neighbors only. The private AS values are 64512 to 65535. When an update is passed to the external neighbor, if the AS path includes private AS numbers, the software will drop the private AS numbers

- The neighbor remove-private-as command removes private AS numbers from the AS path even if the path contains both public and private ASNs
- The neighbor remove-private-as command removes private AS numbers even if the AS path contains only private AS numbers. There is no likelihood of a 0-length AS path because this command can be applied to eBGP peers only, in which case the AS number of the local router is appended to the AS path. The neighbor remove-private-as command removes private AS numbers even if the private ASNs appear before the Confederation segments in the AS path.

- Upon removing private AS numbers from the AS path, the path length of prefixes being sent out will decrease. Because the AS path length is a key element of BGP best path selection, it might be necessary to retain the path length. The replace-as keyword ensures that the path length is retained by replacing all removed AS numbers with the local router's AS number.
- The feature can be applied to neighbors per address family. Therefore, you can apply the feature to a neighbor in one address family and not in another, affecting update messages on the outbound side for only the address family for which the feature is configured.

## Examples

The following example shows a configuration that removes the private AS number from the updates sent to 172.16.2.33. The result is that the AS path for the paths advertised by 10.108.1.1 through AS 100 will contain only “100” (as seen by autonomous system 2051).

```
ciscoasa(config)# router bgp 100
ciscoasa(config-router)# address-family ipv4 unicast
ciscoasa(config-router-af)# neighbor 10.108.1.1 description peer with private-as
ciscoasa(config-router-af)# neighbor 10.108.1.1 remote-as 65001
ciscoasa(config-router-af)# neighbor 172.16.2.33 description eBGP peer

ciscoasa(config-router-af)# neighbor 172.16.2.33 remote-as 2051
ciscoasa(config-router-af)# neighbor 172.16.2.33 remove-private-as


Router-in-AS100# show bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 15
Paths: (1 available, best #1)
 Advertised to non peer-group peers:
   172.16.2.33
 65001
    10.108.1.1 from 10.108.1.1
     Origin IGP, metric 0, localpref 100, valid, external, best
Router-in-AS2501# show bgp 10.0.0.0
BGP routing table entry for 10.0.0.0/8, version 3
Paths: (1 available, best #1)
  Not advertised to any peer
  2
    172.16.2.32 from 172.16.2.32
     Origin IGP, metric 0, localpref 100, valid, external, best
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor description | Associates a description with a neighbor |
| neighbor remote-as | Adds a BGP or multi-protocol BGP routing entry to the routing table. |

# neighbor route-map

To apply a route map to incoming or outgoing routes, use the neighbor route-map command in address-family configuration mode. To remove a route map, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **route-map** *map-name* { **in** | **out** }
**no neighbor** { *ip_address* | *ipv6-address* } **route-map** *map-name* { **in** | **out** }

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| map-name | Name of a route-map. |
| in | Applies route map to incoming routes. |
| out | Applies route map to outgoing routes. |

## Command Default

No route maps are applied to a peer.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

When specified in address-family configuration mode, this command applies a route map to that particular address family only. When specified in router configuration mode, this command applies a route map to IPv4 unicast routes only.

If an outbound route map is specified, it is proper behavior to only advertise routes that match at least one section of the route map.

## Examples

The following example applies a route map named internal-map to a BGP incoming route from 172.16.70.24:

```
ciscoasa(config)# router bgp 5
ciscoasa(config-router)# address-family ipv4
```

```
ciscoasa(config-router-af)# neighbor 172.16.70.24 route-map internal-map in
ciscoasa(config-router-af)# route-map internal-map
ciscoasa(config-route-map)# match as-path 1
ciscoasa(config-route-map)# set local-preference 100
```

The following example applies a route map named internal-map to BGP incoming route from 2001::1:

```
ciscoasa(config-router-af)# neighbor 2001::1 route-map internal-map in
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| match as-path | Matches a BGP autonomous system path that is specified by an access list |
| route-map | Defines the conditions for redistributing routes from one routing protocol into another. |
| match as-path | Match a BGP autonomous system path that is specified by an access list. |
| set local-preference | Specify a preference value for the autonomous system path. |

# neighbor send-community

To specify that a communities attribute should be sent to a BGP neighbor, use the neighbor send-community command in address-family configuration mode. To remove the entry, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **send-community**
**no neighbor** { *ip_address* | *ipv6-address* } **send-community**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |

## Command Default

No communities attribute is sent to any neighbor.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Examples

In the following address-family configuration mode example, the router belongs to autonomous system 109 and is configured to send the communities attribute to its neighbor at IP address 172.16.70.23:

```
ciscoasa(config)# router bgp 109
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# neighbor 172.16.70.23 send-community
```

In the following example, the router is configured to send the communities attribute to its neighbor at IP address 2001::1:

```
ciscoasa(config-router-af)# neighbor 2001::1 send-community
```

**Related Commands**

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |

# neighbor shutdown

To disable a neighbor, use the neighbor shutdown command in address-family configuration mode. To re-enable the neighbor, use the no form of this command.

**neighbor** *ip_address* **shutdown**
**no neighbor** *ip_address* **shutdown**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |

## Command Default

No change is made to the status of any BGP neighbor.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |

## Usage Guidelines

The neighbor shutdown command terminates any active session for the specified neighbor and removes all associated routing information .

To display a summary of BGP neighbors, use the show bgp summary command. Those neighbors with an Idle status and the Admin entry have been disabled by the neighbor shutdown command.

'State/PfxRcd' shows the current state of the BGP session or the number of prefixes the router has received from a neighbor. When the maximum number (as set by the neighbor maximum-prefix command) is reached, the string 'PfxRcd' appears in the entry, the neighbor is shut down, and the connection is idle.

## Examples

The following example disables any active session for the neighbor 172.16.70.23:

```
ciscoasa(config-router-af)# neighbor 172.16.70.23 shutdown
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables information exchange with a BGP neighbor. |

| Command | Description |
| --- | --- |
| show bgp summary | Displays a summary of BGP neighbor status. |

# neighbor timers

To set the timers for a specific BGP peer, use the neighbor timers command in address-family configuration mode. To clear the timers for a specific BGP peer , use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **timers** *keepalive holdtime* [ *min-holdtime* ]

**no neighbor** { *ip_address* | *ipv6-address* } **timers**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| keepalive | Frequency (in seconds) with which the ASA software sends keepalive messages to its peer. The default is 60 seconds.The range is from 0 to 65535. |
| holdtime | Interval (in seconds) after not receiving a keepalive message that the software declares a peer dead. The default is 180 seconds. The range is from 0 to 65535. |
| min-holdtime | (Optional) Interval (in seconds) specifying the minimum acceptable hold-time from a BGP neighbor. The minimum acceptable hold-time must be less than, or equal to, the interval specified in the holdtime argument. The range is from 0 to 65535. |

## Command Default

Keepalive time: 60 seconds

Holdtime: 180 seconds

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

- The timers configured for a specific neighbor override the timers configured for all BGP neighbors using the timers bgp command.
- When configuring the holdtime argument for a value of less than twenty seconds, the following warning is displayed: A hold time of less than 20 seconds increases the chances of peer flapping.

- If the minimum acceptable hold-time interval is greater than the specified hold-time, a notification is displayed: Minimum acceptable hold time should be less than or equal to the configured hold time.

**Note** When the minimum acceptable hold-time is configured on a BGP router, a remote BGP peer session is established only if the remote peer is advertising a hold-time that is equal to, or greater than, the minimum acceptable hold-time interval. If the minimum acceptable hold-time interval is greater than the configured hold-time, the next time the remote session tries to establish, it will fail and the local router will send a notification stating "unacceptable hold time."

## Examples

The following example changes the keepalive timer to 70 seconds and the hold-time timer to 210 seconds for the BGP peer 192.168.47.0:

```
ciscoasa(config-router-af)# neighbor 192.168.47.0 timers 70 210
```

The following example changes the keepalive timer to 70 seconds, the hold-time timer to 130 seconds, and the minimum hold-time interval to 100 seconds for the BGP peer 192.168.1.2:

```
ciscoasa(config-router-af)# neighbor 192.168.1.2 timers 70 130 100
```

The following example changes the keepalive timer to 70 seconds and the hold-time timer to 210 seconds, for the BGP peer 2001::1:

```
ciscoasa(config-router-af)# neighbor 2001::1 timers 70 210
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables exchange of information with a BGP neighbor. |

# neighbor transport

To enable a TCP transport session option for a Border Gateway Protocol (BGP) session, use the neighbor transport command in router or address-family configuration mode. To disable a TCP transport session option for a BGP session, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **transport** { **connection-mode** { **active** | **passive** } | **path-mtu-discovery** [ **disable** ] }
**no neighbor** { *ip_address* | *ipv6-address* } **transport** { **connection-mode** { **active** | **passive** } | **path-mtu-discovery** [ **disable** ] }

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| connection-mode | Specifies the type of connection - active or passive. |
| active | Specifies an active connection. |
| passive | Specifies a passive connection. |
| path-mtu-discovery | Enables TCP transport path maximum transmission unit (MTU) discovery. TCP path MTU discovery is enabled by default. |
| disable | Disables TCP path MTU discovery. |

## Command Default

If this command is not configured, TCP path MTU discovery is enabled by default, but no other TCP transport session options are enabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

This command is used to specify various transport options. An active or passive transport connection can be specified for a BGP session. TCP transport path MTU discovery can be enabled to allow a BGP session to take advantage of larger MTU links. Use the show bgp neighbors command to determine whether TCP path

MTU discovery is enabled.If you use the disable keyword to disable discovery, discovery is also disabled on any peer that inherits the template in which you disabled discovery.

