Adding an Extended Access Control List

This chapter describes how to configure extended access control lists (ACLs), and it includes the following sections:

Information About Extended ACLs

ACLs are used to control network access or to specify traffic for many features to act upon. An extended ACL is made up of one or more access control entries (ACEs). Each ACE specifies a source and destination for matching traffic. You can identify parameters within the **access-list** command, or you can create objects or object groups for use in the ACL.

Access Control Entry Order

An ACL is made up of one or more ACEs. Each ACE that you enter for a given ACL name is appended to the end of the ACL.

The order of ACEs is important. When the security appliance decides whether to forward or drop a packet, the security appliance tests the packet against each ACE in the order in which the entries are listed. After a match is found, no more ACEs are checked. For example, if you create an ACE at the beginning of an ACL that explicitly permits all traffic, no further statements are ever checked.

You can disable an ACE by making it inactive.

NAT and ACLs

When using NAT or PAT, mapped addresses and ports are no longer required in an ACL for several features. You should now always use the real, untranslated addresses and ports for these features. Using the real address and port means that if the NAT configuration changes, you do not need to change the ACLs.

Note For ACL migration information, see the *Cisco ASA 5500 Migration to Version 8.3 and Later.*

Features That Use Real IP Addresses

The following commands and features use real IP addresses in the ACLs:

- **access-group** command
- Modular Policy Framework **match access-list** command
- Botnet Traffic Filter **dynamic-filter enable classify-list** command
- AAA **aaa ... match** commands
- WCCP **wccp redirect-list group-list** command

For example, if you configure NAT for an inside server, 10.1.1.5, so that it has a publicly routable IP address on the outside, 209.165.201.5, then the access rule to allow the outside traffic to access the inside server needs to reference the server's real IP address (10.1.1.5), and not the mapped address (209.165.201.5):

```
hostname(config)# object network server1
hostname(config-network-object)# host 10.1.1.5
hostname(config-network-object)# nat (inside,outside) static 209.165.201.5

hostname(config)# access-list OUTSIDE extended permit tcp any host 10.1.1.5 eq www
hostname(config)# access-group OUTSIDE in interface outside
```

Features That Use Mapped IP Addresses

The following features use ACLs, but these ACLs will continue to use the mapped values as seen on an interface:

- IPsec ACLs
- **capture** command ACLs
- Per-user ACLs
- Routing protocol ACLs
- All other feature ACLs...

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.

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 ASA finishes any currently running task and then services the command to deactivate the ACL.

Licensing Requirements for Extended ACLs

Model	License Requirement
All models	Base License.

Guidelines and Limitations

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall modes.

IPv6 Guidelines

Supports IPv6.

Features That Do Not Support IDFW, FQDN, and TrustSec ACLs

The following features use ACLs, but cannot accept an ACL with IDFW, FQDN, or TrustSec values:

- **route-map** command
- VPN **crypto map** command
- VPN **group-policy** command, except for **vpn-filter**
- WCCP
- DAP

Additional Guidelines and Limitations

- **Tip:** Enter the ACL name in uppercase letters so that the name is easy to see in the configuration. You might want to name the ACL for the interface (for example, INSIDE), or you can name it for the purpose for which it is created (for example, NO_NAT or VPN).
- Typically, you identify the **ip** keyword for the protocol, but other protocols are accepted. For a list of protocol names, see the "Protocols and Applications" section on page 1-11.
- You can specify the source and destination ports only for the TCP or UDP protocols. For a list of permitted keywords and well-known port assignments, see the "TCP and UDP Ports" section on page 1-11. DNS, Discard, Echo, Ident, NTP, RPC, SUNRPC, and Talk each require one definition for TCP and one for UDP. TACACS+ requires one definition for port 49 on TCP.
- When you specify a network mask, the method is different from the Cisco IOS software **access-list** command. The ASA uses a network mask (for example, 255.255.255.0 for a Class C mask). The Cisco IOS mask uses wildcard bits (for example, 0.0.0.255).

Default Settings

Table 1-1 lists the default settings for extended ACL parameters.

Table 1-1 Default Extended ACL Parameters

Parameters	Default
ACE logging	ACE logging generates system log message 106023 for denied packets. A deny ACE must be present to log denied packets.
log	When the **log** keyword is specified, the default level for system log message 106100 is 6 (informational), and the default interval is 300 seconds.

