CHAPTER 16

Configuring Object Groups

You can configure access lists in modules, or object groups, to simplify access list creation and maintenance. This chapter describes how to configure, organize, and display object groups, and it includes the following sections:

Configuring Object Groups

This section includes the following topics:

Information About Object Groups

By grouping like objects together, you can use the object group in an ACE instead of having to enter an ACE for each object separately. You can create the following types of object groups:

- Protocol
- Network
- Service
- ICMP type

For example, consider the following three object groups:

- MyServices—Includes the TCP and UDP port numbers of the service requests that are allowed access to the internal network.
- TrustedHosts—Includes the host and network addresses allowed access to the greatest range of services and servers.
- PublicServers—Includes the host addresses of servers to which the greatest access is provided.

After creating these groups, you could use a single ACE to allow trusted hosts to make specific service requests to a group of public servers.

You can also nest object groups in other object groups.

Note The ACE system limit applies to expanded access lists. If you use object groups in ACEs, the number of actual ACEs that you enter is fewer, but the number of expanded ACEs is the same as without object groups. In many cases, object groups create more ACEs than if you added them manually because creating ACEs manually leads you to summarize addresses more than an object group does. For example, consider a network object group with 100 sources, a network object group with 100 destinations, and a port object group with 5 ports. Permitting the ports from sources to destinations could result in 50,000 ACEs (5 x 100 x 100) in the expanded access list.

Licensing Requirements for Object Groups

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations for Object Groups

This section includes the guidelines and limitations for this feature:

- Context Mode Guidelines, page 16-3
- Firewall Mode Guidelines, page 16-3
- IPv6 Guidelines, page 16-3
- Additional Guidelines and Limitations, page 16-3

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall modes.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines and Limitations

The following guidelines and limitations apply to object groups:

- Object groups must have unique names. While you might want to create a network object group named “Engineering” and a service object group named “Engineering,” you need to add an identifier (or “tag”) to the end of at least one object group name to make it unique. For example, you can use the names “Engineering_admins” and “Engnineering_hosts” to make the object group names unique and to aid in identification.
- After you add an object group you can add more objects as required by following the same procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects: the command you already set remains in place unless you remove the object group with the **no** form of the command.
- Objects such as hosts, protocols, or services can be grouped, and then you can enter a single command using the group name to apply 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.

Note You cannot remove an object group or make an object group empty if it is used in an access list. For information about removing object groups, see the “Removing Object Groups” section on page 16-8.

- The security appliance does not support IPv6 nested object groups, so you cannot group an object with IPv6 entities under another IPv6 object-group.

Adding Object Groups

This section includes the following topics:

- Adding a Protocol Object Group, page 16-4
- Adding a Network Object Group, page 16-5
- Adding a Service Object Group, page 16-6
- Adding an ICMP Type Object Group, page 16-7

Adding a Protocol Object Group

To add or change a protocol object group, perform the steps in this section. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

Detailed Steps

	Command	Purpose

Step 1	**`object-group protocol`** *`obj_grp_id`*
	
	Example:
	`hostname(config)# object-group protocol tcp_udp_icmp`
		Adds a protocol group. The *obj_grp_id* is a text string up to 64 characters in length and can be any combination of letters, digits, and the following characters:
		• underscore "_"
		• dash "-"
		• period "."
		The prompt changes to protocol configuration mode.

Step 2	**`description`** *`text`*
	
	Example:
	`hostname(config-protocol)# description New` *`Group`*
		(Optional) Adds a description. The description can be up to 200 characters.

Step 3	**`protocol-object`** *`protocol`*
	
	Example:
	`hostname(config-protocol)# protocol-object tcp`
		Defines the protocols in the group. Enter the command for each protocol. The protocol is the numeric identifier of the specified IP protocol (1 to 254) or a keyword identifier (for example, **icmp**, **tcp**, or **udp**). To include all IP protocols, use the keyword **ip**. For a list of protocols that you can specify, see the "Protocols and Applications" section on page C-11.

Example

To create a protocol group for TCP, UDP, and ICMP, enter the following commands:

```
hostname (config)# object-group protocol tcp_udp_icmp
hostname (config-protocol)# protocol-object tcp
hostname (config-protocol)# protocol-object udp
hostname (config-protocol)# protocol-object icmp
```

Adding a Network Object Group

A network object group supports IPv4 and IPv6 addresses, depending upon the type of access list. For more information about IPv6 access lists, see Chapter 15, "Adding an IPv6 Access List."

To add or change a network object group, perform the steps in this section. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the no form of the command.