## Examples

The following example shows how to configure the TCP transport connection to be active for a single internal BGP (iBGP) neighbor:

```
ciscoasa(config)# router bgp 45000
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# neighbor 172.16.1.2 remote-as 45000
ciscoasa(config-router-af)# neighbor 172.16.1.2 activate
ciscoasa(config-router-af)# neighbor 172.16.1.2 transport connection-mode active
```

The following example shows how to configure the TCP transport connection to be passive for a single external BGP (eBGP) neighbor:

```
ciscoasa(config)# router bgp 45000
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# neighbor 192.168.1.2 remote-as 40000
ciscoasa(config-router-af)# neighbor 192.168.1.2 activate
ciscoasa(config-router-af)# neighbor 192.168.1.2 transport connection-mode passive
```

The following example shows how to disable TCP path MTU discovery for a single BGP neighbor:

```
ciscoasa(config)# router bgp 45000
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# neighbor 172.16.1.2 remote-as 45000
ciscoasa(config-router-af)# neighbor 172.16.1.2 activate
ciscoasa(config-router-af)# no neighbor 172.16.1.2 transport path-mtu-discovery
```

The following example shows how to configure the TCP transport connection to be active for a single BGPv6 neighbor:

```
ciscoasa(config-router-af)#neighbor 2001::1 transport connection-mode active
```

The following example shows how to enable TCP path MTU discovery for a single BGPv6 neighbor:

```
ciscoasa(config-router-af)#neighbor 2001::1 transport path-mtu-discovery
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables exchange of information with a BGP neighbor. |
| neighbor remote-as | Adds an entry to the BGP or multi-protocol BGP routing table. |
| show bgp neighbor | Displays information about BGP neighbors |

# neighbor ttl-security

To secure a Border Gateway Protocol (BGP) peering session and to configure the maximum number of hops that separate two external BGP (eBGP) peers, use the neighbor ttl-security command in address-family configuration mode. To disable this feature, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **ttl-security hops** *hop-count*
**no neighbor** { *ip_address* | *ipv6-address* } **ttl-security hops** *hop-count*

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| hop-count | Number of hops that separate the eBGP peers. The TTL value is calculated by the router from the configured hop-count argument.<br>Valid values are a number between 1 and 254. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

The neighbor ttl-security command provides a lightweight security mechanism to protect BGP peering sessions from CPU utilization-based attacks. These types of attacks are typically brute force Denial of Service (DoS) attacks that attempt to disable the network by flooding the network with IP packets that contain forged source and destination IP addresses in the packet headers.

This feature leverages designed behavior of IP packets by accepting only IP packets with a TTL count that is equal to or greater than the locally configured value. Accurately forging the TTL count in an IP packet is generally considered to be impossible. Accurately forging a packet to match the TTL count from a trusted peer is not possible without internal access to the source or destination network.

This feature should be configured on each participating router. It secures the BGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router. When this feature is enabled, BGP will establish or maintain a session only if the TTL value in the IP packet header is equal to or greater

than the TTL value configured for the peering session. This feature has no effect on the BGP peering session, and the peering session can still expire if keepalive packets are not received. If the TTL value in a received packet is less than the locally configured value, the packet is silently discarded and no Internet Control Message Protocol (ICMP) message is generated. This is designed behavior; a response to a forged packet is not necessary.

To maximize the effectiveness of this feature, the hop-count value should be strictly configured to match the number of hops between the local and external network. However, you should also take path variation into account when configuring this feature for a multihop peering session.

The following restrictions apply to the configuration of this command:

- This feature is not supported for internal BGP (iBGP) peers.
- The neighbor ttl-security command cannot be configured for a peer that is already configured with the neighbor ebgp-multihop command. The configuration of these commands is mutually exclusive, and only one of these commands is needed to enable a multihop eBGP peering session. An error message will be displayed in the console if you attempt to configure both commands for the same peering session.
- The effectiveness of this feature is reduced in large-diameter multihop peerings. In the event of a CPU utilization-based attack against a BGP router that is configured for large-diameter peering, you may still need to shut down the affected peering sessions to handle the attack.
- This feature is not effective against attacks from a peer that has been compromised inside of your network. This restriction also includes peers that are on the network segment between the source and destination network.

## Examples

The following example sets the hop count to 2 for a directly connected neighbor. Because the hop-count argument is set to 2, BGP will accept only IP packets with a TTL count in the header that is equal to or greater than 253. If a packet is received with any other TTL value in the IP packet header, the packet will be silently discarded.

```
ciscoasa(config-router-af)# neighbor 10.0.0.1 ttl-security hops 2
```

The following example sets the hop count to 2 for a directly connected BGPv6 neighbor.

```
ciscoasa(config-router-af)#neighbor 2001::1 ttl-security hops 2
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables exchange of information with a BGP neighbor. |
| neighbor ebgp-multihop | Accepts and attempts BGP connections to external peers residing on networks that are not directly connected |

# neighbor update-source

To configure an interface as the source for a BGP-speaking neighbor, use the **neighbor update-source** command in address-family configuration mode. To disable this feature, use the no form of this command.

**neighbor** { *ipv_address* | *ipv6-address* } **update-source** { *interface name* }

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| interface name | Specifies the name of the interface, as specified by the `nameif` command, that the ASA uses as the source for BGP routing. |

## Command Default

This command is disabled by default.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode. | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.18(2) | This command is added. |

## Usage Guidelines

This command is useful to run BGP protocol over the loopback interface and allow the loopback interface to participate in redistribution and prefix advertisement.

## Examples

The following example updates loopback interface loop1 as source for BGP neighbor 10.108.1.1:

```
ciscoasa(config)# router bgp 109
ciscoasa(config-router)# address-family ipv4 unicast
ciscoasa(config-router-af)# neighbor 10.108.1.1 remote-as 109
ciscoasa(config-router-af)# neighbor 10.108.1.1 update-source loop1
```

The following example updates loopback interface loop1 as source for BGP neighbor 2001::1:

```
ciscoasa(config)# router bgp 109
ciscoasa(config-router)# address-family ipv6 unicast
ciscoasa(config-router-af)# neighbor 2001::1 remote-as 109
ciscoasa(config-router-af)# neighbor 2001::1 update-source loop1
```

**Related Commands**

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables information exchange with a BGP neighbor. |
| neighbor remote-as | Adds a BGP or multi-protocol BGP routing entry to the routing table. |

# neighbor version

To configure the ASA software to accept only a particular BGP version, use the neighbor version command in the address-family configuration mode. To use the default version level of a neighbor, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **version** *number*
**no neighbor** { *ip_address* | *ipv6-address* } **version** *number*

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| number | BGP version number. The version can be set to 2 to force the software to use only Version 2 with the specified neighbor. The default is to use Version 4 and dynamically negotiate down to Version 2 if requested. |

## Command Default

BGP version 4.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

Entering this command disables dynamic version negotiation.

## Examples

The following example locks down to Version 4 of the BGP protocol:

```
ciscoasa(config)# router bgp 109
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# neighbor 172.16.27.2 version 4
ciscoasa(config-router-af)# neighbor 2001::1 version 4
```

**Related Commands**

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables exchange of information with a BGP neighbor. |

# neighbor weight

To assign a weight to a neighbor connection, use the neighbor weight command in address-family configuration mode. To remove a weight assignment, use the no form of this command.

**neighbor** { *ip_address* | *ipv6-address* } **weight number**
**no neighbor** { *ip_address* | *ipv6-address* } **weight number**

## Syntax Description

| | |
|---|---|
| *ip_address* | IP address of the neighbor router. |
| ipv6-address | IPv6 address of the neighbor router. |
| number | Weight to assign.<br>Valid values are between 0 and 65535. |

## Command Default

Routes learned through another BGP peer have a default weight of 0 and routes sourced by the local router have a default weight of 32768.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

All routes learned from this neighbor will have the assigned weight initially. The route with the highest weight will be chosen as the preferred route when multiple routes are available to a particular network.

The weights assigned with the set weight route-map command override the weights assigned using the neighbor weight command.

## Examples

The following address-family configuration mode example sets the weight of all routes learned via 172.16.12.1 to 50:

```
ciscoasa(config-router-af)# neighbor 172.16.12.1 weight 50
```

The following address-family configuration mode example sets the weight of all routes learned via 2001::1:

```
ciscoasa(config-router-af)# neighbor 2001::1 weight 50
```

## Related Commands

| Command | Description |
|---|---|
| **address-family ipv4** | Enters address-family configuration mode. |
| neighbor activate | Enables exchange of information with a BGP neighbor. |

# nem

To enable network extension mode for hardware clients, use the **nem enable** command in group-policy configuration mode. To disable NEM, use the **nem disable** command. To remove the NEM attribute from the running configuration, use the **no** form of this command. This option allows inheritance of a value from another group policy.

**nem** { **enable**|**disable** }
**no nem**

## Syntax Description

| | |
|---|---|
| **disable** | Disables Network Extension Mode. |
| **enable** | Enables Network Extension Mode. |

## Command Default

Network extension mode is disabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Group-policy configuration | • Yes | — | • Yes | — | — |

## Usage Guidelines

Network Extension mode lets hardware clients present a single, routable network to the remote private network over the VPN tunnel. IPsec encapsulates all traffic from the private network behind the hardware client to networks behind the ASA. PAT does not apply. Therefore, devices behind the ASA have direct access to devices on the private network behind the hardware client over the tunnel, and only over the tunnel, and vice versa. The hardware client must initiate the tunnel, but after the tunnel is up, either side can initiate data exchange.

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |

## Examples

The following example shows how to set NEM for the group policy named FirstGroup:

```
ciscoasa
(config)#
 group-policy FirstGroup attributes
ciscoasa
(config-group-policy)
# nem enable
```

# netmod

To disable a network module, use the **netmod** command in global configuration mode. To enable a network module, use the **no** form of this command.

