CHAPTER 21

nac through override-account-disable Commands

nac

To enable or disable Network Admission Control, use the **nac** command in group-policy configuration mode. To inherit the NAC setting from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command.

nac {enable | disable}

no nac [enable | disable]

Syntax Description

enable	Enables NAC, which requires posture validation for remote access. If the remote computer passes the validation checks, the ACS server downloads the access policy for the security appliance to enforce
disable	Disables NAC.

Defaults

The default setting 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-policy configuration	•	—	•	—	—

Command History

Release	Modification
7.2(1)	This command was introduced.

Usage Guidelines

An Access Control Server must be present on the network.

Examples

The following example enables NAC for the group policy:

```
hostname(config-group-policy)# nac enable
hostname(config-group-policy)
```

The following example disables NAC for the group policy:

```
hostname(config-group-policy)# nac disable
hostname(config-group-policy)
```

The following example inherits the NAC setting from the default group policy:

```
hostname(config-group-policy)# no nac
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-authentication-server-group	Identifies the group of authentication servers to be used for Network Admission Control Posture Validation

nac-authentication-server-group

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

SyntaxDescription

server-group	Name of the posture validation server group, as configured on the security appliance 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
7.2(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-default-acl

To specify the ACL to be used as the default ACL for Network Admission Control sessions that fail posture validation, use the **nac-default-acl** command in group-policy configuration mode.

nac-default-acl value *acl-name*

nac-default-acl none

To inherit the ACL from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command.

no nac-default-acl

Syntax Description

acl-name	Names the posture validation server group, as configured on the security appliance using the **aaa-server host** command. The name must match the server-tag variable specified in that command.
none	Disables inheritance of the ACL from the default group policy and does not apply an ACL to NAC sessions that fail posture validation.

Defaults

The default setting is **none**.

Because NAC is disabled by default, VPN traffic traversing the security appliance is not subject to the NAC Default ACL until NAC is enabled.

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
7.2(1)	This command was introduced.

Examples

The following example identifies acl-1 as the ACL to be applied when posture validation fails:

```
hostname(config-group-policy)# nac-default-acl value acl-1
hostname(config-group-policy)
```

The following example inherits the ACL from the default group policy.

```
hostname(config-group-policy)# no nac-default-acl
hostname(config-group-policy)
```

The following example disables inheritance of the ACL from the default group policy and does not apply an ACL to NAC sessions that fail posture validation:

```
hostname(config-group-policy)# nac-default-acl none
hostname(config-group-policy)
```

Related Commands

Command	Description
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-reval-period

To specify the interval between each successful posture validation in a Network Admission Control session, use the **nac-reval-period** command in group-policy configuration mode. To inherit the value of the Revalidation Timer from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command.

nac-reval-period *seconds*

no nac-reval-period [*seconds*]

Syntax Description

seconds	Number of seconds between each successful posture validation. The range is 300 to 86400.

Defaults

The default value is 36000.

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
7.2(1)	This command was introduced.

Usage Guidelines

The security appliance starts the Revalidation Timer after each successful posture validation. The expiration of this timer triggers the next unconditional posture validation. The security appliance maintains posture validation during revalidation. The default group policy becomes effective if the Access Control Server is unavailable during posture validation or revalidation.

Examples

The following example changes the revalidation timer to 86400 seconds:

```
hostname(config-group-policy)# nac-reval-period 86400
hostname(config-group-policy)
```

The following example inherits the value of the revalidation timer from the default group policy:

```
hostname(config-group-policy)# no nac-reval-period
hostname(config-group-policy)
```

Related Commands

Command	Description
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-sq-period

To specify the interval between each successful posture validation in a Network Admission Control session and the next query for changes in the host posture, use the **nac-sq-period** command in group-policy configuration mode. To inherit the value of the status query timer from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command.

nac-sq-period *seconds*

no nac-sq-period [*seconds*]

Syntax Description

seconds	Number of seconds between each successful posture validation. The range is 300 to 1800.

Defaults

The default value is 300.

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
7.2(1)	This command was introduced.

Usage Guidelines

The security appliance starts the status query timer after each successful posture validation and status query response. The expiration of this timer triggers a query for changes in the host posture, referred to as a *status query.*

Examples

The following example changes the value of the status query timer to 1800 seconds:

```
hostname(config-group-policy)# nac-sq-period 1800
hostname(config-group-policy)
```

The following example inherits the value of the status query timer from the default group policy:

```
hostname(config-group-policy)# no nac-sq-period
hostname(config-group-policy)
```

Related Commands

Command	Description
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-reval-period	Specifies the interval between each successful posture validation in a Network Admission Control session

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
Preexisting	This command was preexissting.
7.0(4)	This command was enhanced to include an optional description.

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.

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 security appliance 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
Preexisting	This command was preexisting.

Usage Guidelines

The **names** command is used to enable the association of a name with an IP address that you configured with the **name** command. The order in which you enter the **name** or **names** commands is irrelevant.

Examples

The following example shows how to enable the association of a name with an IP address:

```
hostname(config)# names
```

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.

nat

To identify addresses on one interface that are translated to mapped addresses on another interface, use the **nat** command in global configuration mode. This command configures dynamic NAT or PAT, where an address is translated to one of a pool of mapped addresses. To remove the **nat** command, use the **no** form of this command.

For regular dynamic NAT:

nat (*real_ifc) nat_id real_ip* [*mask* [**dns**] [**outside]** [[**tcp**] *tcp_max_conns* [*emb_limit*]] [**udp** *udp_max_conns*] [**norandomseq**]]

no nat (*real_ifc) nat_id real_ip* [*mask* [**dns**] [**outside]** [[**tcp**] *tcp_max_conns* [*emb_limit*]] [**udp** *udp_max_conns*] [**norandomseq**]]

For policy dynamic NAT and NAT exemption:

nat (*real_ifc) nat_id* **access-list** *access_list_name* [**dns**] [**outside]** [[**tcp**] *tcp_max_conns* [*emb_limit*]] [**udp** *udp_max_conns*] [**norandomseq**]

no nat (*real_ifc) nat_id* **access-list** *access_list_name* [**dns**] [**outside]** [[**tcp**] *tcp_max_conns* [*emb_limit*]] [**udp** *udp_max_conns*] [**norandomseq**]

Syntax Description

access-list *access_list_name*	Identifies the local addresses and destination addresses using an extended access list, also known as policy NAT. Create the access list using the **access-list** command. You can optionally specify the local and destination ports in the access list using the **eq** operator. If the NAT ID is **0**, then the access list specifies addresses that are exempt from NAT. NAT exemption is not the same as policy NAT; you cannot specify the port addresses, for example. **Note** Access list hit counts, as shown by the **show access-list** command, do not increment for NAT exemption access lists.
dns	(Optional) Rewrites the A record, or address record, in DNS replies that match this command. For DNS replies traversing from a mapped interface to any other interface, the A record is rewritten from the mapped value to the real value. Inversely, for DNS replies traversing from any interface to a mapped interface, the A record is rewritten from the real value to the mapped value. If your NAT statement includes the address of a host that has an entry in a DNS server, and the DNS server is on a different interface from a client, then the client and the DNS server need different addresses for the host; one needs the global address and one needs the local address.The translated host needs to be on the same interface as either the client or the DNS server. Typically, hosts that need to allow access from other interfaces use a static translation, so this option is more likely to be used with the **static** command.

emb_limit	(Optional) Specifies the maximum number of embryonic connections per host. The default is 0, which means unlimited embryonic connections. Limiting the number of embryonic connections protects you from a DoS attack. The security appliance uses the embryonic limit to trigger TCP Intercept, which protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN packets. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination.
real_ifc	Specifies the name of the interface connected to the real IP address network.
real_ip	Specifies the real address that you want to translate. You can use **0.0.0.0** (or the abbreviation **0**) to specify all addresses.
mask	(Optional) Specifies the subnet mask for the real addresses. If you do not enter a mask, then the default mask for the IP address class is used.
nat_id	Specifies an integer for the NAT ID. For regular NAT, this integer is between 1 and 2147483647. For policy NAT (nat id access-list), this integer is between 1 and 65535. Identity NAT (**nat 0**) and NAT exemption (**nat 0 access-list**) use the NAT ID of **0**. This ID is referenced by the **global** command to associate a global pool with the *real_ip*.
norandomseq	(Optional) Disables TCP ISN randomization protection. Each TCP connection has two ISNs: one generated by the client and one generated by the server. The security appliance randomizes the ISN of the TCP SYN passing in both the inbound and outbound directions. Randomizing the ISN of the protected host prevents an attacker from predecting the next ISN for a new connection and potentially hijacking the new session. TCP initial sequence number randomization can be disabled if required. For example: • If another in-line firewall is also randomizing the initial sequence numbers, there is no need for both firewalls to be performing this action, even though this action does not affect the traffic. • If you use eBGP multi-hop through the security appliance, and the eBGP peers are using MD5. Randomization breaks the MD5 checksum. • You use a WAAS device that requires the security appliance not to randomize the sequence numbers of connections.
outside	(Optional) If this interface is on a lower security level than the interface you identify by the matching **global** statement, then you must enter **outside**. This feature is called outside NAT or bidirectional NAT.

tcp *tcp_max_conns*	Specifies the maximum number of simultaneous TCP connections for the entire subnet. The default is 0, which means unlimited connections. (Idle connections are closed after the idle timeout specified by the **timeout conn** command.)
udp *udp_max_conns*	(Optional) Specifies the maximum number of simultaneous UDP connections for the entire subnet. The default is 0, which means unlimited connections. (Idle connections are closed after the idle timeout specified by the **timeout conn** command.)

Defaults

The default value for *tcp_max_conns*, *emb_limit*, and *udp_max_conns* is 0 (unlimited), which is the maximum available.

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
Preexisting	This command was preexisting.

Usage Guidelines

For dynamic NAT and PAT, you first configure a **nat** command identifying the real addresses on a given interface that you want to translate. Then you configure a separate **global** command to specify the mapped addresses when exiting another interface (in the case of PAT, this is one address). Each **nat** command matches a **global** command by comparing the NAT ID, a number that you assign to each command.

The security appliance translates an address when a NAT rule matches the traffic. If no NAT rule matches, processing for the packet continues. The exception is when you enable NAT control using the **nat-control** command. NAT control requires that packets traversing from a higher security interface (inside) to a lower security interface (outside) match a NAT rule, or else processing for the packet stops. NAT is not required between same security level interfaces even if you enable NAT control. You can optionally configure NAT if desired.

Dynamic NAT translates a group of real addresses to a pool of mapped addresses that are routable on the destination network. The mapped pool can include fewer addresses than the real group. When a host you want to translate accesses the destination network, the security appliance assigns it an IP address from the mapped pool. The translation is added only when the real host initiates the connection. The translation is in place only for the duration of the connection, and a given user does not keep the same IP address after the translation times out (see the **timeout xlate** command). Users on the destination network, therefore, cannot reliably initiate a connection to a host that uses dynamic NAT (or PAT, even if the connection is allowed by an access list), and the security appliance rejects any attempt to connect to a real host address directly. See the **static** command for reliable access to hosts.

Dynamic NAT has these disadvantages:

- If the mapped pool has fewer addresses than the real group, you could run out of addresses if the amount of traffic is more than expected.

  Use PAT if this event occurs often, because PAT provides over 64,000 translations using ports of a single address.

- You have to use a large number of routable addresses in the mapped pool; if the destination network requires registered addresses, such as the Internet, you might encounter a shortage of usable addresses.

The advantage of dynamic NAT is that some protocols cannot use PAT. For example, PAT does not work with IP protocols that do not have a port to overload, such as GRE version 0. PAT also does not work with some applications that have a data stream on one port and the control path on another and are not open standard, such as some multimedia applications.

PAT translates multiple real addresses to a single mapped IP address. Specifically, the security appliance translates the real address and source port (real socket) to the mapped address and a unique port (mapped socket). If the source port is TCP/UDP, the source address is translated using PAT to one in the same range. Ranges include: 1–511, 512–1023, and 1024–65535. Each connection requires a separate translation, because the source port differs for each connection. For example, 10.1.1.1:1025 requires a separate translation from 10.1.1.1:1026.

After the connection expires, the port translation also expires after 30 seconds of inactivity. The timeout is not configurable.

PAT lets you use a single mapped address, thus conserving routable addresses. You can even use the security appliance interface IP address as the PAT address. PAT does not work with some multimedia applications that have a data stream that is different from the control path.

Note For the duration of the translation, a remote host can initiate a connection to the translated host if an access list allows it. Because the address (both real and mapped) is unpredictable, a connection to the host is unlikely. However in this case, you can rely on the security of the access list.

If you enable NAT control, then inside hosts must match a NAT rule when accessing outside hosts. If you do not want to perform NAT for some hosts, then you can bypass NAT for those hosts (alternatively, you can disable NAT control). You might want to bypass NAT, for example, if you are using an application that does not support NAT. You can use the **static** command to bypass NAT, or one of the following options:

- Identity NAT (**nat 0** command)—When you configure identity NAT (which is similar to dynamic NAT), you do not limit translation for a host on specific interfaces; you must use identity NAT for connections through all interfaces. Therefore, you cannot choose to perform normal translation on real addresses when you access interface A, but use identity NAT when accessing interface B. Regular dynamic NAT, on the other hand, lets you specify a particular interface on which to translate the addresses. Make sure that the real addresses for which you use identity NAT are routable on all networks that are available according to your access lists.

  For identity NAT, even though the mapped address is the same as the real address, you cannot initiate a connection from the outside to the inside (even if the interface access list allows it). Use static identity NAT or NAT exemption for this functionality.

- NAT exemption (**nat 0 access-list** command)—NAT exemption allows both translated and remote hosts to initiate connections. Like identity NAT, you do not limit translation for a host on specific interfaces; you must use NAT exemption for connections through all interfaces. However, NAT exemption does let you specify the real and destination addresses when determining the real addresses to translate (similar to policy NAT), so you have greater control using NAT exemption.

However unlike policy NAT, NAT exemption does not consider the ports in the access list. You can only apply one NAT exemption rule per interface. If you enter another rule for the same interface, the old rule is overwritten.

Policy NAT lets you identify real addresses for address translation by specifying the source and destination addresses in an extended access list. You can also optionally specify the source and destination ports. Regular NAT can only consider the real addresses. For example, you can translate the real address to mapped address A when it accesses server A, but translate the real address to mapped address B when it accesses server B.

When you specify the ports in policy NAT for applications that require application inspection for secondary channels (FTP, VoIP, etc.), the security appliance automatically translates the secondary ports.

Note All types of NAT support policy NAT except for NAT exemption. NAT exemption uses an access list to identify the real addresses, but differs from policy NAT in that the ports are not considered. You can accomplish the same result as NAT exemption using **static** identity NAT, which does support policy NAT.

You can alternatively set connection limits (but not embryonic connection limits) using the Modular Policy Framework. See the **set connection** commands for more information. You can only set embryonic connection limits using NAT. If you configure these settings for the same traffic using both methods, then the security appliance uses the lower limit. For TCP sequence randomization, if it is disabled using either method, then the security appliance disables TCP sequence randomization.

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.

Examples

For example, to translate the 10.1.1.0/24 network on the inside interface, enter the following command:

```
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.1-209.165.201.30
```

To identify a pool of addresses for dynamic NAT as well as a PAT address for when the NAT pool is exhausted, enter the following commands:

```
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.5
hostname(config)# global (outside) 1 209.165.201.10-209.165.201.20
```

To translate the lower security dmz network addresses so they appear to be on the same network as the inside network (10.1.1.0), for example, to simplify routing, enter the following commands:

```
hostname(config)# nat (dmz) 1 10.1.2.0 255.255.255.0 outside dns
hostname(config)# global (inside) 1 10.1.1.45
```

To identify a single real address with two different destination addresses using policy NAT, enter the following commands:

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
hostname(config)# access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
hostname(config)# nat (inside) 1 access-list NET1 tcp 0 2000 udp 10000
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list NET2 tcp 1000 500 udp 2000
hostname(config)# global (outside) 2 209.165.202.130
```

To identify a single real address/destination address pair that use different ports using policy NAT, enter the following commands:

```
hostname(config)# access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 80
hostname(config)# access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 23
hostname(config)# nat (inside) 1 access-list WEB
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list TELNET
hostname(config)# global (outside) 2 209.165.202.130
```

Related Commands

Command	Description
access-list deny-flow-max	Specifies the maximum number of concurrent deny flows that can be created.
clear configure nat	Removes the NAT configuration.
global	Creates entries from a pool of global addresses.
interface	Creates and configures an interface.
show running-config nat	Displays a pool of global IP addresses that are associated with the network.

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 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	•	—	•	—	—

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
```

Related Commandsh

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

nat-control

To enforce NAT control, use the **nat-control** command in global configuration mode. NAT control requires NAT for inside hosts when they access the outside. To disable NAT control, use the **no** form of this command.

nat-control

no nat-control

Syntax Description

This command has no arguments or keywords.

Defaults

NAT control is disabled by default (**no nat-control** command). If you upgraded from an earlier version of software, however, NAT control might be enabled on your system because it was the default in some earlier versions.

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

NAT control requires that packets traversing from an inside interface to an outside interface match a NAT rule; for any host on the inside network to access a host on the outside network, you must configure NAT to translate the inside host address.

Interfaces at the same security level are not required to use NAT to communicate. However, if you configure dynamic NAT or PAT on a same security interface with NAT control enabled, then all traffic from the interface to a same security interface or an outside interface must match a NAT rule.

Similarly, if you enable outside dynamic NAT or PAT with NAT control, then all outside traffic must match a NAT rule when it accesses an inside interface.

Static NAT with NAT control does not cause these restrictions.

By default, NAT control is disabled, so you do not need to perform NAT on any networks unless you choose to perform NAT.

Note In multiple context mode, the packet classifier relies on the NAT configuration in some cases to assign packets to contexts. If you do not perform NAT because NAT control is disabled, then the classifier might require changes in your network configuration.

If you want the added security of NAT control but do not want to translate inside addresses in some cases, you can apply a NAT exemption (**nat 0 access-list**) or identity NAT (**nat 0** or **static**) rule on those addresses.

When NAT control is disabled with the **no-nat control** command, and a NAT and a global command pair are configured for an interface, the real IP addresses cannot go out on other interfaces unless you define those destinations with the **nat 0 access-list** command.

For example, the following NAT is the that one you want performed when going to the outside network:

```
nat (inside) 1 0.0.0.0 0.0.0.0
global (outside) 1 209.165.201.2
```

The above configuration catches everything on the inside network, so if you do not want to translate inside addresses when they go to the DMZ, then you need to match that traffic for NAT exemption, as shown in the following example:

```
access-list EXEMPT extended permit ip any 192.168.1.0 255.255.255.0
access-list EXEMPT remark This matches any traffic going to DMZ1
access-list EXEMPT extended permit ip any 10.1.1.0 255.255.255.0
access-list EXEMPT remark This matches any traffic going to DMZ1
nat (inside) 0 access-list EXEMPT
```

Alternately, you can perform NAT translation on all interfaces:

```
nat (inside) 1 0.0.0.0 0.0.0.0
global (outside) 1 209.165.201.2
global (dmz1) 1 192.168.1.230
global (dmz2) 1 10.1.1.230
```

Examples

The following example enables NAT control:

```
hostname(config)# nat-control
```

Related Commands

Command	Description
nat	Defines an address on one interface that is translated to a mapped address on another interface.
show running-config nat-control	Shows the NAT configuration requirement.
static	Translates a real address to a mapped address.

nat-rewrite

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

nat-rewrite

no nat-rewrite

Syntax Description

This command has no arguments or keywords.

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 security appliance 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 security appliance 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 security appliance 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 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 security appliance 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 security appliance 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 security appliance 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. The **neighbor** command is used to advertise OSPF routes over VPN tunnels.

neighbor *ip_address* [**interface** *name*]

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

Syntax Description

interface *name*	(Optional) 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
7.0(1)	This command was introduced.

Usage Guidelines

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.

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-policy	•	—	•	—	—

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 security appliance. PAT does not apply. Therefore, devices behind the security appliance 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.

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 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
Preexisting	This command was preexisting.

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 security appliance.

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 network object to a network object group, use the **network-object** command in network configuration mode. To remove network objects, use the **no** form of this command.

network-object host *host_addr* | *host_name*

no network-object host *host_addr* | *host_name*

network-object *net_addr netmask*

no network-object *net_addr netmask*

Syntax Description

host_addr	Host IP address (if the host name is not already defined using the **name** command).
host_name	Host name (if the host name is defined using the **name** command.
net_addr	Network address; used with *netmask* to define a subnet object.
netmask	Netmask; used with *net_addr* to define a subnet object.

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
Network configuration	•	•	•	•	—

Command History

Release	Modification
Preexisting	This command was preexisting.

Usage Guidelines

The **network-object** command is used with the **object-group** command to define a host or a subnet object in network configuration mode.

Examples

The following example shows how to use the **network-object** command in network configuration mode to create a new network object group:

```
hostname(config)# object-group network sjj_eng_ftp_servers
hostname(config-network)# network-object host sjj.eng.ftp
hostname(config-network)# network-object host 172.16.56.195
hostname(config-network)# network-object 192.168.1.0 255.255.255.224
hostname(config-network)# group-object sjc_eng_ftp_servers
hostname(config-network)# quit
```

```
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-group	Defines object groups to optimize your configuration.
port-object	Adds a port object to a service object group.
show running-config object-group	Displays the current object groups.

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 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	Specify 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	•	•	•	•	—

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 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
Preexisting	This command was preexisting.

Usage Guidelines

If you enable authentication, the security appliance only communicates with an NTP server if it uses the correct trusted key in the packets (see the **ntp trusted-key** command). The security appliance also uses an authentication key to synchronize with the NTP server (see the **ntp authentication-key** command).

Examples

The following example configures the security appliance 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 security appliance to use in packets for authentication with an NTP server.

Command	Description
show ntp associations	Shows the NTP servers with which the security appliance 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** *key*]

Syntax Description

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.
key	Sets the key value as a string up to 32 characters in length.

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
Preexisting	This command was preexisting.

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 security appliance to use in packets for authentication with an NTP server.
show ntp associations	Shows the NTP servers with which the security appliance 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 security appliance, use the **ntp server** command in global configuration mode. To remove the server, use the **no** form of this command. You can identify multiple servers; the security appliance uses the most accurate server. In multiple context mode, set the NTP server in the system configuration only.

Note When using an NTP server, only NTP is supported; SNTP is not supported.

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 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 security appliance uses the more accurate one. For example, the security appliance 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.

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 security appliance to use in packets for authentication with an NTP server.
show ntp associations	Shows the NTP servers with which the security appliance 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
Preexisting	This command was preexisting.

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 security appliance 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.

object-group

To define object groups that you can use to optimize your configuration, use the **object-group** command in global configuration mode. Use the **no** form of this command to remove object groups from the configuration. This command supports IPv4 and IPv6 addresses.

object-group {**protocol** | **network** | **icmp-type**} *obj_grp_id*

no object-group {**protocol** | **network** | **icmp-type**} *obj_grp_id*

object-group service *obj_grp_id* {**tcp** | **udp** | **tcp-udp**}

no object-group service *obj_grp_id* {**tcp** | **udp** | **tcp-udp**}

Syntax Description

icmp-type	Defines a group of ICMP types such as echo and echo-reply. After entering the main **object-group icmp-type** command, add ICMP objects to the ICMP type group with the **icmp-object** and the **group-object** commands.
network	Defines a group of hosts or subnet IP addresses. After entering the main **object-group network** command, add network objects to the network group with the **network-object** and the **group-object** commands.
obj_grp_id	Identifies the object group (one to 64 characters) and can be any combination of letters, digits, and the "_", "-", "." characters.
protocol	Defines a group of protocols such as TCP and UDP. After entering the main **object-group protocol** command, add protocol objects to the protocol group with the **protocol-object** and the **group-object** commands.
service	Defines a group of TCP/UDP port specifications such as "eq smtp" and "range 2000 2010." After entering the main **object-group service** command, add port objects to the service group with the **port-object** and the **group-object** commands.
tcp	Specifies that service group is used for TCP.
tcp-udp	Specifies that service group can be used for TCP and UDP.
udp	Specifies that service group is used for UDP.

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
Preexisting	This command was preexisting.

Usage Guidelines

Objects such as hosts, protocols, or services can be grouped, and then you can issue a single command using the group name to apply to every item in the group.

When you define a group with the **object-group** command and then use any security appliance command, the command applies to every item in that group. This feature can significantly reduce your configuration size.

Once you define an object group, you must use the **object-group** keyword before the group name in all applicable security appliance commands as follows:

```
hostname# show running-config object-group group_name
```

where *group_name* is the name of the group.

This example shows the use of an object group once it is defined:

```
hostname(config)# access-list access_list_name permit tcp any object-group group_name
```

In addition, you can group **access list** command arguments:

Individual Arguments	Object Group Replacement
protocol	**object-group** *protocol*
host and subnet	**object-group** *network*
service	**object-group** *service*
icmp_type	**object-group** *icmp_type*

You can group commands hierarchically; an object group can be a member of another object group.

To use object groups, you must do the following:

- Use the **object-group** keyword before the object group name in all commands as follows:

  ```
  hostname(config)# access-list acl permit tcp object-group remotes object-group locals
  object-group eng_svc
  ```

  where *remotes* and *locals* are sample object group names.
- The object group must be nonempty.
- You cannot remove or empty an object group if it is currently being used in a command.

After you enter a main **object-group** command, the command mode changes to its corresponding mode. The object group is defined in the new mode. The active mode is indicated in the command prompt format. For example, the prompt in the configuration terminal mode appears as follows:

```
hostname(config)#
```

where *hostname* is the name of the security appliance.

