CHAPTER 36

nac-authentication-server-group through num-packets Commands

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.

Defaults

This command has no arguments or keywords.

Command Modes

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

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

Command History

Release	Modification
7.3(0)	This command was deprecated. The **authentication-server-group** command in nac-policy-nac-framework configuration mode replaced it.
7.2(1)	This command was introduced.

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:

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

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

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

```
hostname(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

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. 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.

Defaults

This command has no default settings.

Command Modes

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

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

Command History

Release	Modification
8.0(2)	This command was introduced.

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:

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

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

```
hostname(config)# no nac-policy nac-framework1
hostname(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

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.

Defaults

This command has no arguments or keywords.

Command Modes

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

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

Command History

Release	Modification
8.0(2)	This command was introduced.

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:

```
hostname(config-group-policy)# no nac-settings
hostname(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.

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

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.

Defaults

No default behaviors or values.

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was introduced.
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:

```
hostname(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.

```
hostname(config)# names
hostname(config)# name 192.168.42.3 sa_inside
hostname(config)# name 209.165.201.3 sa_outside

hostname(config-if)# ip address inside sa_inside 255.255.255.0
hostname(config-if)# ip address outside sa_outside 255.255.255.224

hostname(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

hostname(config)# no names
hostname(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

hostname(config)# names
hostname(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.

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.

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
8.2(1)	This command was introduced.

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:

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

Command	Description
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.

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.

Defaults

No default behavior or values.

Command Modes

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

	Firewall Mode		Security Context		
				Multiple	
Command Mode	Routed	Transparent	Single	Context	System
Interface configuration	•	•	•	•	—

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:”

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

Defaults

No default behaviors or values.

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was introduced.

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.

```
hostname(config)# names
hostname(config)# name 192.168.42.3 sa_inside
hostname(config)# name 209.165.201.3 sa_outside

hostname(config-if)# ip address inside sa_inside 255.255.255.0
```

```
hostname(config-if)# ip address outside sa_outside 255.255.255.224

hostname(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

hostname(config)# no names
hostname(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

hostname(config)# names
hostname(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

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).

Defaults

The default is ":" (colon).

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was introduced.

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:

```
hostname(config)# pop3s
hostname(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, use the **name-server** command in dns server-group configuration mode. To remove a server or servers, use the **no** form of this command. The ASA uses DNS to resolve server names in your SSL VPN configuration or certificate configuration (see "Usage Guidelines" for a list of supported commands). Other features that define server names (such as AAA) do not support DNS resolution. You must enter the IP address or manually resolve the name to an IP address by using the **name** command.

name-server *ip_address* [*ip_address2*] [...] [*ip_address6*]

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

Syntax Description

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.

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
7.1(1)	This command was introduced.

Usage Guidelines

To enable DNS lookup, configure the **domain-name** command in dns server-group configuration mode. If you do not enable DNS lookup, the DNS servers are not used.

SSL VPN commands that support DNS resolution include the following:

- **server (pop3s)**
- **server (imap4s)**
- **server (smtps)**
- **port-forward**
- **url-list**

Certificate commands that support DNS resolution include the following:

- **enrollment url**
- **url**

You can manually enter names and IP addresses using the **name** command.

Examples

The following example adds three DNS servers to the group "dnsgroup1":

```
hostname(config)# dns server-group dnsgroup1
hostname(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:

```
hostname(config)# dns server-group dnsgroup1
hostname(config-dns-server-group)# name-server 10.5.1.1 10.8.3.8
```

To verify the dns server group configuration, enter the **show running-config dns** command in global configuration mode:

```
hostname(config)# show running-config dns
name-server 10.1.1.1
name-server 10.2.3.4
name-server 192.168.5.5
name-server 10.5.1.1
name-server 10.8.3.8
...
```

Or you can enter them as two separate commands:

```
hostname(config)# dns server-group dnsgroup1
hostname(config-dns-server-group)# name-server 10.5.1.1
hostname(config)# name-server 10.8.3.8
```

