object network through override-svc-download Commands

object network

To configure a named network object, use the **object network** command in global configuration mode. Use the **no** form of this command to remove the object from the configuration.

object network *name* [**rename** *new_obj_name*]

no object network *name*

Syntax Description

name	Specifies the name of the network object. The name can be from 1 to 64 characters in length, consisting of letters, numbers, and the following special characters: underscore, hyphen, comma, forward slash, and period. Objects and object groups share the same name space.
rename *new_obj_name*	(Optional) Renames the object to the new object name.

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
8.3(1)	This command was introduced.
8.4(2)	Support for FQDNs was introduced. See the **fqdn** command.

Usage Guidelines

The network object can contain a host, a network, a range IP addresses (IPv4 or IPv6), or an FQDN.

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

If you configure an existing network object with a different IP address, the new configuration will replace the existing configuration.

You cannot remove an object or make an object empty if it is used in a command.

Examples

The following example shows how to create a network object:

```
hostname (config)# object network OBJECT1
hostname (config-network-object)# host 10.1.1.1
```

Related Commands

Command	Description
clear configure object	Clears all objects created.
description	Adds a description to the network object.
fqdn	Specifies a fully-qualified domain name network object.
host	Specifies a host network object.
nat	Enables NAT for the network object.
object-group network	Creates a network object group.
range	Specifies a range of addresses for the network object.
show running-config object network	Shows the network object configuration.
subnet	Specifies a subnet network object.

object service

To configure a service object that is automatically reflected in all configurations in which the object is used, use the **object service** command in global configuration mode. Use the **no** form of this command to remove the object.

object service *name* [**rename** *new_obj_name*]

no object service *object name* [**rename** *new_obj_name*]

Syntax Description

name	Specifies the name of the existing service object. The name can be from 1 to 64 characters in length, consisting of letters, numbers, and the following special characters: underscore, hyphen, comma, and period. The object name must start with a letter.
rename *new_obj_name*	(Optional) Renames the object to the new object name.

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
8.3(1)	This command was introduced.

Usage Guidelines

The service object can contain a protocol, ICMP, ICMPv6, TCP or UDP port or port ranges.

If you configure an existing service object with a different protocol and port (or ports), the new configuration replaces the existing protocol and port (or ports) with the new ones.

Examples

The following example shows how to create a service object:

```
hostname(config)# object service SERVOBJECT1
hostname(config-service-object)# service tcp source eq www destination eq ssh
```

Related Commands

Command	Description
clear configure object	Clears all objects created.
service	Configures the protocol and port for the service object.

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** | **security** | **service** [**tcp** | **udp** | **tcp-udp**] | **user**} *grp_name*

Syntax Description

grp_name	Identifies the object group (one to 64 characters) and can be any combination of letters, digits, and the "_", "-", "." characters.
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. You can create a group with a mix of IPv4 and IPv6 addresses. **Note** You cannot use a mixed object group for NAT.
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.
security	Creates a security group object for use with Cisco TrustSec.
service	Defines a group of ports for a protocol (TCP, UDP, or TCP-UDP), or a group of services (a mix of protocols and ports). To define a group of ports, use the **tcp**, **udp**, or **tcp-udp** keywords. After entering the main **object-group service** *protocol* command, add port objects to the service group with the **port-object** and the **group-object** commands. To define a mixed group of services, do not specify the protocol type for the object-group. After entering the main **object-group service** command, add service objects to the service group with the **service-object** and the **group-object** commands.
tcp	(Optional) Specifies that the service group is used for TCP.
tcp-udp	(Optional) Specifies that the service group is used for ports in both TCP and UDP.
udp	(Optional) Specifies that the service group is used for UDP.
user	Defines object groups that you can use to control access with the Identity Firewall.

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)	We introduced this command.
8.4(2)	We added support for the **user** keyword to support Identity Firewall.
9.0(1)	You can now create network object groups that can support a mix of both IPv4 and IPv6 addresses. We added support for the **security** keyword to support Cisco TrustSec.

Usage Guidelines

Objects such as hosts, protocols, or services can be grouped, and then you can use the object group in features such as ACLs (**access-list**) and NAT (**nat**). This example shows the use of a network object group in an ACL:

```
hostname(config)# access-list access_list_name permit tcp any object-group NWgroup1
```

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

You cannot remove or empty an object group if it is currently being used in a command.