Detailed Steps

	Command	Purpose

Step 1	**`object-group network`** *`grp_id`*
	Example:
	`hostname(config)# object-group network admins`
		Adds a network group.
		The *grp_id* is a text string up to 64 characters in length and can be any combination of letters, digits, and the following characters:
		• underscore "_"
		• dash "-"
		• period "."
		The prompt changes to protocol configuration mode.

Step 2	**`description`** *`text`*
	Example:
	`hostname(config-network)# Administrator Addresses`
		(Optional) Adds a description. The description can be up to 200 characters.

Step 3	**`network-object`** *`network`* `{host ip_address | ip_address mask}`
	Example:
	`hostname(config-network)# network-object host 10.1.1.4`
		Defines the networks in the group. Enter the command for each network or address.

Example

To create a network group that includes the IP addresses of three administrators, enter the following commands:

```
hostname (config)# object-group network admins
hostname (config-protocol)# description Administrator Addresses
hostname (config-protocol)# network-object host 10.1.1.4
hostname (config-protocol)# network-object host 10.1.1.78
hostname (config-protocol)# network-object host 10.1.1.34
```

Adding a Service Object Group

To add or change a service object group, perform the steps in this section. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

Detailed Steps

	Command	Purpose

Step 1	**object-group service** *grp_id* {**tcp** | **udp** | **tcp-udp**}
	Example:
	`hostname(config)# object-group service services1 tcp-udp`
		Adds a service group. The *grp_id* is a text string up to 64 characters in length and can be any combination of letters, digits, and the following characters:
		• underscore "_"
		• dash "-"
		• period "."
		Specify the protocol for the services (ports) you want to add with either the **tcp**, **udp**, or **tcp-udp** keywords. Enter the **tcp-udp** keyword if your service uses both TCP and UDP with the same port number, for example, DNS (port53).
		The prompt changes to service configuration mode.

Step 2	**description** *text* **Example:** `hostname(config-service)# description DNS Group`	(Optional) Adds a description. The description can be up to 200 characters.

Step 3	**port-object** {**eq** *port* | **range** *begin_port end_port*}
	Example:
	`hostname(config-service)# port-object eq domain`
		Defines the ports in the group. Enter the command for each port or range of ports. For a list of permitted keywords and well-known port assignments, see the "Protocols and Applications" section on page C-11.

Example

To create service groups that include DNS (TCP/UDP), LDAP (TCP), and RADIUS (UDP), enter the following commands:

```
hostname (config)# object-group service services1 tcp-udp
hostname (config-service)# description DNS Group
hostname (config-service)# port-object eq domain

hostname (config)# object-group service services2 udp
hostname (config-service)# description RADIUS Group
hostname (config-service)# port-object eq radius
hostname (config-service)# port-object eq radius-acct

hostname (config)# object-group service services3 tcp
hostname (config-service)# description LDAP Group
hostname (config-service)# port-object eq ldap
```

Adding an ICMP Type Object Group

To add or change an ICMP type object group, perform the steps in this section. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

Detailed Steps

	Command	Purpose

Step 1	`object-group icmp-type grp_id`
	Example:
	`hostname(config)# object-group icmp-type ping`
		Adds an ICMP type object group. The *grp_id* is a text string up to 64 characters in length and can be any combination of letters, digits, and the following characters:
		• underscore "_"
		• dash "-"
		• period "."
		The prompt changes to ICMP type configuration mode.

Step 2	`description text`
	Example:
	`hostname(config-icmp-type)# description Ping Group`
		(Optional) Adds a description. The description can be up to 200 characters.

Step 3	`icmp-object icmp-type`
	Example:
	`hostname(config-icmp-type)# icmp-object echo-reply`
		Defines the ICMP types in the group. Enter the command for each type. For a list of ICMP types, see the"ICMP Types" section on page C-15.

Example

Create an ICMP type group that includes echo-reply and echo (for controlling ping) by entering the following commands.

```
hostname (config)# object-group icmp-type ping
hostname (config-service)# description Ping Group
hostname (config-service)# icmp-object echo
hostname (config-service)# icmp-object echo-reply
```

Removing Object Groups

You can remove a specific object group or remove all object groups of a specified type; however, you cannot remove an object group or make an object group empty if it is used in an access list.

Detailed Step

Step 1 Do one of the following:

no object-group *grp_id* **Example:** `hostname(config)# no object-group Engineering_host`	Removes the specified object group. The *grp_id* is a text string up to 64 characters in length and can be any combination of letters, digits, and the following characters: • underscore "_" • dash "-" • period "."
clear object-group [**protocol** | **network** | **services** | **icmp-type**] **Example:** `hostname(config)# clear-object group network`	Removes all object groups of the specified type. **Note** If you do not enter a type, all object groups are removed.