Configuring Extended ACLs

This section shows how to add ACEs of various types to an ACL and includes the following topics:

Adding an ACE for IP Address or Fully Qualified Domain Name-Based Policy

This section lets you control traffic based on IP addresses or fully qualified domain names (FQDNs). An ACL is made up of one or more access control entries (ACEs) with the same ACL ID. To create an ACL you start by creating an ACE and applying a list name. An ACL with one entry is still considered a list, although you can add multiple entries to the list.

Prerequisites

(Optional) Create network objects or object groups according to the "Configuring Network Objects and Groups" section on page 1-2. Objects can contain an IP address (host, subnet, or range) or an FQDN. Object groups contain multiple objects or inline entries.

Guidelines

To delete an ACE, enter the **no access-list** command with the entire command syntax string as it appears in the configuration. To remove the entire ACL, use the **clear configure access-list** command.

Detailed Steps

Command	Purpose
access-list *access_list_name* [**line** *line_number*] **extended** {**deny** | **permit**} *protocol_argument source_address_argument dest_address_argument* [**log** [[*level*] [**interval** *secs*] | **disable** | **default**]] [**inactive** | **time-range** *time_range_name*] **Example:** `hostname(config)# access-list ACL_IN extended permit ip any any`	Adds an ACE for IP address or FQDN policy. • Line number—The **line** *line_number* option specifies the line number at which insert the ACE; otherwise, the ACE is added to the end of the ACL. • Permit or Deny—The **deny** keyword denies or exempts a packet if the conditions are matched. The **permit** keyword permits a packet if the conditions are matched. • Protocol—The *protocol_argument* specifies the IP protocol: – *name* or *number*—Specifies the protocol name or number. Specify **ip** to apply to all protocols. – **object-group** *protocol_grp_id*—Specifies a protocol object group created using the **object-group protocol** command. – **object** *service_obj_id*—Specifies a service object created using the **object service** command. A TCP, UDP, or ICMP service object can include a protocol *and* a source and/or destination port or ICMP type and code. – **object-group** *service_grp_id*—Specifies a service object group created using the **object-group service** command. • Source Address, Destination Address—The *source_address_argument* specifies the IP address or FQDN from which the packet is being sent, and the *dest_address_argument* specifies the IP address or FQDN to which the packet is being sent: – **host** *ip_address*—Specifies an IPv4 host address. – *dest_ip_address mask*—Specifies an IPv4 network address and subnet mask. – *ipv6-address*/*prefix-length*—Specifies an IPv6 host or network address and prefix. – **any**, **any4**, and **any6**—**any** specifies both IPv4 and IPv6 traffic; **any4** specifies only IPv4 traffic; and **any6** specifies any6 traffic. – **object** *nw_obj_id*—Specifies a network object created using the **object network** command. – **object-group** *nw_grp_id*—Specifies a network object group created using the **object-group network** command. • Logging—**log** arguments set logging options when an ACE matches a packet for network access (an ACL applied with the **access-group** command). • Activation—Inactivates or enables a time range that the ACE is active; see the **time-range** command for information about defining a time range.

Adding an ACE for TCP or UDP-Based Policy, with Ports

This section lets you control traffic based on IP addresses or fully qualified domain names (FQDNs) along with TCP or UDP ports. An ACL is made up of one or more access control entries (ACEs) with the same ACL ID. To create an ACL you start by creating an ACE and applying a list name. An ACL with one entry is still considered a list, although you can add multiple entries to the list.

Prerequisites

- (Optional) Create network objects or object groups according to the "Configuring Network Objects and Groups" section on page 1-2. Objects can contain an IP address (host, subnet, or range) or an FQDN. Object groups contain multiple objects or inline entries.
- (Optional) Create service objects or groups according to the "Configuring Service Objects and Service Groups" section on page 1-5.

Guidelines

To delete an ACE, enter the **no access-list** command with the entire command syntax string as it appears in the configuration. To remove the entire ACL, use the **clear configure access-list** command.