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-object-group)# icmp-object echo
hostname(config-icmp-object-group)# icmp-object time-exceeded
hostname(config-icmp-object-group)# 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-object-group)# network-object host sjc.eng.ftp.servcers
hostname(config-network-object-group)# network-object host 172.23.56.194
hostname(config-network-object-group)# network-object 192.1.1.0 255.255.255.224
hostname(config-network-object-group)# 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-object-group)# network-object host sjc.ftp.servers
hostname(config-network-object-group)# network-object host 172.23.56.195
hostname(config-network-object-group)# network-object 193.1.1.0 255.255.255.224
hostname(config-network-object-group)# group-object sjc_eng_ftp_servers
hostname(config-network-object-group)# 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-object-group)# protocol-object udp
hostname(config-protocol-object-group)# protocol-object ipsec
hostname(config-protocol-object-group)# exit

hostname(config)# object-group protocol proto_grp_2
hostname(config-protocol-object-group)# protocol-object tcp
hostname(config-protocol-object-group)# group-object proto_grp_1
hostname(config-protocol-object-group)# 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-object-group)# group-object eng_www_service
hostname(config-service-object-group)# port-object eq ftp
hostname(config-service-object-group)# port-object range 2000 2005
hostname(config-service-object-group)# 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-object-group)# description This group of protocols is for our
internal network

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

hostname(config-protocol-object-group)# no description
hostname(config-protocol-object-group)# 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-object-group)# network-object host 192.168.1.1
hostname(config-network-object-group)# network-object host 192.168.1.2
hostname(config-network-object-group)# exit

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

hostname(config)# object-group network all_hosts
hostname(config-network-object-group)# group-object host_grp_1
hostname(config-network-object-group)# group-object host_grp_2
hostname(config-network-object-group)# 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-object-group)# network-object host kqk.suu.dri.ixx
hostname(config-network-object-group)# network-object host kqk.suu.pyl.gnl

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

hostname(config)# object-group service eng_svc tcp
hostname(config-service-object-group)# port-object eq www
hostname(config-service-object-group)# port-object eq smtp
hostname(config-service-object-group)# 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-object-group)# network-object host kqk.suu.dri.ixx
hostname(config-network-object-group)# network-object host kqk.suu.pyl.gnl

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

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

Command	Description
port-object	Adds a port object to a service object group.
show running-config object-group	Displays the current object groups.

object-group user

To create a user group object that support the Identity Firewall feature, use the **object-group user** command in global configuration mode. Use the **no** form of this command to disable the user group object.

object-group user *user_group_name*

[no] object-group user *user_group_name*

Syntax Description

user_group_name	Specifies the name for the user group. The group name can contain any character including [a-z], [A-Z], [0-9], [!@#$%^&()-_{}.]. If the group name contains a space, you must enclose the name in quotation marks.

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
8.4(2)	This command was introduced.

Usage Guidelines

The ASA sends an LDAP query to the Active Directory server for user groups globally defined in the Active Directory domain controller. The ASA imports these groups for the Identity Firewall feature. However, the ASA might have localized network resources that are not defined globally that require local user groups with localized security policies. Local user groups can contain nested groups and user groups that are imported from Active Directory. The ASA consolidates local and Active Directory groups. A user can belong to local user groups and user groups imported from Active Directory.

The ASA supports up to 256 user groups (including imported user groups and local user groups).

You active user group objects by including them within an access group, capture, or service policy.

Within a user group object, you can define the following object types:

- **User**—Adds a single user to the object-group user.

  The user can be either a LOCAL user or imported user. The *user_name* argument that you specify with the **user** keyword contains an ASCII user name and does not specify an IP address.

The name of an imported user must be the sAMAccountName, which is unique, rather than the common name (cn), which might not be unique. However, some Active Directory server administrators might require that the sAMAccountName and the cn be identical. In this case, the cn that the ASA displays in the output of the **show user-identity ad-group-member** command can be used for imported users defined by the user object.

- **User-group**—Adds an imported user group, which is defined by an external directory server, such as Microsoft Active Directory server, to the group-object user.