**Note** This command is only supported on the Secure Firewall 3100.

**netmod 2 disable**
**no netmod 2 disable**

## Syntax Description

| | |
|---|---|
| **2** | Specifies the module in slot 2. |
| **disable** | Disabled the network module. |

## Command Default

If the module is installed when you first boot up, then it is enabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Global configuration | • Yes | • Yes | • Yes | — | • Yes |

## Command History

| Release | Modification |
|---|---|
| 9.17(1) | This command was introduced for the Secure Firewall 3100. |

## Usage Guidelines

If you install a network module before you first power on the firewall, no action is required; the network module is enabled and ready for use. If you need to make changes to your network module installation after initial bootup, then use this command.

Adding a new module or permanently removing a module requires a reload. You can hot swap a network module for a new module of the same type without having to reload. However, you must shut down the current module to remove it safely. If you replace a network module with a different type, then a reload is required. If the new module has fewer interfaces than the old module, you will have to manually remove any configuration related to interfaces that will no longer be present.

### Examples

The following example disables the network module.

```
ciscoasa(config)# netmod 2 disable
```

The following example enabled the network module.

```
ciscoasa(config)# no netmod 2 disable
```

# network (address-family)

To specify the networks to be advertised by the Border Gateway Protocol (BGP) routing processes, use the network command in address-family configuration mode. To remove an entry from the routing table, use the no form of this command.

**network** { *ipv4_address* [ **mask** *network_mask* ] | *IPv6_prefix* | *prefix_length* | *prefix_delegation_name* [ *subnet_prefix* | *prefix_length* ] } [ **route-map** *route_map_name* ]
**no network** { *ipv4_address* [ **mask** *network_mask* ] | *IPv6_prefix* | *prefix_length* | *prefix_delegation_name* [ *subnet_prefix* | *prefix_length* ] } [ **route-map** *route_map_name* ]

## Syntax Description

| | |
|---|---|
| *ipv4_address* | The IPv4 network that BGP or multiprotocol BGP will advertise. |
| *ipv6_prefix*/*prefix_length* | The IPv6 network that BGP or multiprotocol BGP will advertise. |
| mask network_mask | (Optional) Network or subnetwork mask with mask address. |
| *prefix_delegation_name* | If you enable the DHCPv6 Prefix Delegation client (**ipv6 dhcp client pd**), then you can advertise the prefix(es). |
| *subnet_prefix*/*prefix_length* | (Optional) To subnet the prefix, specify the subnet_prefix/prefix length. |
| route-map route_map_name | (Optional) Identifier of a configured route map. The route map should be examined to filter the networks to be advertised. If not specified, all networks are advertised. If the keyword is specified, but no route map tags are listed, no networks will be advertised. |

## Command Default

No networks are specified.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Address-family configuration | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.2(1) | This command was added. |
| 9.6(2) | We added the *prefix_delegation_name* [*subnet_prefix*/*prefix_length*] arguments. |

## Usage Guidelines

BGP and multiprotocol BGP networks can be learned from connected routes, from dynamic routing, and from static route sources.

The maximum number of network commands you can use is determined by the resources of the router, such as the configured NVRAM or RAM.

## Examples

The following example sets up network 10.108.0.0 to be included in the BGP updates:

```
ciscoasa(config)# router bgp 65100
ciscoasa(config-router)# address-family ipv4
ciscoasa(config-router-af)# network 10.108.0.0
```

## Related Commands

| Command | Description |
|---|---|
| **show bgp interfaces** | Displays entries in the BGP routing table. |

# network (router eigrp)

To specify a list of networks for the EIGRP routing process, use the **network** command in router configuration mode. To remove a network definition, use the **no** form of this command.

**network** *ip_addr* [ *mask* ]
**no network** *ip_addr* [ *mask* ]

## Syntax Description

| | |
|---|---|
| *ip_addr* | The IP address of a directly connected network. The interface connected to the specified network will participate in the EIGRP routing process. |
| *mask* | (Optional) The network mask for the IP address. |

## Command Default

No networks are specified.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Router configuration | • Yes | — | • Yes | — | — |

## Command History

| Release | Modification |
|---|---|
| 8.0(2) | This command was added. |

## Usage Guidelines

The **network** command starts EIGRP on all interfaces with at least one IP address in the specified network. It inserts the connected subnet from the specified network in the EIGRP topology table.

The ASA then establishes neighbors through the matched interfaces. There is no limit to the number of **network** commands that can be configured on the ASA.

## Examples

The following example defines EIGRP as the routing protocol to be used on all interfaces connected to networks 10.0.0.0 and 192.168.7.0:

```
ciscoasa(config)# router eigrp 100
ciscoasa(config-router)# network 10.0.0.0 255.0.0.0
ciscoasa(config-router)# network 192.168.7.0 255.255.255.0
```

## Related Commands

| Command | Description |
|---|---|
| **show eigrp interfaces** | Displays information about interfaces configured for EIGRP. |

| Command | Description |
| --- | --- |
| **show eigrp topology** | Displays the EIGRP topology table. |

# network (router rip)

To specify a list of networks for the RIP routing process, use the **network** command in router configuration mode. To remove a network definition, use the **no** form of this command.

**network** { *ip_addr* | *ipv6-address* } | < *prefix-length* >
**no network** { *ip_addr* | *ipv6-address* } |< *prefix-length* > [ **route-map** *route-map-name* ]

## Syntax Description

| | |
|---|---|
| *ip_addr* | The IP address of a directly connected network. The interface connected to the specified network will participate in the RIP routing process. |
| ipv6-address | The IPv6 address to be used. The IPv6 address must be entered in the format X:X:X:X::X. |
| prefix-length | The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.<br>Valid values are between 0 and 128. |
| route-map-name | The route-map whose attributes will be modified. |

## Command Default

No networks are specified.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Router configuration, Address-family configuration mode | • Yes | — | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 7.2(1) | This command was added. |
| 9.0(1) | Support for multiple context mode was added. |
| 9.3(2) | The ipv6-address argument was added and support was added for the IPv6 address-family. |

## Usage Guidelines

The network number specified must not contain any subnet information. There is no limit to the number of network commands you can use on the router. RIP routing updates will be sent and received only through interfaces on the specified networks. Also, if the network of an interface is not specified, the interface will not be advertised in any RIP update.

## Examples

The following example defines RIP as the routing protocol to be used on all interfaces connected to networks 10.0.0.0 and 192.168.7.0:

```
ciscoasa(config)# router rip
ciscoasa(config-router)# network 10.0.0.0
ciscoasa(config-router)# network 192.168.7.0
In the following example the attributes of the test-route-map route map connected to the
2001::1 network will be modified.
ciscoasa(config-router)# network 2001:0:0:0::1 route-map test-route-map
```

## Related Commands

| Command | Description |
|---|---|
| **router rip** | Enters router configuration mode. |
| **show running-config router** | Displays the commands in the global router configuration. |

# network-acl

To specify a firewall ACL name that you configured previously using the **access-list** command, use the **network-acl** command in dynamic-access-policy-record configuration mode. To remove an existing network ACL, use the **no** form of this command. To remove all network ACLs, use the command without arguments.

**network-acl** *name*
**no network-acl** [ *name* ]

## Syntax Description

| | |
|---|---|
| *name* | Specifies the name of the network ACL. The maximum number for a name is 240 characters. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Dynamic-access-policy-record configuration | • Yes | • Yes | • Yes | — | — |

## Command History

| Release | Modification |
|---|---|
| 8.0(2) | This command was added. |

## Usage Guidelines

Use this command multiple time to assign multiple firewall ACLs to the DAP record.

The ASA verifies each of the ACLs you specify to make sure they contain only permit rules or only deny rules for the access-list entries. If any of the specified ACLs contain mixed permit and deny rules, then the ASA rejects the command.

The following example shows how to apply a network ACL called Finance Restrictions to the DAP record named Finance.

```
ciscoasa
(config)#

dynamic-access-policy-record Finance
ciscoasa
(config-dynamic-access-policy-record)#
 network-acl Finance Restrictions
ciscoasa
(config-dynamic-access-policy-record)#
```

## Related Commands

| Command | Description |
|---|---|
| access-policy | Configures a firewall access policy. |

| Command | Description |
| --- | --- |
| dynamic-access-policy-record | Creates a DAP record. |
| **show running-config dynamic-access-policy-record** [*name* ] | Displays the running configuration for all DAP records, or for the named DAP record. |

# network area

To define the interfaces on which OSPF runs and to define the area ID for those interfaces, use the **network area** command in router configuration mode. To disable OSPF routing for interfaces defined with the address/netmask pair, use the **no** form of this command.

**network** *addr mask* **area** *area_id*
**no network** *addr mask* **area** *area_id*

## Syntax Description

| | |
|---|---|
| *addr* | IP address. |
| **area** *area_id* | Specifies the area that is to be associated with the OSPF address range. The *area_id* can be specified in either IP address format or in decimal format. When specified in decimal format, valid values range from 0 to 4294967295. |
| *mask* | The network mask. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Router configuration | • Yes | • — | • Yes | • — | — |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |

## Usage Guidelines

For OSPF to operate on the interface, the address of the interface must be covered by the **network area** command. If the **network area** command does not cover the IP address of the interface, it will not enable OSPF over that interface.

There is no limit to the number of **network area** commands you can use on the ASA.