However, when you enter the **object-group** command, the prompt appears as follows:

```
hostname(config-type)#
```

where *hostname* is the name of the security appliance, and *type is the object-group type.*

Use the **exit**, **quit**, or any valid config-mode commands such as **access-list** to close an **object-group** mode and exit the **object-group** main command.

The **show running-config object-group** command displays all defined object groups by their *grp_id* when the **show running-config object-group** *grp_id* command is entered, and by their group type when you enter the **show running-config object-group** *grp_type* command. When you enter the **show running-config object-group** command without an argument, all defined object groups are shown.

Use the **clear configure object-group** command to remove a group of previously defined **object-group** commands. Without an argument, the **clear configure object-group** command lets you to remove all defined object groups that are not being used in a command. The *grp_type* argument removes all defined object groups that are not being used in a command for that group type only.

You can use all other security appliance commands in an object-group mode, including the**show running-config** and **clear configure** commands.

Commands within the object-group mode appear indented when displayed or saved by the **show running-config object-group**, **write**, or **config** commands.

Commands within the object-group mode have the same command privilege level as the main command.

When you use more than one object group in an **access-list** command, the elements of all object groups that are used in the command are linked together, starting with the elements of the first group with the elements of the second group, then the elements of the first and second groups together with the elements of the third group, and so on.

The starting position of the description text is the character right after the white space (a blank or a tab) following the **description** keyword.

Examples

The following example shows how to use the **object-group icmp-type** mode to create a new icmp-type object group:

```
hostname(config)# object-group icmp-type icmp-allowed
hostname(config-icmp-type)# icmp-object echo
hostname(config-icmp-type)# icmp-object time-exceeded
hostname(config-icmp-type)# exit
```

The following example shows how to use the **object-group network** command to create a new network object group:

```
hostname(config)# object-group network sjc_eng_ftp_servers
hostname(config-network)# network-object host sjc.eng.ftp.servcers
hostname(config-network)# network-object host 172.23.56.194
hostname(config-network)# network-object 192.1.1.0 255.255.255.224
hostname(config-network)# exit
```

The following example shows how to use the **object-group network** command to create a new network object group and map it to an existing object-group:

```
hostname(config)# object-group network sjc_ftp_servers
hostname(config-network)# network-object host sjc.ftp.servers
hostname(config-network)# network-object host 172.23.56.195
hostname(config-network)# network-object 193.1.1.0 255.255.255.224
hostname(config-network)# group-object sjc_eng_ftp_servers
hostname(config-network)# exit
```

The following example shows how to use the **object-group protocol** mode to create a new protocol object group:

```
hostname(config)# object-group protocol proto_grp_1
hostname(config-protocol)# protocol-object udp
```

```
hostname(config-protocol)# protocol-object ipsec
hostname(config-protocol)# exit

hostname(config)# object-group protocol proto_grp_2
hostname(config-protocol)# protocol-object tcp
hostname(config-protocol)# group-object proto_grp_1
hostname(config-protocol)# exit
```

The following example shows how to use the **object-group service** mode to create a new port (service) object group:

```
hostname(config)# object-group service eng_service tcp
hostname(config-service)# group-object eng_www_service
hostname(config-service)# port-object eq ftp
hostname(config-service)# port-object range 2000 2005
hostname(config-service)# exit
```

The following example shows how to add and remove a text description to an object group:

```
hostname(config)# object-group protocol protos1
hostname(config-protocol)# description This group of protocols is for our internal network

hostname(config-protocol)# show running-config object-group id protos1
object-group protocol protos1
description: This group of protocols is for our internal network

hostname(config-protocol)# no description
hostname(config-protocol)# show running-config object-group id protos1
object-group protocol protos1
```

The following example shows how to use the **group-object** mode to create a new object group that consists of previously defined objects:

```
hostname(config)# object-group network host_grp_1
hostname(config-network)# network-object host 192.168.1.1
hostname(config-network)# network-object host 192.168.1.2
hostname(config-network)# exit

hostname(config)# object-group network host_grp_2
hostname(config-network)# network-object host 172.23.56.1
hostname(config-network)# network-object host 172.23.56.2
hostname(config-network)# exit

hostname(config)# object-group network all_hosts
hostname(config-network)# group-object host_grp_1
hostname(config-network)# group-object host_grp_2
hostname(config-network)# exit

hostname(config)# access-list grp_1 permit tcp object-group host_grp_1 any eq ftp
hostname(config)#access-list grp_2 permit tcp object-group host_grp_2 any eq smtp
hostname(config)#access-list all permit tcp object-group all_hosts any eq www
```

Without the **group-object** command, you need to define the *all_hosts* group to include all the IP addresses that have already been defined in *host_grp_1* and *host_grp_2*. With the **group-object** command, the duplicated definitions of the hosts are eliminated.

The following examples show how to use object groups to simplify the access list configuration:

```
hostname(config)# object-group network remote
hostname(config-network)# network-object host kqk.suu.dri.ixx
hostname(config-network)# network-object host kqk.suu.pyl.gnl

hostname(config)# object-group network locals
hostname(config-network)# network-object host 209.165.200.225
```

```
hostname(config-network)# network-object host 209.165.200.230
hostname(config-network)# network-object host 209.165.200.235
hostname(config-network)# network-object host 209.165.200.240

hostname(config)# object-group service eng_svc tcp
hostname(config-service)# port-object eq www
hostname(config-service)# port-object eq smtp
hostname(config-service)# port-object range 25000 25100
```

This grouping enables the access list to be configured in 1 line instead of 24 lines, which would be needed if no grouping is used. Instead, with the grouping, the access list configuration is as follows:

```
hostname(config)# access-list acl permit tcp object-group remote object-group locals
object-group eng_svc
```

The following example shows how to use the **service-object** subcommand, which is useful for grouping TCP and UDP services:

```
hostname(config)# object-group network remote
hostname(config-network)# network-object host kqk.suu.dri.ixx
hostname(config-network)# network-object host kqk.suu.pyl.gnl

hostname(config)# object-group network locals
hostname(config-network)# network-object host host 209.165.200.225
hostname(config-network)# network-object host host 209.165.200.230
hostname(config-network)# network-object host host 209.165.200.235
hostname(config-network)# network-object host host 209.165.200.240

hostname(config)# object-group service usr_svc
hostname(config-service)# service-object tcp eq www
hostname(config-service)# service-object tcp eq https
hostname(config-service)# service-object tcp eq pop3
hostname(config-service)# service-object udp eq ntp
hostname(config-service)# service-object udp eq domain

hostname(config)# access-list acl permit object-group usr_svc object-group locals
object-group remote
```

Note The **show running-config object-group** and **write** commands allow you to display the access list as configured with the object group names. The **show access-list** command displays the access list entries that are expanded out into individual entries without their object groupings.