  The group name of the user-group must be the sAMAccountName, which is unique, rather than the cn, which might not be unique. However, some Active Directory server administrators might require that the sAMAccountName and the cn be identical. In this case, the cn that the ASA displays in the output of the **show user-identity ad-group-member** command can be used in the *user_group_name* argument specified with the **user-group** keyword.

  Note You can add *domain_nickname*\\user_group_name or *domain_nickname**user_ name* directly within a user group object without specifying them in the object first. If the *domain_nickname* is associated with a AAA server, the ASA imports the detailed nested user groups and the users defined in the external directory server, such as the Microsoft Active Directory server, to the ASA when the user object group is activated.

- **Group-object**—Adds a group defined locally on the ASA to the object-group user.

  Note When including an object-group within a object-group user object, the ASA does not expand the object-group in access groups even when you enable ACL optimization. The output of the **show object-group** command does not display the hit count, which is available only for regular network object-group when ACL optimization is enabled.

- **Description**—Adds a description for the object-group user.

Examples

The following example shows how to use the **object-group user** command to create user group objects for use with the Identity Firewall feature:

```
hostname(config)# object-group user sampleuser1-group
hostname(config-object-group user)# description group members of sampleuser1-group
hostname(config-object-group user)# user-group CSCO\\group.sampleusers-all
hostname(config-object-group user)# user CSCO\user2
hostname(config-object-group user)# exit
hostname(config)# object-group user sampleuser2-group
hostname(config-object-group user)# description group members of sampleuser2-group
hostname(config-object-group user)# group-object sampleuser1-group
hostname(config-object-group user)# user-group CSCO\\group.sampleusers-marketing
hostname(config-object-group user)# user CSCO\user3
```

Related Commands

Command	Description
description	Adds a description to the group created with the **object-group user** command.
group-object	Adds a locally defined object group to a user object group created with the **object-group user** command for use with the Identity Firewall feature.

Command	Description
user	Adds a user to the group created with the **object-group user** command.
user-group	Adds a user group imported from Microsoft Active Directory to the group created with the **object-group user** command.
user-identity enable	Creates the Cisco Identify Firewall instance.

object-group-search

To enable ACL optimization, use the **object-group-search** command in global configuration mode. Use the **no** form of this command to disable ACL optimization.

object-group-search access-control

no object-group-search access-control

Syntax Description

access-control	Searches for the access-control domain.

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
8.3(1)	This command was introduced.

Usage Guidelines

The **object-group-search** command optimizes all ACLs in the inbound direction.

When the **object-group-search** command is enabled, all of the old NP rules are removed from the soft-NP and reinserted with object-group IDs. When the command is disabled, all of the old rules are removed from the soft-NP and reinserted by expanding the object groups.

When the **object-group-search access-control** command is enabled on an ASA, with a significant number of features enabled, a large number of active connections and loaded with a large ACL, there will be a connection drop during the operation and a performance drop while establishing new connections.

Examples

The following example shows how to use the **object-group-search** command to enable ACL optimization:

```
hostname(config)# object-group-search access-control
```

The following is sample output from the **show access-list** command when **object-group-search** is not enabled:

```
hostname# show access-list KH-BLK-Tunnel
access-list KH-BLK-Tunnel; 9 elements
```

```
access-list KH-BLK-Tunnel line 1 extended permit ip object-group KH-LAN object-group
BLK-LAN 0x724c956b
   access-list KH-BLK-Tunnel line 1 extended permit ip 192.168.97.0 255.255.255.0
192.168.4.0 255.255.255.0 (hitcnt=10) 0x30fe29a6
   access-list KH-BLK-Tunnel line 1 extended permit ip 13.13.13.0 255.255.255.0
192.168.4.0 255.255.255.0 (hitcnt=4) 0xc6ef2338
   access-list KH-BLK-Tunnel line 1 extended permit ip 192.168.97.0 255.255.255.0
14.14.14.0 255.255.255.0 (hitcnt=2) 0xce8596ec
   access-list KH-BLK-Tunnel line 1 extended permit ip 13.13.13.0 255.255.255.0 14.14.14.0
255.255.255.0 (hitcnt=0) 0x9a2f1c4d
access-list KH-BLK-Tunnel line 2 extended permit ospf interface pppoe1 host 87.139.87.200
(hitcnt=0) 0xb62d5832
access-list KH-BLK-Tunnel line 3 extended permit ip interface pppoe1 any (hitcnt=0)
0xa2c9ed34
access-list KH-BLK-Tunnel line 4 extended permit ip host 1.1.1.1 any (hitcnt=0) 0xd06f7e6b
access-list KH-BLK-Tunnel line 5 extended deny ip 1.1.0.0 255.255.0.0 any (hitcnt=0)
0x9d979934
access-list KH-BLK-Tunnel line 6 extended permit ip 1.1.1.0 255.255.255.0 any (hitcnt=0)
0xa52a0761
```