## Examples

The following example enables OSPF on the 192.168.1.1 interface and assigns it to area 2:

```
ciscoasa(config-router)# network 192.168.1.1 255.255.255.0 area 2
```

## Related Commands

| Command | Description |
|---|---|
| **router ospf** | Enters router configuration mode. |

| Command | Description |
| --- | --- |
| **show running-config router** | Displays the commands in the global router configuration. |

# network-object

To add a host object, a network object, or a subnet object to a network object group, use the network-object command in object-group network configuration mode. To remove network objects, use the **no** form of this command.

**network-object** { **host** *address* | *IPv4_address mask* | *IPv6_address* | *IPv6_prefix* | **object** *name* }
**no network-object** { **host** *ip_address* | *ip_address mask* | **object** *name* }

## Syntax Description

| | |
|---|---|
| **host** *ip_address* | Specifies a host IPv4 or IPv6 address. |
| *IPv4_address mask* | Specifies an IPv4 network address and subnet mask. |
| *IPv6_address/IPv6_prefix* | Specifies an IPv6 network address and prefix length. |
| **object** *name* | Specifies a network object (created by the **object network** command). |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Object-group network configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 8.3(1) | The **object** argument was added to support network objects (**object network** command). |
| 9.0(1) | Previously, network object groups could only contain all IPv4 addresses or all IPv6 addresses. Now network object groups can support a mix of both IPv4 and IPv6 addresses, although you cannot use a mixed group in NAT. |

## Usage Guidelines

The **network-object** command is used with the **object-group** command to define a host object, a network object, or a subnet object.

## Examples

The following example shows how to use the **network-object** command to create a new host object in a network object group:

```
ciscoasa(config)# object-group network sjj_eng_ftp_servers
ciscoasa(config-network-object-group)# network-object host sjj.eng.ftp
ciscoasa(config-network-object-group)# network-object host 172.16.56.195
ciscoasa(config-network-object-group)# network-object 192.168.1.0 255.255.255.224
```

```
ciscoasa(config-network-object-group)# group-object sjc_eng_ftp_servers
ciscoasa(config)#
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure object-group** | Removes all the **object-group** commands from the configuration. |
| **group-object** | Adds network object groups. |
| object network | Adds a network object. |
| **object-group network** | Defines network object groups. |
| **show running-config object-group** | Displays the current object groups. |

# network-service-member

To add a network-service object to a network-service group, use the **network-service-member** command in object group configuration mode. Use to **no** form of the command to remove an object from a group

**network-service-member** *object_name*
**no network-service-member** *object_name*

## Syntax Description

| | |
|---|---|
| *object_name* | The name of a network-service object. If there are spaces in the name, enclose the name in double quotation marks. |

## Command Default

No default values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Network-service object-group configuration mode | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.17(1) | This command was added. |

### Example

The following example adds three existing network-service objects to a network-service object group.

```
object-group network-service SaaS_Applications
    description This group includes relevant 'Software as a Service' applications
    network-service-member "outlook 365"
    network-service-member webex
    network-service-member box
```

## Related Commands

| Command | Description |
|---|---|
| **clear object-group** | Clears hit counts for object groups. |
| **object-group network-service** | Defines network-service object groups. |
| **show object-group network-service** | Displays network-service objects and their hit counts. |

# nis address

To provide the Network Information Service (NIS) address to StateLess Address Auto Configuration (SLAAC) clients when you configure the DHCPv6 server, use the **nis address** command in ipv6 dhcp pool configuration mode. To remove the NIS server, use the **no** form of this command.

**nis address** *nis_ipv6_address*
**no nis address** *nis_ipv6_address*

## Syntax Description

| | |
|---|---|
| *nis_ipv6_address* | Specifies the NIS IPv6 address. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Ipv6 dhcp pool configuration | • Yes | — | • Yes | — | — |

## Command History

| Release | Modification |
|---|---|
| 9.6(2) | We introduced this command. |

## Usage Guidelines

For clients that use SLAAC in conjunction with the Prefix Delegation feature, you can configure the ASA to provide information in an **ipv6 dhcp pool**, including the NIS address, when they send Information Request (IR) packets to the ASA. The ASA only accepts IR packets, and does not assign addresses to the clients. Configure the DHCPv6 stateless server using the **ipv6 dhcp server** command; you specify an **ipv6 dhcp pool** name when you enable the server.

Configure Prefix Delegation using the **ipv6 dhcp client pd** command.

This feature is not supported in clustering.

## Examples

The following example creates two IPv6 DHCP pools, and enables the DHCPv6 server on two interfaces:

```
ipv6 dhcp pool Eng-Pool
domain-name eng.example.com
dns-server 2001:DB8:1::1
nis domain-name eng.example.com
nis address 2001:DB8:1::2
ipv6 dhcp pool IT-Pool
domain-name it.example.com
dns-server 2001:DB8:1::1
nis domain-name it.example.com
nis address 2001:DB8:1::2
```

```
interface gigabitethernet 0/0
ipv6 address dhcp setroute default
ipv6 dhcp client pd Outside-Prefix
interface gigabitethernet 0/1
ipv6 address Outside-Prefix ::1:0:0:0:1/64
ipv6 dhcp server Eng-Pool
ipv6 nd other-config-flag
interface gigabitethernet 0/2
ipv6 address Outside-Prefix ::2:0:0:0:1/64
ipv6 dhcp server IT-Pool
ipv6 nd other-config-flag
```

## Related Commands

| Command | Description |
|---|---|
| **clear ipv6 dhcp statistics** | Clears DHCPv6 statistics. |
| **domain-name** | Configures the domain name provided to SLAAC clients in responses to IR messages. |
| **dns-server** | Configures the DNS server provided to SLAAC clients in responses to IR messages. |
| **import** | Uses one or more parameters that the ASA obtained from the DHCPv6 server on the Prefix Delegation client interface, and provides them to SLAAC clients in responses to IR messages. |
| **ipv6 address** | Enables IPv6 and configures the IPv6 addresses on an interface. |
| **ipv6 address dhcp** | Obtains an address using DHCPv6 for an interface. |
| **ipv6 dhcp client pd** | Uses a delegated prefix to set the address for an interface. |
| ipv6 dhcp client pd hint | Provides one or more hints about the delegated prefix you want to receive. |
| **ipv6 dhcp pool** | Creates a pool that includes information that you want to provide to SLAAC clients on a given interface using the DHCPv6 stateless server. |
| **ipv6 dhcp server** | Enables the DHCPv6 stateless server. |
| **network** | Configures BGP to advertise the delegated prefix received from the server. |
| **nis address** | Configures the NIS address provided to SLAAC clients in responses to IR messages. |
| **nis domain-name** | Configures the NIS domain name provided to SLAAC clients in responses to IR messages. |
| **nisp address** | Configures the NISP address provided to SLAAC clients in responses to IR messages. |
| **nisp domain-name** | Configures the NISP domain name provided to SLAAC clients in responses to IR messages. |
| **show bgp ipv6 unicast** | Displays entries in the IPv6 BGP routing table. |

| Command | Description |
|---|---|
| **show ipv6 dhcp** | Shows DHCPv6 information. |
| **show ipv6 general-prefix** | Shows all the prefixes acquired by the DHCPv6 Prefix Delegation clients and the ASA distribution of that prefix to other processes. |
| **sip address** | Configures the SIP address provided to SLAAC clients in responses to IR messages. |
| **sip domain-name** | Configures the SIP domain name provided to SLAAC clients in responses to IR messages. |
| **sntp address** | Configures the SNTP address provided to SLAAC clients in responses to IR messages. |

# nis domain-name

To provide the Network Information Service (NIS) domain name to StateLess Address Auto Configuration (SLAAC) clients when you configure the DHCPv6 server, use the **nis domain-name** command in ipv6 dhcp pool configuration mode. To remove the NIS domain name, use the **no** form of this command.

**nis domain-name** *nis_domain_name*
**no nis domain-name** *nis_domain_name*

## Syntax Description

| | |
|---|---|
| *nis_domain_name* | Specifies the NIS domain name. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Ipv6 dhcp pool configuration | • Yes | — | • Yes | — | — |

## Command History

| Release | Modification |
|---|---|
| 9.6(2) | We introduced this command. |

## Usage Guidelines

For clients that use SLAAC in conjunction with the Prefix Delegation feature, you can configure the ASA to provide information in an **ipv6 dhcp pool**, including the NIS domain name, when they send Information Request (IR) packets to the ASA. The ASA only accepts IR packets, and does not assign addresses to the clients. Configure the DHCPv6 stateless server using the **ipv6 dhcp server** command; you specify an **ipv6 dhcp pool** name when you enable the server.

Configure Prefix Delegation using the **ipv6 dhcp client pd** command.

This feature is not supported in clustering.

## Examples

The following example creates two IPv6 DHCP pools, and enables the DHCPv6 server on two interfaces:

```
ipv6 dhcp pool Eng-Pool
domain-name eng.example.com
dns-server 2001:DB8:1::1
nis domain-name eng.example.com
nis address 2001:DB8:1::2
ipv6 dhcp pool IT-Pool
domain-name it.example.com
dns-server 2001:DB8:1::1
nis domain-name it.example.com
nis address 2001:DB8:1::2
```