Monitoring Object Groups

To monitor object groups, enter the following commands:

Command	Purpose
show access-list	Displays the access list entries that are expanded out into individual entries without their object groupings.
show running-config object-group	Displays all current object groups.
show running-config object-group *grp_id*	Displays the current object groups by their group ID.
show running-config object-group *grp_type*	Displays the current object groups by their group type.

Nesting Object Groups

You can nest object groups heirarchically so that one object group can contain other object groups of the same type. However, the security appliance does not support IPv6 nested object groups, so you cannot group an object with IPv6 entities under another IPv6 object-group.

To nest an object group within another object group of the same type, first create the group that you want to nest (see the "Adding Object Groups" section on page 16-4) and then perform the steps in this section.

Detailed Steps

	Command	Purpose
Step 1	`object-group group {{protocol	network	icmp-type} grp_id |service grp_id {tcp	udp	tcp-udp}}`  **Example:** `hostname(config)# object-group network Engineering_group`	Adds or edits the specified object group type under which you want to nest another object group. The **service**_*grp_id*_ is a text string up to 64 characters in length and can be any combination of letters, digits, and the following characters: • underscore "_" • dash "-" • period "."

Step 2	`group-object group_id`
	
	Example:
	`hostname(config-network)# network-object host 10.1.1.5`
	`hostname(config-network)# network-object host 10.1.1.7`
	`hostname(config-network)# network-object host 10.1.1.9`
		Adds the specified group under the object group you specified in Step 1. The nested group must be of the same type. You can mix and match nexted group objects and regular objects within an object group.

Examples

Create network object groups for privileged users from various departments by entering the following commands:

```
hostname (config)# object-group network eng
hostname (config-network)# network-object host 10.1.1.5
hostname (config-network)# network-object host 10.1.1.9
hostname (config-network)# network-object host 10.1.1.89

hostname (config)# object-group network hr
hostname (config-network)# network-object host 10.1.2.8
hostname (config-network)# network-object host 10.1.2.12

hostname (config)# object-group network finance
hostname (config-network)# network-object host 10.1.4.89
hostname (config-network)# network-object host 10.1.4.100
```

You then nest all three groups together as follows:

```
hostname (config)# object-group network admin
hostname (config-network)# group-object eng
hostname (config-network)# group-object hr
hostname (config-network)# group-object finance
```

You only need to specify the admin object group in your ACE as follows:

```
hostname (config)# access-list ACL_IN extended permit ip object-group admin host
209.165.201.29
```

Feature History for Object Groups

Table 16-1 lists the release history for this feature.

Table 16-1 Feature History for Object Groups

Feature Name	Releases	Feature Information
Object groups	7.0	Object groups simplify access list creation and maintenance. The following commands were introduced or modified: **object-group** *protocol*, **object-group** *network*, **object-group** *service*, **object-group** *icmp_type*.

Using Object Groups with Access Lists

This section contains the following topics:

- Information About Using Object Groups with Access Lists, page 16-10
- Licensing Requirements for Using Object Groups with Access Lists, page 16-10
- Guidelines and Limitations for Using Object Groups with Access Lists, page 16-11
- Configuring Object Groups with Access Lists, page 16-11
- Monitoring the Use of Object Groups with Access Lists, page 16-12
- Configuration Examples for Using Object Groups with Access Lists, page 16-12
- Feature History for Using Object Groups with Access Lists, page 16-13

Information About Using Object Groups with Access Lists

You can use object groups in an access list, replace the normal protocol (*protocol*), network (*source_address mask*, and so on) service (*operator port*), or ICMP type (*icmp_type*) parameter with the **object-group** *grp_id* parameter.

Licensing Requirements for Using Object Groups with Access Lists

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations for Using Object Groups with Access Lists

This section includes the guidelines and limitations for this feature:

- Context Mode Guidelines, page 16-11
- Firewall Mode Guidelines, page 16-11
- IPv6 Guidelines, page 16-3
- Additional Guidelines and Limitations, page 16-11

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall modes.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines and Limitations

The following guidelines and limitations apply to using object groups with access lists:

You do not have to use object groups for all parameters; for example, you can use an object group for the source address but identify the destination address with an address and mask.