The following is sample output from the **show access-list** command when **object-group-search** is enabled:

```
hostname# show access-list KH-BLK-Tunnel
access-list KH-BLK-Tunnel; 6 elements
access-list KH-BLK-Tunnel line 1 extended permit ip object-group KH-LAN(1) object-group
BLK-LAN(2)(hitcount=16) 0x724c956b
access-list KH-BLK-Tunnel line 2 extended permit ospf interface pppoe1 host 87.139.87.200
(hitcnt=0) 0xb62d5832
access-list KH-BLK-Tunnel line 3 extended permit ip interface pppoe1 any (hitcnt=0)
0xa2c9ed34
access-list KH-BLK-Tunnel line 4 extended permit ip host 1.1.1.1 any (hitcnt=0) 0xd06f7e6b
access-list KH-BLK-Tunnel line 5 extended deny ip 1.1.0.0 255.255.0.0 any (hitcnt=0)
0x9d979934
access-list KH-BLK-Tunnel line 6 extended permit ip 1.1.1.0 255.255.255.0 any (hitcnt=0)
0xa52a0761
```

Related Commands

Command	Description
clear config object-group search	Clears the object-group-search configuration.
show object-group	Shows the hit count if the object group is of the network object-group type.
show running-config object-group	Displays the current object groups.
show running-config object-group-search	Show the object-group-search configuration in the running configuration.

ocsp disable-nonce

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

ocsp disable-nonce

no ocsp disable-nonce

Syntax Description

This command has no arguments or keywords.

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 configuration	•	•	•	•	•

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 ASA does not check it. 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.

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 configuration mode. Use this command in global configuration mode.
match certificate	Configures an OCSP override rule.

Command	Description
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 ASA 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 configuration mode. To remove the server from the configuration, use the **no** form of this command.

ocsp url *URL*

no ocsp url

Syntax Description

URL	Specifies the HTTP URL for the OCSP server.

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 configuration	•	•	•	•	•

Command History

Release	Modification
7.2(1)	This command was introduced.

Usage Guidelines

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

The ASA 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 ASA 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 configuration 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.

onscreen-keyboard

To insert an onscreen keyboard into the logon pane or all panes with a login/password requirement, use the **onscreen-keyboard** command in webvpn mode. To remove a previously configured onscreen keyboard, use the **no** version of the command.

onscreen-keyboard {logon | all}

no onscreen-keyboard [logon | all]

Syntax Description

logon	Inserts the onscreen keyboard for the logon pane.
all	Inserts the onscreen keyboard for the logon pane, and for all other panes with a login/password requirement.

Defaults

No onscreen keyboard.

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
Webvpn configuration mode	•	—	•	—	—

Command History

Release	Modification
8.0(2)	This command was introduced.

Usage Guidelines

The onscreen keyboard lets you enter user credentials without keystrokes.

Examples

The following example shows how to enable the onscreen keyboard for the logon page:

```
hostname(config)# webvpn
hostname(config-webvpn)# onscreen-keyboard logon
hostname(config-webvpn)#
```

Related Commands

Command	Description
webvpn	Enters webvpn mode, which lets you configure attributes for clientless SSLVPN connections.

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

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 [**0** | **8**] *password*

no ospf authentication-key

Syntax Description<

0	Specifies an unencrypted password will follow
8	Specifies an encrypted password will follow.
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
7.0(1)	This command was introduced.
9.0(1)	Multiple context mode is supported.

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 8 yWIvi0qJAnGK5MRWQzrhIohkGP1wKb
```

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

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
7.0(1)	This command was introduced.

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 *seconds*

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:

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

Command History

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

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

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** [**0** | **8**] *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.
0	Specifies an unencrypted password will follow
8	Specifies an encrypted password will follow.

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.
9.0(1)	Multiple context mode is supported.

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 8 yWIvi0qJAnGK5MRWQzrhIohkGP1wKb
```

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
7.0(1)	This command was introduced.