Detailed Steps

Command	Purpose
access-list *access_list_name* [**line** *line_number*] **extended** {**deny** | **permit**} {**tcp** | **udp**} *source_address_argument* [*port_argument*] *dest_address_argument* [*port_argument*] [**log** [[*level*] [**interval** *secs*] | **disable** | **default**]] [**inactive** | **time-range** *time_range_name*] **Example:** `hostname(config)# access-list ACL_IN extended deny tcp any host 209.165.201.29 eq www`	Adds an ACE for IP address or FQDN policy, as well as optional TCP or UDP ports. For common keywords and arguments, see the "Adding an ACE for IP Address or Fully Qualified Domain Name-Based Policy" section on page 1-4. Keywords and arguments specific to this type of ACE include the following: *port_argument* specifies the source and/or destination port. Available arguments include: • *operator port*—The *operator* can be one of the following: – **lt**—less than – **gt**—greater than – **eq**—equal to – **neq**—not equal to – **range**—an inclusive range of values. When you use this operator, specify two port numbers, for example: `range 100 200` The *port* can be the integer or name of a TCP or UDP port. DNS, Discard, Echo, Ident, NTP, RPC, SUNRPC, and Talk each require one definition for TCP and one for UDP. TACACS+ requires one definition for port 49 on TCP. • **object-group** *service_grp_id*—Specifies a service object group created using the **object-group service** command.

Adding an ACE for ICMP-Based Policy, with ICMP Type

This section lets you control traffic based on IP addresses or fully qualified domain names (FQDNs) along with the ICMP type. An ACL is made up of one or more access control entries (ACEs) with the same ACL ID. To create an ACL you start by creating an ACE and applying a list name. An ACL with one entry is still considered a list, although you can add multiple entries to the list.

Prerequisites

- (Optional) Create network objects or object groups according to the "Configuring Network Objects and Groups" section on page 1-2. Objects can contain an IP address (host, subnet, or range) or an FQDN. Object groups contain multiple objects or inline entries.
- (Optional) Create ICMP groups according to the "Configuring an ICMP Group" section on page 1-10.

Guidelines

To delete an ACE, enter the **no access-list** command with the entire command syntax string as it appears in the configuration. To remove the entire ACL, use the **clear configure access-list** command.

Detailed Steps

Command	Purpose
access-list *access_list_name* [**line** *line_number*] **extended** {**deny** | **permit**} **icmp** *source_address_argument dest_address_argument* [*icmp_argument*] [**log** [[*level*] [**interval** *secs*] | **disable** | **default**]] [**inactive** | **time-range** *time_range_name*] **Example:** `hostname(config)# access-list abc extended permit icmp any any object-group obj_icmp_1`	Adds an ACE for IP address or FQDN policy, as well as optional TCP or UDP ports. For common keywords and arguments, see the "Adding an ACE for IP Address or Fully Qualified Domain Name-Based Policy" section on page 1-4. Keywords and arguments specific to this type of ACE include the following: *icmp_argument* specifies the ICMP type and code. • *icmp_type* [*icmp_code*]—Specifies the ICMP type by name or number, and the optional ICMP code for that type. If you do not specify the code, then all codes are used. • **object-group** *icmp_grp_id*—Specifies an ICMP object group created using the **object-group icmp** command.

Adding an ACE for User-Based Policy (Identity Firewall)

If you configure the identity firewall feature, you can control traffic based on user identity.

Prerequisites

See Chapter 1, "Configuring the Identity Firewall," to enable IDFW.

Detailed Steps

Command	Purpose
access-list *access_list_name* [**line** *line_number*] **extended** {**deny** | **permit**} *protocol_argument* [*user_argument*] *source_address_argument* [*port_argument*] *dest_address_argument* [*port_argument*] [**log** [[*level*] [**interval** *secs*] | **disable** | **default**]] [**inactive** | **time-range** *time_range_name*] **Example:** `hostname(config)# access-list v1 extended permit ip user LOCAL\idfw any 10.0.0.0 255.255.255.0`	Adds an ACE for IP address or FQDN policy, as well as optional usernames and/or groups. For common keywords and arguments, see the "Adding an ACE for IP Address or Fully Qualified Domain Name-Based Policy" section on page 1-4. Keywords and arguments specific to this type of ACE include the following: *user_argument* is for use with the identity firewall feature, and specifies the user or group for which to match traffic in addition to the source address. Available arguments include the following: • **object-group-user** *user_obj_grp_id*—Specifies a user object group created using the **object-group user** command. • **user** {[*domain_nickname*\]*name* | **any** | **none**}—Specifies a username. Specify **any** to match all users with user credentials, or **none** to match users without user credentials. These options are especially useful for combining **access-group** and **aaa authentication match** policies. • **user-group** [*domain_nickname*\\]*user_group_name*—Specifies a user group name. **Note** Although not shown in the syntax at left, you can also use TrustSec security group arguments.