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

```
hostname(config)# dns server-group dnsgroup1
hostname(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*] [**pat-pool** *mapped_obj* [**round-robin**] [**extended**] [**flat** [**include-reserve**]]]
[**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*] [**pat-pool** *mapped_obj* [**round-robin**] [**extended**] [**flat** [**include-reserve**]]]
[**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. In transparent mode, you must specify the real and mapped interfaces; you cannot use **any**. 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: • 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. **any** is not available in transparent mode. • Static NAT source real and mapped IP addresses—You can specify **source static any any** to enable identity NAT for all addresses. • Dynamic NAT or PAT source real addresses—You can translate all addresses on the source interface by specifying **source dynamic any** *mapped_obj*. 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. **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.
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. 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**]	(Optional) 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.
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. 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. For dynamic PAT, you can specify **interface** alone for the source mapped address. For static NAT with port translation (source or destination), be sure to also configure the **service** keyword. For this option, you must configure a specific interface for the *mapped_ifc*. This option is not available in transparent mode.
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**). For dynamic NAT, you typically configure a larger group of addresses to be mapped to a smaller group. **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. 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. 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. 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. • 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. • 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. 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. 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. 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.

Defaults

- 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:

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

Command History

Release	Modification
8.3(1)	This command was introduced.
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. The default behavior for identity NAT was changed to have proxy ARP enabled, matching other static NAT rules. 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.
8.4(3)	The **extended**, **flat**, and **include-reserve** keywords were added. 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. *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.

Usage Guidelines

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.

- 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.
- (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.

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:

- 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*.

```
hostname(config)# object network myInsideNetwork
hostname(config-network-object)# subnet 10.1.2.0 255.255.255.0

hostname(config)# object network DMZnetwork1
hostname(config-network-object)# subnet 209.165.201.0 255.255.255.224

hostname(config)# object network PATaddress1
hostname(config-network-object)# host 209.165.202.129

hostname(config)# nat (inside,dmz) source dynamic myInsideNetwork PATaddress1 destination
static DMZnetwork1 DMZnetwork1

hostname(config)# object network DMZnetwork2
hostname(config-network-object)# subnet 209.165.200.224 255.255.255.224

hostname(config)# object network PATaddress2
hostname(config-network-object)# host 209.165.202.130

hostname(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*.

```
hostname(config)# object network myInsideNetwork
hostname(config-network-object)# subnet 10.1.2.0 255.255.255.0

hostname(config)# object network TelnetWebServer
hostname(config-network-object)# host 209.165.201.11

hostname(config)# object network PATaddress1
hostname(config-network-object)# host 209.165.202.129

hostname(config)# object service TelnetObj
hostname(config-network-object)# service tcp destination eq telnet

hostname(config)# nat (inside,outside) source dynamic myInsideNetwork PATaddress1
destination static TelnetWebServer TelnetWebServer service TelnetObj TelnetObj

hostname(config)# object network PATaddress2
hostname(config-network-object)# host 209.165.202.130

hostname(config)# object service HTTPObj
hostname(config-network-object)# service tcp destination eq http

hostname(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.

```
hostname(config)# object service FTP_PASV_PORT_RANGE
hostname(config-service-obvject)# service tcp source range 65000 65004

hostname(config)# object network HOST_FTP_SERVER
hostname(config-network-obvject)# host 192.168.10.100

hostname(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:

```
hostname(config)# object network INSIDE_NW
hostname(config-network-object)# subnet 2001:DB8:AAAA::/96

hostname(config)# object network MAPPED_1
hostname(config-network-object)# range 209.165.200.225 209.165.200.254

hostname(config)# object network MAPPED_2
hostname(config-network-object)# range 209.165.202.129 209.165.200.158

hostname(config)# object network SERVERS_1
hostname(config-network-object)# subnet 209.165.201.0 255.255.255.224

hostname(config)# object network SERVERS_2
hostname(config-network-object)# subnet 203.0.113.0 255.255.255.0

hostname(config)# nat (inside,outside) source dynamic INSIDE_NW MAPPED_1 destination
static SERVERS_1 SERVERS_1
hostname(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.