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.

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.
9.0(1)	Multiple context mode is supported.

Usage Guidelines

The **ospf network point-to-point non-broadcast** command lets you to transmit OSPF routes over VPN tunnels.

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

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

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

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.

otp expiration

To specify the duration in hours that an issued One-Time Password (OTP) for the local Certificate Authority (CA) enrollment page is valid, use the **otp expiration** command in ca server configuration mode. To reset the duration to the default number of hours, use the **no** form of this command.

otp expiration *timeout*

no otp expiration

Syntax Description

timeout	Specifies the time in hours users have to enroll for a certificate from the local CA before the OTP for the enrollment page expires. Valid values range from 1 to 720 hours (30 days).

Defaults

By default, a OTP expiration for certificate enrollment is 72 hours (3 days).

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
Ca server configuration	•	—	•	—	—

Command History

Release	Modification
8.0(2)	This command was introduced.

Usage Guidelines

The OTP expiration period specifies the number of hours that a user has to log in to the enrollment page of the CA server. After the user logs in and enrolls for a certificate, the time period specified by the **enrollment retrieval** command starts.

Note The user OTP for enrolling for a certificate with the enrollment interface page is also used as the password to unlock the PKCS12 file containing the issued certificate and keypair for that user.

Examples

The following example specifies that the OTP for the enrollment page applies for 24 hours:

```
hostname(config)# crypto ca server
hostname(config-ca-server)# otp expiration 24
hostname(config-ca-server)#
```

The following example resets the OTP duration to the default of 72 hours:

```
hostname(config)# crypto ca server
hostname(config-ca-server))# no otp expiration
hostname(config-ca-server)#
```

Related Commands

Command	Description
crypto ca server	Provides access to the ca server configuration mode command set, which allows you to configure and manage the local CA.
enrollment-retrieval	Specifies the time in hours that an enrolled user can retrieve a PKCS12 enrollment file.
show crypto ca server	Displays the certificate authority configuration.

outstanding

To limit the number of unauthenticated e-mail proxy sessions, use the **outstanding** command in the applicable e-mail proxy configuration mode. To remove the attribute from the configuration, use the **no** form of this command.

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.

Usage Guidelines

Use the **no** version of this command to remove the attribute from the configuration, which permits an unlimited number of unauthenticated sessions. This also limit s 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 ASA authenticates the connection, it enters the “authenticated” state.

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

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.

override-svc-download

To configure the connection profile to override the group policy or username attributes configuration for downloading an AnyConnect or SSL VPN client, use the **override-svc-download** command from tunnel-group webvpn attributes configuration mode. To remove the command from the configuration, use the **no** form of the command:

override-svc-download enable

no override-svc-download enable

Defaults

The default is disabled. The ASA does not override the group policy or username attributes configuration for downloading the client.

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
8.0(2)	This command was introduced.

Usage Guidelines

The security appliance allows clientless, AnyConnect, or SSL VPN client connections for remote users based on whether clientless and/or SSL VPN is enabled in the group policy or username attributes with the **vpn-tunnel-protocol** command. The **svc ask** command further modifies the client user experience by prompting the user to download the client or return to the WebVPN home page.

However, you may want clientless users logging in under specific tunnel groups to not experience delays waiting for the download prompt to expire before being presented with the clientless SSL VPN home page. You can prevent delays for these users at the connection profile level with the **override-svc-download** command. This command causes users logging through a connection profile to be immediately presented with the clientless SSL VPN home page regardless of the **vpn-tunnel-protocol** or **svc ask** command settings.

Examples

In the following example, the user enters tunnel-group webvpn attributes configuration mode for the connection profile *engineering* and enables the connection profile to override the group policy and username attribute settings for client download prompts:

```
hostname(config)# tunnel-group engineering webvpn-attributes
hostname(config-tunnel-webvpn)# override-svc-download
```

Related Commands

Command	Description
show webvpn svc	Displays information about installed SSL VPN clients.
svc	Enables or requires the SSL VPN client for a specific group or user.
svc image	Specifies a client package file that the ASA expands in cache memory for downloading to remote PCs.