```
interface gigabitethernet 0/0
ipv6 address dhcp setroute default
ipv6 dhcp client pd Outside-Prefix
interface gigabitethernet 0/1
ipv6 address Outside-Prefix ::1:0:0:0:1/64
ipv6 dhcp server Eng-Pool
ipv6 nd other-config-flag
interface gigabitethernet 0/2
ipv6 address Outside-Prefix ::2:0:0:0:1/64
ipv6 dhcp server IT-Pool
ipv6 nd other-config-flag
```

## Related Commands

| Command | Description |
|---|---|
| **clear ipv6 dhcp statistics** | Clears DHCPv6 statistics. |
| **domain-name** | Configures the domain name provided to SLAAC clients in responses to IR messages. |
| **dns-server** | Configures the DNS server provided to SLAAC clients in responses to IR messages. |
| **import** | Uses one or more parameters that the ASA obtained from the DHCPv6 server on the Prefix Delegation client interface, and provides them to SLAAC clients in responses to IR messages. |
| **ipv6 address** | Enables IPv6 and configures the IPv6 addresses on an interface. |
| **ipv6 address dhcp** | Obtains an address using DHCPv6 for an interface. |
| **ipv6 dhcp client pd** | Uses a delegated prefix to set the address for an interface. |
| ipv6 dhcp client pd hint | Provides one or more hints about the delegated prefix you want to receive. |
| **ipv6 dhcp pool** | Creates a pool that includes information that you want to provide to SLAAC clients on a given interface using the DHCPv6 stateless server. |
| **ipv6 dhcp server** | Enables the DHCPv6 stateless server. |
| **network** | Configures BGP to advertise the delegated prefix received from the server. |
| **nis address** | Configures the NIS address provided to SLAAC clients in responses to IR messages. |
| **nis domain-name** | Configures the NIS domain name provided to SLAAC clients in responses to IR messages. |
| **nisp address** | Configures the NISP address provided to SLAAC clients in responses to IR messages. |
| **nisp domain-name** | Configures the NISP domain name provided to SLAAC clients in responses to IR messages. |
| **show bgp ipv6 unicast** | Displays entries in the IPv6 BGP routing table. |

| Command | Description |
|---|---|
| **show ipv6 dhcp** | Shows DHCPv6 information. |
| **show ipv6 general-prefix** | Shows all the prefixes acquired by the DHCPv6 Prefix Delegation clients and the ASA distribution of that prefix to other processes. |
| **sip address** | Configures the SIP address provided to SLAAC clients in responses to IR messages. |
| **sip domain-name** | Configures the SIP domain name provided to SLAAC clients in responses to IR messages. |
| **sntp address** | Configures the SNTP address provided to SLAAC clients in responses to IR messages. |

# nisp address

To provide the Network Information Service Plus (NIS+) server IP address to StateLess Address Auto Configuration (SLAAC) clients when you configure the DHCPv6 server, use the **nisp address** command in ipv6 dhcp pool configuration mode. To remove the NIS+ server, use the **no** form of this command.

**nisp address** *nisp_ipv6_address*
**no nisp address** *nisp_ipv6_address*

## Syntax Description

| | |
|---|---|
| *nisp_ipv6_address* | Specifies the NIS+ server IPv6 address. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Ipv6 dhcp pool configuration | • Yes | — | • Yes | — | — |

## Command History

| Release | Modification |
|---|---|
| 9.6(2) | We introduced this command. |

## Usage Guidelines

For clients that use SLAAC in conjunction with the Prefix Delegation feature, you can configure the ASA to provide information in an **ipv6 dhcp pool**, including the NIS+ server, when they send Information Request (IR) packets to the ASA. The ASA only accepts IR packets, and does not assign addresses to the clients. Configure the DHCPv6 stateless server using the **ipv6 dhcp server** command; you specify an **ipv6 dhcp pool** name when you enable the server.

Configure Prefix Delegation using the **ipv6 dhcp client pd** command.

This feature is not supported in clustering.

## Examples

The following example creates two IPv6 DHCP pools, and enables the DHCPv6 server on two interfaces:

```
ipv6 dhcp pool Eng-Pool
domain-name eng.example.com
dns-server 2001:DB8:1::1
nisp domain-name eng.example.com
nisp address 2001:DB8:1::2
ipv6 dhcp pool IT-Pool
domain-name it.example.com
dns-server 2001:DB8:1::1
nisp domain-name it.example.com
nisp address 2001:DB8:1::2
```

```
interface gigabitethernet 0/0
ipv6 address dhcp setroute default
ipv6 dhcp client pd Outside-Prefix
interface gigabitethernet 0/1
ipv6 address Outside-Prefix ::1:0:0:0:1/64
ipv6 dhcp server Eng-Pool
ipv6 nd other-config-flag
interface gigabitethernet 0/2
ipv6 address Outside-Prefix ::2:0:0:0:1/64
ipv6 dhcp server IT-Pool
ipv6 nd other-config-flag
```

## Related Commands

| Command | Description |
|---|---|
| **clear ipv6 dhcp statistics** | Clears DHCPv6 statistics. |
| **domain-name** | Configures the domain name provided to SLAAC clients in responses to IR messages. |
| **dns-server** | Configures the DNS server provided to SLAAC clients in responses to IR messages. |
| **import** | Uses one or more parameters that the ASA obtained from the DHCPv6 server on the Prefix Delegation client interface, and provides them to SLAAC clients in responses to IR messages. |
| **ipv6 address** | Enables IPv6 and configures the IPv6 addresses on an interface. |
| **ipv6 address dhcp** | Obtains an address using DHCPv6 for an interface. |
| **ipv6 dhcp client pd** | Uses a delegated prefix to set the address for an interface. |
| ipv6 dhcp client pd hint | Provides one or more hints about the delegated prefix you want to receive. |
| **ipv6 dhcp pool** | Creates a pool that includes information that you want to provide to SLAAC clients on a given interface using the DHCPv6 stateless server. |
| **ipv6 dhcp server** | Enables the DHCPv6 stateless server. |
| **network** | Configures BGP to advertise the delegated prefix received from the server. |
| **nis address** | Configures the NIS address provided to SLAAC clients in responses to IR messages. |
| **nis domain-name** | Configures the NIS domain name provided to SLAAC clients in responses to IR messages. |
| **nisp address** | Configures the NISP address provided to SLAAC clients in responses to IR messages. |
| **nisp domain-name** | Configures the NISP domain name provided to SLAAC clients in responses to IR messages. |
| **show bgp ipv6 unicast** | Displays entries in the IPv6 BGP routing table. |

| Command | Description |
|---|---|
| **show ipv6 dhcp** | Shows DHCPv6 information. |
| **show ipv6 general-prefix** | Shows all the prefixes acquired by the DHCPv6 Prefix Delegation clients and the ASA distribution of that prefix to other processes. |
| **sip address** | Configures the SIP address provided to SLAAC clients in responses to IR messages. |
| **sip domain-name** | Configures the SIP domain name provided to SLAAC clients in responses to IR messages. |
| **sntp address** | Configures the SNTP address provided to SLAAC clients in responses to IR messages. |

# nisp domain-name

To provide the Network Information Service Plus (NIS+) domain name to StateLess Address Auto Configuration (SLAAC) clients when you configure the DHCPv6 server, use the **nisp domain-name** command in ipv6 dhcp pool configuration mode. To remove the NIS+ somain name, use the **no** form of this command.

**nisp domain-name** *nisp_domain_name*
**no nisp domain-name** *nisp_domain_name*

## Syntax Description

| | |
|---|---|
| *nisp_domain_name* | Specifies the NIS+ domain name. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Ipv6 dhcp pool configuration | • Yes | — | • Yes | — | — |

## Command History

| Release | Modification |
|---|---|
| 9.6(2) | We introduced this command. |

## Usage Guidelines

For clients that use SLAAC in conjunction with the Prefix Delegation feature, you can configure the ASA to provide information in an **ipv6 dhcp pool**, including the NIS+ domain name, when they send Information Request (IR) packets to the ASA. The ASA only accepts IR packets, and does not assign addresses to the clients. Configure the DHCPv6 stateless server using the **ipv6 dhcp server** command; you specify an **ipv6 dhcp pool** name when you enable the server.

Configure Prefix Delegation using the **ipv6 dhcp client pd** command.

This feature is not supported in clustering.

## Examples

The following example creates two IPv6 DHCP pools, and enables the DHCPv6 server on two interfaces:

```
ipv6 dhcp pool Eng-Pool
domain-name eng.example.com
dns-server 2001:DB8:1::1
nisp domain-name eng.example.com
nisp address 2001:DB8:1::2
ipv6 dhcp pool IT-Pool
domain-name it.example.com
dns-server 2001:DB8:1::1
nisp domain-name it.example.com
nisp address 2001:DB8:1::2
```