```
hostname(config)# object network INSIDE_NW
hostname(config-network-object)# subnet 192.168.1.0 255.255.255.0

hostname(config)# object network PAT_POOL
hostname(config-network-object)# range 2001:DB8:AAAA::1 2001:DB8:AAAA::200

hostname(config)# object network TELNET_SVR
hostname(config-network-object)# host 2001:DB8::23

hostname(config)# object service TELNET
hostname(config-service-object)# service tcp destination eq 23

hostname(config)# object network SERVERS
hostname(config-network-object)# subnet 2001:DB8:AAAA::/96

hostname(config)# nat (inside,outside) source dynamic INSIDE_NW interface ipv6 destination
static TELNET_SVR TELNET_SVR service TELNET TELNET
hostname(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.

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**]]] [**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**]]] [**interface** [**ipv6**]]} [**dns**]

For static NAT and static NAT with port translation:

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

no nat [(*real_ifc*,*mapped_ifc*)] **static** {*mapped_inline_ip* | *mapped_obj* | **interface** [**ipv6**]} [**net-to-net**] [**dns** | **service** {**tcp** | **udp**} *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. In transparent mode, you must specify the real and mapped interfaces; you cannot use **any**.
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). 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**]	(Optional) 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.

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. 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. If you specify **ipv6**, then the IPv6 address of the interface is used. For static NAT with port translation, you can specify the **interface** keyword if you also configure the **service** keyword. For this option, you must configure a specific interface for the *mapped_ifc*. You cannot specify **interface** in transparent mode.
mapped_inline_host_ip	Specifies the mapped address as an inline value. If you specify **dynamic**, then using a host IP address configures dynamic PAT.
mapped_inline_ip	For static NAT, specifies the mapped IP address as an inline value. 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 a host address. In the case of a range, then the mapped addresses include the same number of addresses as the real range. 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.
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. 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" section on page 36-34 for information about disallowed mapped IP addresses. 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 or UDP 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. 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 or UDP 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**}	(Optional) For static NAT with port translation, specifies the protocol for port translation. Only TCP and UDP are supported.
static	Configures static NAT or static NAT with port translation.

Defaults

- 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:

	Firewall Mode		Security Context		
				Multiple	
Command Mode	**Routed**	**Transparent**	**Single**	**Context**	**System**
Object network configuration	•	•	•	•	—

Command History

Release	Modification
8.3(1)	This command was introduced.
8.4(2)/8.5(1)	The **no-proxy-arp**, **route-lookup**, **pat-pool**, and **round-robin** keywords were added. The default behavior for identity NAT was changed to have proxy ARP enabled, matching other static NAT rules. 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.

Release	Modification
8.4(3)	The **extended**, **flat**, and **include-reserve** keywords were added. 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. *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.

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 discontinous 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.
- 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.
- (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:

- 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:

```
hostname(config)# object network my-range-obj
hostname(config-network-object)# range 2.2.2.1 2.2.2.10
hostname(config)# object network my-inside-net
hostname(config-network-object)# subnet 192.168.2.0 255.255.255.0
hostname(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.