Related Commands

Command	Description
clear configure object-group	Removes all the **object group** commands from the configuration.
group-object	Adds network object groups.
network-object	Adds a network object to a network object group.
port-object	Adds a port object to a service object group.
show running-config object-group	Displays the current object groups.

ocsp disable-nonce

By default, OCSP requests include a nonce extension, which cryptographically binds requests with responses to avoid replay attacks. However, some OCSP servers use pre-generated responses that do not contain this matching nonce extension. To use OCSP with these servers, you must disable the nonce extension.

To disable the nonce extension, use the **ocsp disable-nonce** command in crypto ca trustpoint mode. To re-enable the nonce extension, use the **no** version of this command.

ocsp disable-nonce

no ocsp disable-nonce

Syntax Description

This command has no keywords or arguments.

Defaults

By default, OCSP requests include a nonce extension.

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
crypto ca trustpoint mode	•	•	•	•	•

Command History

Release	Modification
7.2(1)	This command was introduced.

Usage Guidelines

When you use this command, the OCSP request does not include the OCSP nonce extension, and the security appliance does not check it.

Examples

The following example shows how to disable the nonce extension for a trustpoint called newtrust.

```
hostname(config)# crypto ca trustpoint newtrust
hostname(config-ca-trustpoint)# ocsp disable-nonce
hostname(config-ca-trustpoint)#
```

Related Commands

Command	Description
crypto ca trustpoint	Enters crypto ca trustpoint mode. Use this command in global configuration mode.
match certificate	Configures an OCSP override rule,
ocsp url	Specifies the OCSP server to use to check all certificates associated with a trustpoint.
revocation-check	Specifies the method(s) to use for revocation checking, and the order in which to try them.

ocsp url

To configure an OCSP server for the security appliance to use to check all certificates associated with a trustpoint rather than the server specified in the AIA extension of the client certificate, use the **ocsp url** command in crypto ca trustpoint mode. To remove the server from the configuration, use the **no** version of the command.

ocsp url *URL*

no ocsp url

Syntax Description

This command has no keywords or arguments.

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
crypto ca trustpoint mode	•	•	•	•	•

Command History

Release	Modification
7.2(1)	This command was introduced.

Usage Guidelines

The security appliance supports only HTTP URLs, and you can specify only one URL per trustpoint.

The security appliance provides three ways to define an OCSP server URL, and it attempts to use OCSP servers according to how you define them, in the following order:

- An OCSP server you set using **match certificate** command.
- An OCSP server you set using the **ocsp url** command.
- The OCSP server in the AIA field of the client certificate.

If you do not configure an OCSP URL via the **match certificate** command or the **ocsp url** command, the security appliance uses the OCSP server in the AIA extension of the client certificate. If the certificate does not have an AIA extension, revocation status checking fails.

Examples

The following example shows how to configure an OCSP server with the URL http://10.1.124.22.

```
hostname(config)# crypto ca trustpoint newtrust
hostname(config-ca-trustpoint)# ocsp url http://10.1.124.22
hostname(config-ca-trustpoint)#
```

Related Commands

Command	Description
crypto ca trustpoint	Enters crypto ca trustpoint mode. Use this command in global configuration mode.
match certificate	Configures an OCSP override rule,
ocsp disable-nonce	Disables the nonce extension of the OCSP request.
revocation-check	Specifies the method(s) to use for revocation checking, and the order in which to try them.

ospf authentication

To enable the use of OSPF authentication, use the **ospf authentication** command in interface configuration mode. To restore the default authentication stance, use the **no** form of this command.

ospf authentication [**message-digest** | **null**]

no ospf authentication

Syntax Description

message-digest	(Optional) Specifies to use OSPF message digest authentication.
null	(Optional) Specifies to not use OSPF authentication.

Defaults

By default, OSPF authentication is not enabled.

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
Preexisting	This command was preexisting.

Usage Guidelines

Before using the **ospf authentication** command, configure a password for the interface using the **ospf authentication-key** command. If you use the **message-digest** keyword, configure the message-digest key for the interface with the **ospf message-digest-key** command.

For backward compatibility, authentication type for an area is still supported. If the authentication type is not specified for an interface, the authentication type for the area will be used (the area default is null authentication).

When this command is used without any options, simple password authentication is enabled.

Examples

The following example shows how to enable simple password authentication for OSPF on the selected interface:

```
hostname(config-if)# ospf authentication
hostname(config-if)#
```

Related Commands

Command	Description
ospf authentication-key	Specifies the password used by neighboring routing devices.
ospf message-digest-key	Enables MD5 authentication and specifies the MD5 key.

ospf authentication-key

To specify the password used by neighboring routing devices, use the **ospf authentication-key** command in interface configuration mode. To remove the password, use the **no** form of this command.

ospf authentication-key *password*

no ospf authentication-key

Syntax Description<

password	Assigns an OSPF authentication password for use by neighboring routing devices. The password must be less than 9 characters. You can include blank space between two characters. Spaces at the beginning or end of the password are ignored.

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
Preexisting	This command was preexisting.

Usage Guidelines

The password created by this command is used as a key that is inserted directly into the OSPF header when routing protocol packets are originated. A separate password can be assigned to each network on a per-interface basis. All neighboring routers on the same network must have the same password to be able to exchange OSPF information.

ExamplesNote

The following example shows how to specify a password for OSPF authentication:

```
hostname(config-if)# ospf authentication-key ThisMyPW
```

Related Commands

Command	Description
area authentication	Enables OSPF authentication for the specified area.
ospf authentication	Enables the use of OSPF authentication.

ospf cost

To specify the cost of sending a packet through the interface, use the **ospf cost** command in interface configuration mode. To reset the interface cost to the default value, use the **no** form of this command.

ospf cost *interface_cost*

no ospf cost

Syntax Description

interface_cost	The cost (a link-state metric) of sending a packet through an interface. This is an unsigned integer value from 0 to 65535. 0 represents a network that is directly connected to the interface, and the higher the interface bandwidth, the lower the associated cost to send packets across that interface. In other words, a large cost value represents a low bandwidth interface and a small cost value represents a high bandwidth interface. The OSPF interface default cost on the security appliance is 10. This default differs from Cisco IOS software, where the default cost is 1 for fast Ethernet and Gigabit Ethernet and 10 for 10BaseT. This is important to take into account if you are using ECMP in your network.