```
interface gigabitethernet 0/0
ipv6 address dhcp setroute default
ipv6 dhcp client pd Outside-Prefix
interface gigabitethernet 0/1
ipv6 address Outside-Prefix ::1:0:0:0:1/64
ipv6 dhcp server Eng-Pool
ipv6 nd other-config-flag
interface gigabitethernet 0/2
ipv6 address Outside-Prefix ::2:0:0:0:1/64
ipv6 dhcp server IT-Pool
ipv6 nd other-config-flag
```

## Related Commands

| Command | Description |
|---|---|
| **clear ipv6 dhcp statistics** | Clears DHCPv6 statistics. |
| **domain-name** | Configures the domain name provided to SLAAC clients in responses to IR messages. |
| **dns-server** | Configures the DNS server provided to SLAAC clients in responses to IR messages. |
| **import** | Uses one or more parameters that the ASA obtained from the DHCPv6 server on the Prefix Delegation client interface, and provides them to SLAAC clients in responses to IR messages. |
| **ipv6 address** | Enables IPv6 and configures the IPv6 addresses on an interface. |
| **ipv6 address dhcp** | Obtains an address using DHCPv6 for an interface. |
| **ipv6 dhcp client pd** | Uses a delegated prefix to set the address for an interface. |
| ipv6 dhcp client pd hint | Provides one or more hints about the delegated prefix you want to receive. |
| **ipv6 dhcp pool** | Creates a pool that includes information that you want to provide to SLAAC clients on a given interface using the DHCPv6 stateless server. |
| **ipv6 dhcp server** | Enables the DHCPv6 stateless server. |
| **network** | Configures BGP to advertise the delegated prefix received from the server. |
| **nis address** | Configures the NIS address provided to SLAAC clients in responses to IR messages. |
| **nis domain-name** | Configures the NIS domain name provided to SLAAC clients in responses to IR messages. |
| **nisp address** | Configures the NISP address provided to SLAAC clients in responses to IR messages. |
| **nisp domain-name** | Configures the NISP domain name provided to SLAAC clients in responses to IR messages. |
| **show bgp ipv6 unicast** | Displays entries in the IPv6 BGP routing table. |

| Command | Description |
|---|---|
| **show ipv6 dhcp** | Shows DHCPv6 information. |
| **show ipv6 general-prefix** | Shows all the prefixes acquired by the DHCPv6 Prefix Delegation clients and the ASA distribution of that prefix to other processes. |
| **sip address** | Configures the SIP address provided to SLAAC clients in responses to IR messages. |
| **sip domain-name** | Configures the SIP domain name provided to SLAAC clients in responses to IR messages. |
| **sntp address** | Configures the SNTP address provided to SLAAC clients in responses to IR messages. |

# nop

To define an action when the No Operation IP option occurs in a packet header with IP Options inspection, use the **nop** command in parameters configuration mode. To disable this feature, use the **no** form of this command.

**nop action** { **allow | clear** }
**no nop action** { **allow | clear** }

## Syntax Description

| | |
|---|---|
| *allow* | Allow packets containing the No Operation IP option. |
| *clear* | Remove the No Operation option from packet headers and then allow the packets. |

## Command Default

By default, IP Options inspection drops packets containing the No Operation IP option.

You can change the default using the **default** command in the IP Options inspection policy map.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Parameters configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 8.2(2) | This command was added. |

## Usage Guidelines

This command can be configured in an IP Options inspection policy map.

You can configure IP Options inspection to control which IP packets with specific IP options are allowed through the ASA. You can allow a packet to pass without change or clear the specified IP options and then allow the packet to pass.

The Options field in the IP header can contain zero, one, or more options, which makes the total length of the field variable. However, the IP header must be a multiple of 32 bits. If the number of bits of all options is not a multiple of 32 bits, the No Operation (NOP) or IP Option 1 is used as “internal padding” to align the options on a 32-bit boundary.

## Examples

The following example shows how to set up an action for IP Options inspection in a policy map:

```
ciscoasa(config)# policy-map type inspect ip-options ip-options_map
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# eool action allow
ciscoasa(config-pmap-p)# nop action allow
ciscoasa(config-pmap-p)# router-alert action allow
```

**Related Commands**

| Command | Description |
|---|---|
| **class** | Identifies a class map name in the policy map. |
| **class-map type inspect** | Creates an inspection class map to match traffic specific to an application. |
| **policy-map** | Creates a Layer 3/4 policy map. |
| **show running-config policy-map** | Display all current policy map configurations. |

# nsf cisco

To enable Cisco nonstop forwarding (NSF) operations on an ASA that is running Open Shortest Path First (OSPF), use the nsf cisco command in router configuration mode. To return to the default, use the no form of this command.

**nsf cisco** [ **enforce global** ]
**no nsf cisco** [ **enforce global** ]

## Syntax Description

| | |
|---|---|
| *enforce global* | (Optional) Cancels NSF restart on all interfaces when neighboring networking devices that are not NSF-aware are detected on any interface during the restart process. |

## Command Default

Cisco NSF graceful restart is disabled by default.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Router configuration mode | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.3(1) | This command was added. |

## Usage Guidelines

This command enables Cisco NSF on an OSPF router. When NSF is enabled on a router, the router is NSF-capable and will operate in restarting mode.

If a router is expected to cooperate with a neighbor that is doing an NSF graceful restart only, the neighbor router must be running a Cisco software release that supports NSF but NSF need not be configured on the router. When a router is running a Cisco software release that supports NSF, the router is NSF-aware.

By default, neighboring NSF-aware routers will operate in NSF helper mode during a graceful restart.

If neighbors that are not NSF-aware are detected on a network interface during an NSF graceful restart, restart is aborted on that interface only and graceful restart will continue on other interfaces. To cancel restart for the entire OSPF process when neighbors that are not NSF-aware are detected during restart, configure this command with the enforce global keywords.

**Note** The NSF graceful restart will also be canceled for the entire process when a neighbor adjacency reset is detected on any interface or when an OSPF interface goes down.

## Examples

The following example enables Cisco NSF graceful restart with the enforce global option:

```
ciscoasa
(config)# router ospf 24
ciscoasa
(config-router)# cisco nsf enforce global
```

## Related Commands

| Command | Description |
|---|---|
| nsf cisco helper | Enables Cisco NSF helper mode on ASA. |
| **nsf ietf** | Enables IETF NSF |

# nsf cisco helper

To enable Cisco nonstop forwarding (NSF) helper mode on an ASA that is running Open Shortest Path First (OSPF), use the nsf cisco helper command in the router configuration mode. The Cisco NSF helper mode is enabled by default and can be disabled by issuing the no nsf cisco helper under router configuration mode.

**nsf cisco helper**
**no nsf cisco helper**

## Syntax Description

This command has no arguments or keywords.

## Command Default

The Cisco NSF helper mode is enabled by default.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Router configuration mode | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.3(1) | This command was added. |

## Usage Guidelines

When an ASA has NSF enabled, the ASA is said to be NSF-capable and will operate in graceful restart mode--the OSPF router process performs nonstop forwarding recovery due to a Route Processor (RP) switchover. By default, the neighboring ASAs of the NSF-capable ASA will be NSF-aware and will operate in NSF helper mode. When the NSF-capable ASA is performing graceful restart, the helper ASAs assist in the nonstop forwarding recovery process. If you do not want the ASA to help the restarting neighbor with nonstop forwarding recovery, enter the no nsf cisco helper command.

## Examples

The following example disables the NSF helper mode:

```
ciscoasa
(config)# router ospf 24
ciscoasa
(config-router)# no nsf cisco helper
```

## Related Commands

| Command | Description |
|---|---|
| nsf cisco | Enables Cisco NSF on ASA. |
| **nsf ietf** | Enables IETF NSF |

# nsf ietf

To configure Internet Engineering Task Force (IETF) NSF operations on an ASA that is running OSPF, use the nsf ietf command in router configuration mode. To return to the default, use the no form of this command.

**nsf ietf** [ **restart-interval** *seconds* ]
**no nsf ietf**

## Syntax Description

| | |
|---|---|
| *restart-interval seconds* | (Optional) Specifies the length of the graceful restart interval, in seconds. The range is from 1 to 1800. The default is 120. |

**Note** For a restart interval below 30 seconds, graceful restart will be terminated.

## Command Default

IETF NSF graceful restart mode is disabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Router configuration mode | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.3(1) | This command was added. |

## Usage Guidelines

This command enables IETF NSF on an ASA. When NSF is enabled on an ASA, the ASA is NSF-capable and will operate in restarting mode.

If an ASA is expected to cooperate with a neighbor that is doing an NSF graceful restart only, the neighbor ASA must support NSF but NSF need not be configured on the router. When an ASA is running an application that supports NSF, the ASA is NSF-aware.

## Examples

The following example disables the NSF helper mode:

```
ciscoasa
(config)# router ospf 24
ciscoasa
(config-router)# nsf ietf restart-interval 240
```

## Related Commands

| Command | Description |
|---|---|
| nsf cisco | Enables Cisco NSF on ASA. |

| Command | Description |
|---|---|
| **nsf cisco helper** | Enables Cisco NSF helper mode on ASA. |
| nsf ietf helper | Enables IETF NSF helper mode on ASA. |

# nsf ietf helper

The IETF NSF helper mode is enabled by default. To enable IETF NSF helper mode explicitly, use the nsf ietf helper command in router configuration mode. It can be disabled by using the no form of the command.

Optionally, strict link-state advertisement (LSA) checking can be enabled by using the nsf ietf helper strict-lsa-checking command.

**nsf ietf helper** [ **strict-lsa-checking** ]
**no nsf ietf helper**

## Syntax Description

| | |
|---|---|
| *strict-lsa-checking* | (Optional) Enables strict link-state advertisement (LSA) checking for helper mode. |

## Command Default

The IETF NSF helper mode is enabled by default.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Router configuration mode | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.3(1) | This command was added. |

## Usage Guidelines

When an ASA has NSF enabled, it is said to be NSF-capable and will operate in graceful restart mode--the OSPF process performs nonstop forwarding recovery due to a Route Processor (RP) switchover. By default, the neighboring ASAs of the NSF-capable ASA will be NSF-aware and will operate in NSF helper mode. When the NSF-capable ASA is performing graceful restart, the helper ASAs assist in the nonstop forwarding recovery process. If you do not want the ASA to help the restarting neighbor with nonstop forwarding recovery, enter the no nsf ietf helper command.

To enable strict LSA checking on both NSF-aware and NSF-capable ASAs, enter the nsf ietf helper strict-lsa-checking command. However, strict LSA checking will not become effective until the ASA becomes a helper ASA during an IETF graceful restart process. With strict LSA checking enabled, the helper ASA will terminate the helping process of the restarting ASA if it detects that there is a change to an LSA that would be flooded to the restarting ASA or if there is a changed LSA on the retransmission list of the restarting ASA when the graceful restart process is initiated.

## Examples

The following example enables IETF NSF helper with strict LSA checking:

```
ciscoasa
(config)# router ospf 24
```

```
ciscoasa
(config-router)# nsf ietf helper strict-lsa-checking
```

## Related Commands

| Command | Description |
|---|---|
| nsf cisco | Enables Cisco NSF on ASA. |
| **nsf cisco helper** | Enables Cisco NSF helper mode on ASA. |
| nsf ietf | Enables IETF NSF on ASA. |

# nt-auth-domain-controller

To specify the name of the NT Primary Domain Controller for this server, use the **nt-auth-domain-controller** command in aaa-server host configuration mode. To remove this specification, use the **no** form of this command.

**nt-auth-domain-controller** *string*
**no nt-auth-domain-controller**

## Syntax Description

*string* Specifies the name, up to 16 characters long, of the Primary Domain Controller for this server.

## Command Default

No default behaviors or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Aaa-server host configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |

## Usage Guidelines

This command is valid only for NT Authentication AAA servers. You must have first used the **aaa-server host** command to enter host configuration mode. The name in the *string* variable must match the NT entry on the server itself.

## Examples

The following example configures the name of the NT Primary Domain Controller for this server as "primary1":

```
ciscoasa
(config)# aaa-server svrgrp1 protocol nt
ciscoasa
(configaaa-sesrver-group)# aaa-server svrgrp1 host 1.2.3.4
ciscoasa
(config-aaa-server-host)# nt-auth-domain-controller primary1
ciscoasa
(config-aaa-server-host)#
```

## Related Commands

| Command | Description |
|---|---|
| **aaa server host** | Enters aaa server host configuration mode so that you can configure AAA server parameters that are host-specific. |
| **clear configure aaa-server** | Remove all AAA command statements from the configuration. |

| Command | Description |
| --- | --- |
| **show running-config aaa-server** | Displays AAA server statistics for all AAA servers, for a particular server group, for a particular server within a particular group, or for a particular protocol. |

# ntp authenticate

To enable authentication with an NTP server, use the **ntp authenticate** command in global configuration mode. To disable NTP authentication, use the **no** form of this command.

**ntp authenticate**
**no ntp authenticate**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Global Configuration | • Yes | • Yes | • Yes | • — | Yes |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |

## Usage Guidelines

If you enable authentication, the ASA only communicates with an NTP server if it uses the correct trusted key in the packets (see the **ntp trusted-key** command). You must also specify the server key (see the **ntp server key** command), or the ASA will communicate to the server without authentication even when you configure the **ntp authenticate** command. The ASA also uses an authentication key to synchronize with the NTP server (see the **ntp authentication-key** command).

## Examples

The following example identifies two NTP servers and enables authentication for the key IDs 1 and 2:

```
ciscoasa(config)# ntp server 10.1.1.1 key 1 prefer
ciscoasa(config)# ntp server 10.2.1.1 key 2
ciscoasa(config)# ntp authenticate
ciscoasa(config)# ntp trusted-key 1
ciscoasa(config)# ntp trusted-key 2
ciscoasa(config)# ntp authentication-key 1 md5
aNiceKey
ciscoasa(config)# ntp authentication-key 2 md5
aNiceKey2
```

**Related Commands**

| Command | Description |
|---|---|
| **ntp authentication-key** | Sets an encrypted authentication key to synchronize with an NTP server. |
| **ntp server** | Identifies an NTP server. |
| **ntp trusted-key** | Provides a key ID for the ASA to use in packets for authentication with an NTP server. |
| **show ntp associations** | Shows the NTP servers with which the ASA is associated. |
| **show ntp status** | Shows the status of the NTP association. |

# ntp authentication-key

To set a key to authenticate with an NTP server, use the **ntp authentication-key** command in global configuration mode. To remove the key, use the **no** form of this command.

**ntp authentication-key** *key_id* { **md5 | sha1 | sha256 | sha512 | cmac** } *key*
**no ntp authentication-key** *key_id* [ { **md5 | sha1 | sha256 | sha512 | cmac** } [ **0|8** ] *key* ]

## Syntax Description

| | |
|---|---|
| 0 | (optional) Indicates <key_value> is plain text. Format is plain text if 0 or 8 is not present. |
| 8 | (optional) Indicates <key_value> is encrypted text. Format is plain text if 0 or 8 is not present. |
| *key* | Sets the key value as a string up to 32 characters in length. |
| *key_id* | Identifies a key ID between 1 and 4294967295. You must specify this ID as a trusted key using the **ntp trusted-key** command. |
| **md5** | Specifies the authentication algorithm as MD5. |
| **sha1** | Specifies the authentication algorithm as SHA-1. |
| **sha256** | Specifies the authentication algorithm as SHA-256. |
| **sha512** | Specifies the authentication algorithm as SHA-512. |
| **cmac** | Specifies the authentication algorithm as AES-CMAC. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Global Configuration | • Yes | • Yes | • Yes | • — | Yes |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |
| 9.13(1) | The **sha1**, **sha256**, **sha512**, and **cmac** keywords were added. |

## Usage Guidelines

To use NTP authentication, also configure the **ntp authenticate** command and **ntp server key** command.

## Examples

The following example identifies two NTP servers and enables authentication for the key IDs 1 and 2:

```
ciscoasa(config)# ntp server 10.1.1.1 key 1 prefer
ciscoasa(config)# ntp server 10.2.1.1 key 2
ciscoasa(config)# ntp authenticate
ciscoasa(config)# ntp trusted-key 1
ciscoasa(config)# ntp trusted-key 2
ciscoasa(config)# ntp authentication-key 1 md5
aNiceKey
ciscoasa(config)# ntp authentication-key 2 md5
aNiceKey2
```

## Related Commands

| Command | Description |
|---|---|
| **ntp authenticate** | Enables NTP authentication. |
| **ntp server** | Identifies an NTP server. |
| **ntp trusted-key** | Provides a key ID for the ASA to use in packets for authentication with an NTP server. |
| **show ntp associations** | Shows the NTP servers with which the ASA is associated. |
| **show ntp status** | Shows the status of the NTP association. |

# ntp server

To identify an NTP server to set the time on the ASA, use the **ntp server** command in global configuration mode. To remove the server, use the **no** form of this command.

**ntp server** *ip_address* [ **key** *key_id* ] [ **source** *interface_name* ] [ **prefer** ]
**no ntp server** *ip_address* [ **key** *key_id* ] [ **source** *interface_name* ] [ **prefer** ]

## Syntax Description

| | |
|---|---|
| *ip_address* | Sets the IPv4 or IPv6 IP address of the NTP server. |
| **key** *key_id* | If you enable authentication using the **ntp authenticate** command, sets the trusted key ID for this server. See also the **ntp trusted-key** command. |
| **source** *interface_name* | Identifies the outgoing interface for NTP packets if you do not want to use the default interface in the routing table. Because the system does not include any interfaces in multiple context mode, specify an interface name defined in the admin context. |
| **prefer** | Sets this NTP server as the preferred server if multiple servers have similar accuracy. NTP uses an algorithm to determine which server is the most accurate and synchronizes to that one. If servers are of similar accuracy, then the **prefer** keyword specifies which of those servers to use. However, if a server is significantly more accurate than the preferred one, the ASA uses the more accurate one. For example, the ASA uses a server of stratum 2 over a server of stratum 3 that is preferred. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Global Configuration | • Yes | • Yes | • Yes | — | • Yes |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was modified to make the source interface optional. |
| 9.12(1) | We added IPv6 support. |
| 9.14(1) | We added NTPv4 support. |

## Usage Guidelines

NTP is used to implement a hierarchical system of servers that provide a precisely synchronized time among network systems. This kind of accuracy is required for time-sensitive operations, such as validating CRLs, which include a precise time stamp. You can configure multiple NTP servers. The ASA chooses the server with the lowest stratum—a measure of how reliable the data is. In multiple context mode, set the NTP server in the system configuration only.

Time derived from an NTP server overrides any time set manually.

The ASA supports NTPv4.

## Examples

The following example identifies two NTP servers and enables authentication for the key IDs 1 and 2:

```
ciscoasa(config)# ntp authenticate
ciscoasa(config)# ntp trusted-key 1
ciscoasa(config)# ntp trusted-key 2
ciscoasa(config)# ntp trusted-key 3
ciscoasa(config)# ntp trusted-key 4
ciscoasa(config)# ntp authentication-key 1 md5
aNiceKey
ciscoasa(config)# ntp authentication-key 2 md5
aNiceKey2
ciscoasa(config)# ntp authentication-key 3 md5 aNiceKey3
ciscoasa(config)# ntp authentication-key 4 md5 aNiceKey4
ciscoasa(config)# ntp server 10.1.1.1 key 1 prefer
ciscoasa(config)# ntp server 10.2.1.1 key 2
ciscoasa(config)# ntp server 2001:DB8::178 key 3
ciscoasa(config)# ntp server 2001:DB8::8945:ABCD key 4
```

## Related Commands

| Command | Description |
|---|---|
| **ntp authenticate** | Enables NTP authentication. |
| **ntp authentication-key** | Sets an encrypted authentication key to synchronize with an NTP server. |
| **ntp trusted-key** | Provides a key ID for the ASA to use in packets for authentication with an NTP server. |
| **show ntp associations** | Shows the NTP servers with which the ASA is associated. |
| **show ntp status** | Shows the status of the NTP association. |

# ntp trusted-key

To specify an authentication key ID to be a trusted key, which is required for authentication with an NTP server, use the **ntp trusted-key** command in global configuration mode. To remove the trusted key, use the **no** form of this command. You can enter multiple trusted keys for use with multiple servers.

**ntp trusted-key** *key_id*
**no ntp trusted-key** *key_id*

## Syntax Description

| | |
|---|---|
| *key_id* | Sets a key ID between 1 and 4294967295. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Global Configuration | • Yes | • Yes | • Yes | • — | Yes |

## Command History

| Release | Modification |
|---|---|
| 7.0(1) | This command was added. |

## Usage Guidelines

To use NTP authentication, also configure the **ntp authenticate** command and **ntp server key** command. To synchronize with a server, set the authentication key for the key ID using the **ntp authentication-key** command.

## Examples

The following example identifies two NTP servers and enables authentication for the key IDs 1 and 2:

```
ciscoasa(config)# ntp server 10.1.1.1 key 1 prefer
ciscoasa(config)# ntp server 10.2.1.1 key 2
ciscoasa(config)# ntp authenticate
ciscoasa(config)# ntp trusted-key 1
ciscoasa(config)# ntp trusted-key 2
ciscoasa(config)# ntp authentication-key 1 md5
aNiceKey
ciscoasa(config)# ntp authentication-key 2 md5
aNiceKey2
```

## Related Commands

| Command | Description |
|---|---|
| **ntp authenticate** | Enables NTP authentication. |

| Command | Description |
|---|---|
| **ntp authentication-key** | Sets an encrypted authentication key to synchronize with an NTP server. |
| **ntp server** | Identifies an NTP server. |
| **show ntp associations** | Shows the NTP servers with which the ASA is associated. |
| **show ntp status** | Shows the status of the NTP association. |

# num-packets

To specify the number of request packets sent during an SLA operation, use the **num-packets** command in sla monitor protocol configuration mode. To restore the default value, use the **no** form of this command.

**num-packets** *number*
**no num-packets** *number*

## Syntax Description

*number* The number of packets sent during an SLA operation. Valid values are from 1 to 100.

**Note** When all the packets specified as the number argument (in this command) are lost, the tracked route has failed.

## Command Default

The default number of packets sent for echo types is 1.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Sla monitor protocol configuration | • Yes | • — | • Yes | • — | — |

## Command History

| Release | Modification |
|---|---|
| 7.2(1) | This command was added. |

## Usage Guidelines

Increase the default number of packets sent to prevent incorrect reachability information due to packet loss.

## Examples

The following example configures an SLA operation with an ID of 123 that uses an ICMP echo request/response time probe operation. It sets the payload size of the echo request packets to 48 bytes and the number of echo requests sent during an SLA operation to 5. All 5 packets must be lost before the tracked route is removed

```
ciscoasa(config)# sla monitor 123
ciscoasa(config-sla-monitor)# type echo protocol ipIcmpEcho 10.1.1.1 interface outside

ciscoasa(config-sla-monitor-echo)# num-packets 5
ciscoasa(config-sla-monitor-echo)# request-data-size 48
ciscoasa(config-sla-monitor-echo)# timeout 4000
ciscoasa(config-sla-monitor-echo)# threshold 2500
ciscoasa(config-sla-monitor-echo)# frequency 10
ciscoasa(config)# sla monitor schedule 123 life forever start-time now
ciscoasa(config)# track 1 rtr 123 reachability
```

**Related Commands**

| Command | Description |
|---|---|
| **request-data-size** | Specifies the size of the request packet payload. |
| **sla monitor** | Defines an SLA monitoring operation. |
| **type echo** | Configures the SLA operation as an echo response time probe operation. |

# nve

To create the Network Virtualization Endpoint (NVE) instance for VXLAN encapsulation, use the **nve** command in global configuration mode. To remove the NVE instance, use the **no** form of this command.

**nve 1**
**no nve 1**

## Syntax Description

| | |
|---|---|
| **1** | Specifies the NVE instance, which is always 1. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Global Configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.4(1) | This command was added. |

## Usage Guidelines

You can configure one VTEP source interface per ASA or per security context. You can configure one NVE instance that specifies this VTEP source interface. All VNI interfaces must be associated with this NVE instance.

## Examples

The following example configures the GigabitEthernet 1/1 interface as the VTEP source interface, and associates the VNI 1 interface wth it:

```
ciscoasa(config)# interface gigabitethernet 1/1
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0
ciscoasa(config)# nve 1
ciscoasa(cfg-nve)# source-interface outside
ciscoasa(config)# interface vni 1
ciscoasa(config-if)# segment-id 1000
ciscoasa(config-if)# vtep-nve 1
ciscoasa(config-if)# nameif vxlan1000
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0 standby 10.1.1.2
ciscoasa(config-if)# ipv6 address 2001:0DB8::BA98:0:3210/48
ciscoasa(config-if)# security-level 50
ciscoasa(config-if)# mcast-group 236.0.0.100
```

**Related Commands**

| Command | Description |
|---|---|
| **debug vxlan** | Debugs VXLAN traffic. |
| **default-mcast-group** | Specifies a default multicast group for all VNI interfaces associated with the VTEP source interface. |
| **encapsulation vxlan** | Sets the NVE instance to VXLAN encapsulation. |
| **inspect vxlan** | Enforces compliance with the standard VXLAN header format. |
| **interface vni** | Creates the VNI interface for VXLAN tagging. |
| **mcast-group** | Sets the multicast group address for the VNI interface. |
| **nve** | Specifies the Network Virtualization Endpoint instance. |
| nve-only | Specifies that the VXLAN source interface is NVE-only. |
| **peer ip** | Manually specifies the peer VTEP IP address. |
| **segment-id** | Specifies the VXLAN segment ID for a VNI interface. |
| **show arp vtep-mapping** | Displays MAC addresses cached on the VNI interface for IP addresses located in the remote segment domain and the remote VTEP IP addresses. |
| **show interface vni** | Shows the parameters, status and statistics of a VNI interface, status of its bridged interface (if configured), and NVE interface it is associated with. |
| **show mac-address-table vtep-mapping** | Displays the Layer 2 forwarding table (MAC address table) on the VNI interface with the remote VTEP IP addresses. |
| **show nve** | Shows the parameters, status and statistics of a NVE interface, status of its carrier interface (source interface), IP address of the carrier interface, VNIs that use this NVE as the VXLAN VTEP, and peer VTEP IP addresses associated with this NVE interface. |
| **show vni vlan-mapping** | Shows the mapping between VNI segment IDs and VLAN interfaces or physical interfaces in transparent mode. |
| **source-interface** | Specifies the VTEP source interface. |
| **vtep-nve** | Associates a VNI interface with the VTEP source interface. |
| **vxlan port** | Sets the VXLAN UDP port. By default, the VTEP source interface accepts VXLAN traffic to UDP port 4789. |

# nve-only

To specify that the VXLAN source interface is NVE-only, use the **nve-only** command in interface configuration mode. To remove the NVE-only restriction, use the **no** form of this command.

**nve-only**
[ **cluster** ]
**no nve-only**

## Syntax Description

## Syntax Description

| | |
|---|---|
| **cluster** | When configuring ASA virtual clustering, you must specify **nev-only cluster** for the cluster control link. |

## Command Default

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| Command Mode | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | Routed | Transparent | Single | Multiple | |
| | | | | Context | System |
| Interface configuration | • Yes | • Yes | • Yes | • Yes | — |

## Command History

| Release | Modification |
|---|---|
| 9.4(1) | This command was added. |
| 9.17(1) | We added the **cluster** keyword to support ASA virtual clustering. |

## Usage Guidelines

You can configure one VTEP source interface per ASA or per security context. The VTEP is defined as a Network Virtualization Endpoint (NVE); VXLAN VTEP is the only supported NVE at this time.

In transparent mode, the **nve-only** setting is required for the VTEP interface and lets you configure an IP address for the interface. This command is optional for routed mode where this setting restricts traffic to VXLAN and common management traffic only on this interface.

For ASA virtual clustering, you must use a VXLAN interface for the cluster control link; in this case, specify **nev-only cluster**.

## Examples

The following example configures the GigabitEthernet 1/1 interface as the VTEP source interface, and specifies that it is NVE-only:

```
ciscoasa(config)# interface gigabitethernet 1/1
ciscoasa(config-if)# nve-only
ciscoasa(config-if)# nameif outside
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0
```

```
ciscoasa(config-if)# nve 1
ciscoasa(cfg-nve)# source-interface outside
```

## Related Commands

| Command | Description |
|---|---|
| **debug vxlan** | Debugs VXLAN traffic. |
| **default-mcast-group** | Specifies a default multicast group for all VNI interfaces associated with the VTEP source interface. |
| **encapsulation vxlan** | Sets the NVE instance to VXLAN encapsulation. |
| **inspect vxlan** | Enforces compliance with the standard VXLAN header format. |
| **interface vni** | Creates the VNI interface for VXLAN tagging. |
| **mcast-group** | Sets the multicast group address for the VNI interface. |
| **nve** | Specifies the Network Virtualization Endpoint instance. |
| nve-only | Specifies that the VXLAN source interface is NVE-only. |
| **peer ip** | Manually specifies the peer VTEP IP address. |
| **segment-id** | Specifies the VXLAN segment ID for a VNI interface. |
| **show arp vtep-mapping** | Displays MAC addresses cached on the VNI interface for IP addresses located in the remote segment domain and the remote VTEP IP addresses. |
| **show interface vni** | Shows the parameters, status and statistics of a VNI interface, status of its bridged interface (if configured), and NVE interface it is associated with. |
| **show mac-address-table vtep-mapping** | Displays the Layer 2 forwarding table (MAC address table) on the VNI interface with the remote VTEP IP addresses. |
| **show nve** | Shows the parameters, status and statistics of a NVE interface, status of its carrier interface (source interface), IP address of the carrier interface, VNIs that use this NVE as the VXLAN VTEP, and peer VTEP IP addresses associated with this NVE interface. |
| **show vni vlan-mapping** | Shows the mapping between VNI segment IDs and VLAN interfaces or physical interfaces in transparent mode. |
| **source-interface** | Specifies the VTEP source interface. |
| **vtep-nve** | Associates a VNI interface with the VTEP source interface. |
| **vxlan port** | Sets the VXLAN UDP port. By default, the VTEP source interface accepts VXLAN traffic to UDP port 4789. |