```
hostname(config)# object network nat-range1
hostname(config-network-object)# range 10.10.10.10 10.10.10.20

hostname(config-network-object)# object network pat-ip1
hostname(config-network-object)# host 10.10.10.21

hostname(config-network-object)# object-group network nat-pat-grp
hostname(config-network-object)# network-object object nat-range1
hostname(config-network-object)# network-object object pat-ip1

hostname(config-network-object)# object network my_net_obj5
hostname(config-network-object)# subnet 10.76.11.0 255.255.255.0
hostname(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.

```
hostname(config)# object network IPv4_NAT_RANGE
hostname(config-network-object)# range 209.165.201.1 209.165.201.30

hostname(config-network-object)# object network IPv4_PAT
hostname(config-network-object)# host 209.165.201.31

hostname(config-network-object)# object-group network IPv4_GROUP
hostname(config-network-object)# network-object object IPv4_NAT_RANGE
hostname(config-network-object)# network-object object IPv4_PAT

hostname(config-network-object)# object network my_net_obj5
hostname(config-network-object)# subnet 2001:DB8::/96
hostname(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:

```
hostname(config)# object network my-inside-net
hostname(config-network-object)# subnet 192.168.2.0 255.255.255.0
hostname(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:

```
hostname(config)# object network my-inside-net
hostname(config-network-object)# subnet 192.168.2.0 255.255.255.0
hostname(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:

```
hostname(config)# object network IPv4_POOL
hostname(config-network-object)# range 203.0.113.1 203.0.113.254
hostname(config)# object network IPv6_INSIDE
hostname(config-network-object)# subnet 2001:DB8::/96
hostname(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.

```
hostname(config)# object network my-host-obj1
hostname(config-network-object)# host 1.1.1.1
hostname(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.

```
hostname(config)# object network my-mapped-obj
hostname(config-network-object)# host 2.2.2.2

hostname(config-network-object)# object network my-host-obj1
hostname(config-network-object)# host 1.1.1.1
hostname(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.

```
hostname(config)# object network my-ftp-server
hostname(config-network-object)# host 1.1.1.1
hostname(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.

```
hostname(config)# object network inside_v4_v6
hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) static 2001:DB8::/96
```

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

```
hostname(config)# object network inside_v6
hostname(config-network-object)# subnet 2001:DB8:AAAA::/96
hostname(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:

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

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

```
hostname(config)# object network my-host-obj1-identity
hostname(config-network-object)# host 10.1.1.1

hostname(config-network-object)# object network my-host-obj1
hostname(config-network-object)# host 10.1.1.1
hostname(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.

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.

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was introduced.

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:

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

```
hostname(config-load-balancing)# cluster port 9023
hostname(config-load-balancing)# participate
hostname(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:

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

Command History

Release	Modification
8.4(3)	We introduced this command.

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 AnyConnect 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:

```
hostname# ip local pool client 10.1.226.4-10.1.226.254
hostname# tunnel-group vpnclient type remote-access
hostname# tunnel-group vpnclient general-attributes
hostname(config-tunnel-general)# address-pool client
hostname(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:

```
hostname# 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 addressess 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.

Defaults

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:

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

Command History

Release	Modification
7.2(1)	This command was introduced.

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:

```
hostname(config)# policy-map type inspect dns preset_dns_map
hostname(config-pmap)# parameters
hostname(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.

Command	Description
policy-map	Creates a Layer 3/4 policy map.
show running-config policy-map	Display all current policy map configurations.

nbns-server (tunnel-group webvpn attributes mode)

To configure an NBNS server, use the **nbns-server** command in tunnel-group webvpn 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.

Defaults

No NBNS server is configured by default.

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was introduced.
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.

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

nbns-server (webvpn mode)

To configure an NBNS server, use the **nbns-server** command in tunnel-group webvpn 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.

Defaults

No NBNS server is configured by default.

Command Modes

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

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

Command History

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

Usage Guidelines

This command is deprecated in webvpn configuration mode. The nbns-server command in tunnel-group webvpn-attributes configuration mode replaces it. 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 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 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.