Defaults

The default *interface_cost* is 10.

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
Preexisting	This command was preexisting.

Usage Guidelines

The **ospf cost** command lets you explicitly specify the cost of sending a packet on an interface. The *interface_cost* parameter is an unsigned integer value from 0 to 65535.

The **no ospf cost** command allows you to reset the path cost to the default value.

Examples

The following example show how to specify the cost of sending a packet on the selected interface:

```
hostname(config-if)# ospf cost 4
```

Related Commands

Command	Description
show running-config interface	Displays the configuration of the specified interface.

ospf database-filter

To filter out all outgoing LSAs to an OSPF interface during synchronization and flooding, use the **ospf database-filter** command in interface configuration mode. To restore the LSAs, use the **no** form of this command.

ospf database-filter all out

no ospf database-filter all out

Syntax Description

all out	Filters all outgoing LSAs to an OSPF interface.

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
Preexisting	This command was preexisting.

Usage Guidelines

The **ospf database-filter** command filters outgoing LSAs to an OSPF interface. The **no ospf database-filter all out** command restores the forwarding of LSAs to the interface.

Examples

The following example shows how to use the **ospf database-filter** command to filter outgoing LSAs:

```
hostname(config-if)# ospf database-filter all out
```

Related Commands

Command	Description
show interface	Displays interface status information.

ospf dead-interval

To specify the interval before neighbors declare a router down, use the **ospf dead-interval** command in interface configuration mode. To restore the default value, use the **no** form of this command.

ospf dead-interval *seconds*

no ospf dead-interval

Syntax Description

seconds	The length of time during which no hello packets are seen. The default for *seconds* is four times the interval set by the **ospf hello-interval** command (which ranges from 1 to 65535).

Defaults

The default value for *seconds* is four times the interval set by the **ospf hello-interval** command.

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
Preexisting	This command was preexisting.

Usage Guidelines

The **ospf dead-interval** command lets you set the dead interval before neighbors to declare the router down (the length of time during which no hello packets are seen). The *seconds* argument specifies the dead interval and must be the same for all nodes on the network. The default for *seconds* is four times the interval set by the **ospf hello-interval** command from 1 to 65535.

The **no ospf dead-interval** command lets restores the default interval value.

Examples

The following example sets the OSPF dead interval to 1 minute:

```
hostname(config-if)# ospf dead-interval 60
```

Related Commands

Command	Description
ospf hello-interval	Specifies the interval between hello packets sent on an interface.
show ospf interface	Displays OSPF-related interface information.

ospf hello-interval

To specify the interval between hello packets sent on an interface, use the **ospf hello-interval** command in interface configuration mode. To return the hello interval to the default value, use the **no** form of this command.

ospf hello-interval *seconds*

no ospf hello-interval

Syntax Description

seconds	Specifies the interval between hello packets that are sent on the interface; valid values are from 1 to 65535 seconds.

Defaults

The default value for **hello-interval** *seconds* is 10 seconds.

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
Preexisting	This command was preexisting.

Usage Guidelines

This value is advertised in the hello packets. The smaller the hello interval, the faster topological changes will be detected, but more routing traffic will ensue. This value must be the same for all routers and access servers on a specific network.

Examples

The following example sets the OSPF hello interval to 5 seconds:

```
hostname(config-if)# ospf hello-interval 5
```

Related Commands

Command	Description
ospf dead-interval	Specifies the interval before neighbors declare a router down.
show ospf interface	Displays OSPF-related interface information.

ospf message-digest-key

To enable OSPF MD5 authentication, use the **ospf message-digest-key** command in interface configuration mode. To remove an MD5 key, use the **no** form of this command.

ospf message-digest-key *key-id* **md5** *key*

no ospf message-digest-key

Syntax Description

key-id	Enables MD5 authentication and specifies the numerical authentication key ID number; valid values are from 1 to 255.
md5 *key*	Alphanumeric password of up to 16 bytes. You can include spaces between key characters. Spaces at the beginning or end of the key are ignored. MD5 authentication verifies the integrity of the communication, authenticates the origin, and checks for timeliness.

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
Preexisting	This command was preexisting.

Usage Guidelines

The **ospf message-digest-key** command lets you enable MD5 authentication. The **no** form of the command let you remove an old MD5 key. *key_id* is a numerical identifier from 1 to 255 for the authentication key. *key* is an alphanumeric password of up to 16 bytes. MD5 verifies the integrity of the communication, authenticates the origin, and checks for timeliness.

Examples

The following example shows how to specify an MD5 key for OSPF authentication:

```
hostname(config-if)# ospf message-digest-key 3 md5 ThisIsMyMd5Key
```

Related Commands

Command	Description
area authentication	Enables OSPF area authentication.
ospf authentication	Enables the use of OSPF authentication.

ospf mtu-ignore

To disable OSPF maximum transmission unit (MTU) mismatch detection on receiving database packets, use the **ospf mtu-ignore** command in interface configuration mode. To restore MTU mismatch detection, use the **no** form of this command.

ospf mtu-ignore

no ospf mtu-ignore

Syntax Description

This command has no arguments or keywords.

Defaults

By default, **ospf mtu-ignore** is enabled.

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
Preexisting	This command was preexisting.

Usage Guidelines

OSPF checks whether neighbors are using the same MTU on a common interface. This check is performed when neighbors exchange Database Descriptor (DBD) packets. If the receiving MTU in the DBD packet is higher than the IP MTU configured on the incoming interface, OSPF adjacency will not be established.The **ospf mtu-ignore** command disables OSPF MTU mismatch detection on receiving DBD packets. It is enabled by default.

Examples

The following example shows how to disable the **ospf mtu-ignore** command:

```
hostname(config-if)# ospf mtu-ignore
```

Related Commands

Command	Description
show interface	Displays interface status information.

ospf network point-to-point non-broadcast

To configure the OSPF interface as a point-to-point, non-broadcast network, use the **ospf network point-to-point non-broadcast** command in interface configuration mode. To remove this command from the configuration, use the **no** form of this command. The **ospf network point-to-point non-broadcast** command lets you to transmit OSPF routes over VPN tunnels.

ospf network point-to-point non-broadcast

no ospf network point-to-point non-broadcast

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
Interface configuration	•	—	•	—	—

Command History

Release	Modification
7.0(1)	This command was introduced.

Usage Guidelines

When the interface is specified as point-to-point, the OSPF neighbors have to be manually configured; dynamic discovery is not possible. To manually configure OSPF neighbors, use the **neighbor** command in router configuration mode.