Configuring Object Groups with Access Lists

To use object groups for all available parameters in the **access-list** {**tcp** | **udp**} command, enter the following command:

Command	Purpose
access-list *access_list_name* [**line** *line_number*] [**extended**] {***deny*** | ***permit***} {**tcp** | **udp**} **object-group** *nw_grp_id* [**object-group** *svc_grp_id*] **object-group** *nw_grp_id* [**object-group** *svc_grp_id*] [**log** [[*level*] [**interval** *secs*] | **disable** | **default**]] [**inactive** | **time-range** *time_range_name*] `hostname(config)# access-list 104 permit tcp object-group A object-group B inactive`	Configures object groups with access lists. For a detailed list of command options, see the **access list estended** command in the *Cisco Adaptive Security Appliance Command Reference*. For a complete configuration example about using object groups with access lists, see the "Configuration Examples for Scheduling Access List Activation" section on page 16-16.

Monitoring the Use of Object Groups with Access Lists

To monitor the use of object groups with accesslists, enter the following commands:

Command	Purpose
show access-list	Displays the access list entries that are expanded out into individual entries without their object groupings.
show object-group [**protocol** | **network** | **service** | **icmp-type** | **id** *grp_id*]	Displays a list of the currently configured object groups. If you enter the command without any parameters, the system displays all configured object groups.
show running-config object-group	Displays all current object groups.
show running-config object-group *grp_id*	Displays the current object groups by their group ID.
show running-config object-group *grp_type*	Displays the current object groups by their group type.

Example

The following is sample output from the **show object-group** command:

```
hostname# show object-group
object-group network ftp_servers
  description: This is a group of FTP servers
  network-object host 209.165.201.3
  network-object host 209.165.201.4
object-group network TrustedHosts
  network-object host 209.165.201.1
  network-object 192.168.1.0 255.255.255.0
  group-object ftp_servers
```

Configuration Examples for Using Object Groups with Access Lists

The following normal access list that does not use object groups restricts several hosts on the inside network from accessing several web servers. All other traffic is allowed.

```
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host 209.165.201.29
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host 209.165.201.29
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host 209.165.201.29
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host 209.165.201.16
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host 209.165.201.16
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host 209.165.201.16
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host 209.165.201.78
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host 209.165.201.78
eq www
```

```
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host 209.165.201.78
eq www
hostname(config)# access-list ACL_IN extended permit ip any any
hostname(config)# access-group ACL_IN in interface inside
```

If you make two network object groups, one for the inside hosts, and one for the web servers, then the configuration can be simplified and can be easily modified to add more hosts:

```
hostname(config)# object-group network denied
hostname(config-network)# network-object host 10.1.1.4
hostname(config-network)# network-object host 10.1.1.78
hostname(config-network)# network-object host 10.1.1.89

hostname(config-network)# object-group network web
hostname(config-network)# network-object host 209.165.201.29
hostname(config-network)# network-object host 209.165.201.16
hostname(config-network)# network-object host 209.165.201.78

hostname(config-network)# access-list ACL_IN extended deny tcp object-group denied
object-group web eq www
hostname(config)# access-list ACL_IN extended permit ip any any
hostname(config)# access-group ACL_IN in interface inside
```

Feature History for Using Object Groups with Access Lists

Table 16-2 lists the release history for this feature.

Table 16-2 Feature History for Using Object Groups with Access Lists

Feature Name	Releases	Feature Information
Object groups	7.0	Object groups simplify access list creation and maintenance. The following commands were introduced or modified: **object-group** *protocol*, **object-group** *network*, **object-group** *service*, **object-group** *icmp_type*.

Adding Remarks to Access Lists

You can include remarks about entries in any access list, including extended, EtherType, IPv6, standard, and Webtype access lists. The remarks make the access list easier to understand.

To add a remark after the last **access-list** command you entered, enter the following command:

Command	Purpose
access-list *access_list_name* **remark** *text* **Example:** `hostname(config)# access-list OUT remark - this is the inside admin address`	Adds a remark after the last access-list command you entered. The text can be up to 100 characters in length. You can enter leading spaces at the beginning of the text. Trailing spaces are ignored. If you enter the remark before any **access-list** command, then the remark is the first line in the access list. If you delete an access list using the **no access-list** *access_list_name* command, then all the remarks are also removed.