```
hostname(config)# webvpn
hostname(config-webvpn)# nbns-server 10.10.10.19 master timeout 10 retry 8
hostname(config-webvpn)# nbns-server 10.10.10.24 timeout 15 retry 8
```

neighbor

To define a static neighbor on a point-to-point, non-broadcast network, use the **neighbor** command in router 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.

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was introduced.
9.0(1)	Multiple context mode is supported.

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:

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

Related Commands

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

neighbor (EIGRP)

To define an EIGRP neighbor router with which to exchange routing information, use the **neighbor** command in router 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	IP address of the neighbor router.

Defaults

No default behavior or values.

Command Modes

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

	Firewall Mode		Security Context		
				Multiple	
Command Mode	**Routed**	**Transparent**	**Single**	**Context**	**System**
Router configuration	•	—	•	—	—

Command History

Release	Modification
8.0(2)	This command was introduced.

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:

```
hostname(config)# router eigrp 100
hostname(config-router)# network 192.168.0.0
```

```
hostname(config-router)# neighbor 192.168.1.1 interface outside
hostname(config-router)# neighbor 192.168.2.2 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.

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.

Defaults

Network extension mode is disabled.

Command Modes

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

	Firewall Mode		Security Context		
				Multiple	
Command Mode	Routed	Transparent	Single	Context	System
Group-policyconfiguration	•	—	•	—	—

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 introduced.

Examples

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

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

network

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*

no network *ip_addr*

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.

Defaults

No networks are specified.

Command Modes

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

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

Command History

Release	Modification
7.2(1)	This command was introduced.
9.0(1)	Multiple context mode is supported.

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:

```
hostname(config)# router rip
hostname(config-router)# network 10.0.0.0
hostname(config-router)# network 192.168.7.0
```

Related Commands

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

network (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.

Defaults

No networks are specified.

Command Modes

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

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

Command History

Release	Modification
8.0(2)	This command was introduced.

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:

```
hostname(config)# router eigrp 100
hostname(config-router)# network 10.0.0.0 255.0.0.0
hostname(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.
show eigrp topology	Displays the EIGRP topology table.

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 ACL, use the command without arguments.

network-acl *name*

no network-acl [*name*]

Syntax Description

name	Specifies the name of the network ACL. Maximum 240 characters.

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
8.0(2)	This command was introduced.

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.

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

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was introduced.

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:

```
hostname(config-router)# network 192.168.1.1 255.255.255.0 area 2
```

Related Commands

Command	Description
router ospf	Enters router configuration mode.
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 *ip_address* | *ip_address mask* | **object** *name*}

no network-object {host *ip_address* | *ip_address mask* | **object** *name*}

Syntax Description

host *ip_address*	Specifies a host IP address.
ip_address mask	Specifies the network address and subnet mask.
object *name*	Specifies a network object (**object network** command).

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
8.3(1)	The **object** argument was added to support network objects (**object network** command).

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:

```
hostname(config)# object-group network sjj_eng_ftp_servers
hostname(config-network-object-group)# network-object host sjj.eng.ftp
hostname(config-network-object-group)# network-object host 172.16.56.195
hostname(config-network-object-group)# network-object 192.168.1.0 255.255.255.224
hostname(config-network-object-group)# group-object sjc_eng_ftp_servers
hostname(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.
service-object	Adds a service object to a service object group.
show running-config object-group	Displays the current object groups.

nop

To define an action when the No Operation IP option occurs in a packet 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	Instructs the ASA to allow a packet containing the No Operation IP option to pass.
clear	Instructs the ASA to clear the No Operation IP option from a packet and then allow the packet to pass.

Defaults

By default, IP Options inspection, drops packets containing the No Operation IP option.

Command Modes

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

	Firewall Mode		Security Context		
				Multiple	
Command Mode	**Routed**	**Transparent**	**Single**	**Context**	**System**
Parameters configuration	•	•	•	•	—

Command History

Release	Modification
8.2(2)	This command was introduced.