When an interface is configured as point-to-point, the following restrictions apply:

- You can define only one neighbor for the interface.
- You need to define a static route pointing to the crypto endpoint.
- The interface cannot form adjacencies unless neighbors are configured explicitly.
- If OSPF over the tunnel is running on the interface, regular OSPF with an upstream router cannot be run on the same interface.
- You should bind the crypto-map to the interface before specifying the OSPF neighbor to ensure that the OSPF updates are passed through the VPN tunnel. If you bind the crypto-map to the interface after specifying the OSPF neighbor, use the **clear local-host all** command to clear OSPF connections so the OSPF adjacencies can be established over the VPN tunnel.

Examples

The following example shows how to configure the selected interface as a point-to-point, non-broadcast interface:

```
hostname(config-if)# ospf network point-to-point non-broadcast
hostname(config-if)#
```

Related Commands

Command	Description
neighbor	Specifies manually configured OSPF neighbors.
show interface	Displays interface status information.

ospf priority

To change the OSPF router priority, use the **ospf priority** command in interface configuration mode. To restore the default priority, use the **no** form of this command.

ospf priority *number*

no ospf priority [*number*]

Syntax Description

number	Specifies the priority of the router; valid values are from 0 to 255.

Defaults

The default value for *number* 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
Interface configuration	•	—	•	—	—

Command History

Release	Modification
Preexisting	This command was preexisting.

Usage Guidelines

When two routers attached to a network both attempt to become the designated router, the one with the higher router priority takes precedence. If there is a tie, the router with the higher router ID takes precedence. A router with a router priority set to zero is ineligible to become the designated router or backup designated router. Router priority is configured only for interfaces to multiaccess networks (in other words, not to point-to-point networks).

Examples

The following example shows how to change the OSPF priority on the selected interface:

```
hostname(config-if)# ospf priority 4
hostname(config-if)#
```

Related Commands

Command	Description
show ospf interface	Displays OSPF-related interface information.

ospf retransmit-interval

To specify the time between LSA retransmissions for adjacencies belonging to the interface, use the **ospf retransmit-interval** command in interface configuration mode. To restore the default value, use the **no** form of this command.

ospf retransmit-interval *seconds*

no ospf retransmit-interval [*seconds*]

Syntax Description

seconds	Specifies the time between LSA retransmissions for adjacent routers belonging to the interface; valid values are from 1 to 65535 seconds.

Defaults

The default value of **retransmit-interval** *seconds* is 5 seconds.

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
Preexisting	This command was preexisting.

Usage Guidelines

When a router sends an LSA to its neighbor, it keeps the LSA until it receives the acknowledgment message. If the router receives no acknowledgment, it will re-send the LSA.

The setting of this parameter should be conservative, or needless retransmission will result. The value should be larger for serial lines and virtual links.

Examples

The following example shows how to change the retransmit interval for LSAs:

```
hostname(config-if)# ospf retransmit-interval 15
hostname(config-if)#
```

Related Commands

Command	Description
show ospf interface	Displays OSPF-related interface information.

ospf transmit-delay

To set the estimated time required to send a link-state update packet on the interface, use the **ospf transmit-delay** command in interface configuration mode. To restore the default value, use the **no** form of this command.

ospf transmit-delay *seconds*

no ospf transmit-delay [*seconds*]

Syntax Description

seconds	Sets the estimated time required to send a link-state update packet on the interface. The default value is 1 second with a range from 1 to 65535 seconds.

Defaults

The default value of *seconds* is 1 second.

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
Preexisting	This command was preexisting.

Usage Guidelines

LSAs in the update packet must have their ages incremented by the amount specified in the *seconds* argument before transmission. The value assigned should take into account the transmission and propagation delays for the interface.

If the delay is not added before transmission over a link, the time in which the LSA propagates over the link is not considered. This setting has more significance on very low-speed links.

Examples

The following example sets the transmit delay to 3 seconds for the selected interface:

```
hostname(config-if)# ospf restransmit-delay 3
hostname(config-if)#
```

Related Commands

Command	Description
show ospf interface	Displays OSPF-related interface information.

outstanding

To limit the number of unauthenticated e-mail proxy sessions, use the **outstanding** command in the applicable e-mail proxy mode. To remove the attribute from the configuration, use the **no** version of this command, which permits an unlimited number of unauthenticated sessions. Use this command to limit DOS attacks on the e-mail ports.

E-mail proxy connections have three states:

1. A new e-mail connection enters the "unauthenticated" state.
2. When the connection presents a username, it enters the "authenticating" state.
3. When the security appliance authenticates the connection, it enters the "authenticated" state.

If the number of connections in the unauthenticated state exceeds the configured limit, the security appliance terminates the oldest unauthenticated connection, preventing overload. It does not terminate authenticated connections.

outstanding {*number*}

no outstanding

Syntax Description

number	The number of unauthenticated sessions permitted. The range is from 1 to 1000.

Defaults

The default is 20.

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.

Examples

The following example shows how to set a limit of 12 unauthenticated sessions for POP3S e-mail proxy.

```
hostname(config)# pop3s
hostname(config-pop3s)# outstanding 12
```

override-account-disable

To override an account-disabled indication from a AAA server, use the **override-account-disable** command in tunnel-group general-attributes configuration mode. To disable an override, use the **no** form of this command.

override-account-disable

no override-account-disable

Syntax Description

This command has no arguments or keywords.

Defaults

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
7.1.1	This command was introduced.

Usage Guidelines

This command is valid for servers, such as RADIUS with NT LDAP, and Kerberos, that return an "account-disabled" indication.

You can configure this attribute for IPSec RA and WebVPN tunnel-groups.

Examples

The following example allows overriding the "account-disabled" indicator from the AAA server for the WebVPN tunnel group "testgroup":

```
hostname(config)# tunnel-group testgroup type webvpn
hostname(config)# tunnel-group testgroup general-attributes
hostname(config-tunnel-general)# override-account-disable
hostname(config-tunnel-general)#
```

The following example allows overriding the "account-disabled" indicator from the AAA server for the IPSec remote access tunnel group "QAgroup":

```
hostname(config)# tunnel-group QAgroup type ipsec-ra
hostname(config)# tunnel-group QAgroup general-attributes
hostname(config-tunnel-general)# override-account-disable
hostname(config-tunnel-general)#
```

Related Commands

Command	Description
clear configure tunnel-group	Clears the tunnel-group database or the configuration for a particular tunnel group.
tunnel-group general-attributes	Configures the tunnel-group general-attributes values.