Example

You can add a remark before each ACE, and the remarks appear in the access list in these location. Entering a dash (-) at the beginning of a remark helps to set it apart from the ACE.

```
hostname(config)# access-list OUT remark - this is the inside admin address
hostname(config)# access-list OUT extended permit ip host 209.168.200.3 any
hostname(config)# access-list OUT remark - this is the hr admin address
hostname(config)# access-list OUT extended permit ip host 209.168.200.4 any
```

Scheduling Extended Access List Activation

This section includes the following topics:

- Information About Scheduling Access List Activation, page 16-14
- Licensing Requirements for Scheduling Access List Activation, page 16-14
- Guidelines and Limitations for Scheduling Access List Activation, page 16-15
- Configuring and Applying Time Ranges, page 16-15
- Configuration Examples for Scheduling Access List Activation, page 16-16
- Feature History for Scheduling Access Lis t Activation, page 16-17

Information About Scheduling Access List Activation

You can schedule each ACE in an access list to be activated at specific times of the day and week by applying a time range to the ACE.

Licensing Requirements for Scheduling Access List Activation

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations for Scheduling Access List Activation

This section includes the guidelines and limitations for this feature:

- Context Mode Guidelines, page 16-15
- Firewall Mode Guidelines, page 16-15
- IPv6 Guidelines, page 16-11
- Additional Guidelines and Limitations, page 16-15

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall modes.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines and Limitations

The following guidelines and limitations apply to using object groups with access lists:

- Users could experience a delay of approximately 80 to 100 seconds after the specified end time for the ACL to become inactive. For example, if the specified end time is 3:50, because the end time is inclusive, the command is picked up anywhere between 3:51:00 and 3:51:59. After the command is picked up, the security appliance finishes any currently running task and then services the command to deactivate the ACL.
- Multiple periodic entries are allowed per **time-range** command. If a **time-range** command has both **absolute** and **periodic** values specified, then the **periodic** commands are evaluated only after the **absolute** start time is reached, and they are not further evaluated after the **absolute** end time is reached.

Configuring and Applying Time Ranges

You can add a time range to implement a time-based access list. To identify the time range, perform the steps in this section.

Detailed Steps

	Command	Purpose

Step 1	**`time-range`** *`name`*
	Example:
	`hostname(config)# time range Sales`
		Identifies the time-range name.

Step 2	Do one of the following:	

	Command	Purpose
	periodic *days-of-the-week time* **to** [*days-of-the-week*] *time* **Example:** hostname(config-time-range)# periodic monday 7:59 to friday 17:01	Specifies a recurring time range. You can specify the following values for *days-of-the-week*: • **monday**, **tuesday**, **wednesday**, **thursday**, **friday**, **saturday**, or **sunday**. • **daily** • **weekdays** • **weekend** The *time* is in the format *hh*:*mm*. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.
	absolute **start** *time date* [**end** *time date*] **Example:** hostname(config-time-range)# absolute start 7:59 2 january 2009	Specifies an absolute time range. The *time* is in the format *hh*:*mm*. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m. The *date* is in the format *day month year*; for example, **1 january 2006**.
Step 3	**access-list** *access_list_name* [**extended**] {**deny** | **permit**}...[**time-range** *name*] **Example:** hostname(config)# access list Marketing extended deny tcp host 209.165.200.225 host 209.165 201.1 time-range Pacific_Coast	Applies the time range to an ACE. **Note** If you also enable logging for the ACE, use the **log** keyword before the **time-range** keyword. If you disable the ACE using the **inactive** keyword, use the **inactive** keyword as the last keyword. See Chapter 11, "Adding an Extended Access List," for complete **access-list** command syntax.

Example

The following example binds an access list named "Sales" to a time range named "New_York_Minute."

```
hostname(config)# access-list Sales line 1 extended deny tcp host 209.165.200.225 host
209.165.201.1 time-range New_York_Minute
```

Configuration Examples for Scheduling Access List Activation

The following is an example of an absolute time range beginning at 8:00 a.m. on January 1, 2006. Because no end time and date are specified, the time range is in effect indefinitely.

```
hostname(config)# time-range for2006
hostname(config-time-range)# absolute start 8:00 1 january 2006
```

The following is an example of a weekly periodic time range from 8:00 a.m. to 6:00 p.m on weekdays:

```
hostname(config)# time-range workinghours
hostname(config-time-range)# periodic weekdays 8:00 to 18:00
```

Feature History for Scheduling Access Lis t Activation

Table 16-3 lists the release history for this feature.

Table 16-3 Feature History for Scheduling Access List Activation

Feature Name	Releases	Feature Information
Scheduling access list activation	7.0	You can schedule each ACE in an access list to be activated at specific times of the day and week. The following commands were introduced or modified: **object-group** *protocol*, **object-group** *network*, **object-group** *service*, **object-group** *icmp_type*.