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. Configuring this inspection instructs the ASA to allow a packet to pass or to 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:

```
hostname(config)# policy-map type inspect ip-options ip-options_map
hostname(config-pmap)# parameters
hostname(config-pmap-p)# eool action allow
hostname(config-pmap-p)# nop action allow
```

```
hostname(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.

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.

Defaults

No default behaviors or values.

Command Modes

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

	Firewall Mode		Security Context		
				Multiple	
Command Mode	Routed	Transparent	Single	Context	System
Aaa-server host configuration	•	•	•	•	—

Command History

Release	Modification
7.0(1)	This command was introduced.

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":

```
hostname(config)# aaa-server svrgrp1 protocol nt
hostname(configaaa-sesrver-group)# aaa-server svrgrp1 host 1.2.3.4
hostname(config-aaa-server-host)# nt-auth-domain-controller primary1
hostname(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.
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.

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was introduced.

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). The ASA also uses an authentication key to synchronize with the NTP server (see the **ntp authentication-key** command).

Examples

The following example configures the ASA to synchronize only to systems that provide authentication key 42 in their NTP packets:

```
hostname(config)# ntp authenticate
hostname(config)# ntp authentication-key 42 md5 aNiceKey
hostname(config)# ntp trusted-key 42
```

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.

Command	Description
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** *key*

no ntp authentication-key *key_id* [**md5** [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, which is the only algorithm supported.

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

To use NTP authentication, also configure the **ntp authenticate** command.

Examples

The following example enables authentications, identifies trusted key IDs 1 and 2, and sets authentication keys for each trusted key ID:

```
hostname(config)# ntp authenticate
hostname(config)# ntp trusted-key 1
hostname(config)# ntp trusted-key 2
hostname(config)# ntp authentication-key 1 md5 aNiceKey
hostname(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 IP address or hostname 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.

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was modified to make the source interface optional.

Usage Guidelines

You can identify multiple servers; the ASA uses the most accurate server. In multiple context mode, set the NTP server in the system configuration only.

Examples

The following example identifies two NTP servers and enables authentication for the key IDs 1 and 2:

```
hostname(config)# ntp server 10.1.1.1 key 1 prefer
```

```
hostname(config)# ntp server 10.2.1.1 key 2
hostname(config)# ntp authenticate
hostname(config)# ntp trusted-key 1
hostname(config)# ntp trusted-key 2
hostname(config)# ntp authentication-key 1 md5 aNiceKey
hostname(config)# ntp authentication-key 2 md5 aNiceKey2
```

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.

Defaults

No default behavior or values.

Command Modes

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

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

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

To use NTP authentication, also configure the **ntp authenticate** command. To synchronize with a server, set the authentication key for the key ID using the **ntp authentication-key** command.

Examples

The following example enables authentications, identifies trusted key IDs 1 and 2, and sets authentication keys for each trusted key ID:

```
hostname(config)# ntp authenticate
hostname(config)# ntp trusted-key 1
hostname(config)# ntp trusted-key 2
hostname(config)# ntp authentication-key 1 md5 aNiceKey
hostname(config)# ntp authentication-key 2 md5 aNiceKey2
```

Related Commands

Command	Description
ntp authenticate	Enables NTP authentication.
ntp authentication-key	Sets an encrypted authentication key to synchronize with an NTP server.
ntp server	Identifies an NTP server.

Command	Description
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.

Defaults

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:

	Firewall Mode		Security Context		
				Multiple	
Command Mode	Routed	Transparent	Single	Context	System
sla monitor protocol configuration	•	—	•	—	—

Command History

Release	Modification
7.2(1)	This command was introduced.

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.

```
hostname(config)# sla monitor 123
hostname(config-sla-monitor)# type echo protocol ipIcmpEcho 10.1.1.1 interface outside
hostname(config-sla-monitor-echo)# num-packets 5
hostname(config-sla-monitor-echo)# request-data-size 48
hostname(config-sla-monitor-echo)# timeout 4000
hostname(config-sla-monitor-echo)# threshold 2500
hostname(config-sla-monitor-echo)# frequency 10
hostname(config)# sla monitor schedule 123 life forever start-time now
hostname(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.