Adding an ACE for Security Group-Based Policy (TrustSec)

If you configure the Cisco TrustSec feature, you can control traffic based on security groups.

Prerequisites

See Chapter 1, "Configuring the ASA to Integrate with Cisco TrustSec," to enable TrustSec.

Detailed Steps

Command	Purpose
access-list *access_list_name* [**line** *line_number*] **extended** {**deny** | **permit**} *protocol_argument* [*security_group_argument*] *source_address_argument* [*port_argument*] [*security_group_argument*] *dest_address_argument* [*port_argument*] [**log** [[*level*] [**interval** *secs*] | **disable** | **default**]] [**inactive** | **time-range** *time_range_name*] **Example:** `hostname(config)# access-list v1 extended permit ip user LOCAL\idfw any 10.0.0.0 255.255.255.0`	Adds an ACE for IP address or FQDN policy, as well as optional security groups. For common keywords and arguments, see the "Adding an ACE for IP Address or Fully Qualified Domain Name-Based Policy" section on page 1-4. Keywords and arguments specific to this type of ACE include the following: *security_group_argument* is for use with the TrustSec feature, and specifies the security group for which to match traffic in addition to the source or destination address. Available arguments include the following: • **object-group-security** *security_obj_grp_id*—Specifies a security object group created using the **object-group security** command. • **security-group** {**name** *security_grp_id* | **tag** *security_grp_tag*}—Specifies a security group name or tag. **Note** Although not shown in the syntax at left, you can also use Identity Firewall user arguments.

Adding Remarks to ACLs

You can include remarks about entries in any ACL. The remarks make the ACL easier to understand.

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

Detailed Steps

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 ACL. If you delete an ACL using the **no access-list** *access_list_name* command, then all the remarks are also removed.

Examples

You can add remarks before each ACE, and the remark appears in the ACL in this location. Entering a dash (-) at the beginning of the remark helps set it apart from the ACEs.

```
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
```

Monitoring Extended ACLs

To monitor extended ACLs, enter one of the following commands:

Command	Purpose
`show access-list`	Displays the ACEs by number.
`show running-config access-list`	Displays the current running access-list configuration.

Configuration Examples for Extended ACLs

This section includes the following topics:

Configuration Examples for Extended ACLs (No Objects)

The following ACL allows all hosts (on the interface to which you apply the ACL) to go through the ASA:

```
hostname(config)# access-list ACL_IN extended permit ip any any
```

The following sample ACL prevents hosts on 192.168.1.0/24 from accessing the 209.165.201.0/27 network. All other addresses are permitted.

```
hostname(config)# access-list ACL_IN extended deny tcp 192.168.1.0 255.255.255.0
209.165.201.0 255.255.255.224
hostname(config)# access-list ACL_IN extended permit ip any any
```

If you want to restrict access to selected hosts only, then enter a limited permit ACE. By default, all other traffic is denied unless explicitly permitted.

```
hostname(config)# access-list ACL_IN extended permit ip 192.168.1.0 255.255.255.0
209.165.201.0 255.255.255.224
```

The following ACL restricts all hosts (on the interface to which you apply the ACL) from accessing a website at address 209.165.201.29. All other traffic is allowed.

```
hostname(config)# access-list ACL_IN extended deny tcp any host 209.165.201.29 eq www
hostname(config)# access-list ACL_IN extended permit ip any any
```

The following ACL that uses object groups restricts several hosts on the inside network from accessing several web servers. All other traffic is allowed.

```
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
```

The following example temporarily disables an ACL that permits traffic from one group of network objects (A) to another group of network objects (B):

```
hostname(config)# access-list 104 permit ip host object-group A object-group B inactive
```

To implement a time-based ACL, use the **time-range** command to define specific times of the day and week. Then use the **access-list extended** command to bind the time range to an ACL. The following example binds an ACL 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
```

The following example shows a mixed IPv4/IPv6 ACL:

```
hostname(config)# access-list demoacl extended permit ip 2001:DB8:1::/64 10.2.2.0
255.255.255.0
hostname(config)# access-list demoacl extended permit ip 2001:DB8:1::/64 2001:DB8:2::/64
hostname(config)# access-list demoacl extended permit ip host 10.3.3.3 host 10.4.4.4
```

Configuration Examples for Extended ACLs (Using Objects)

The following normal ACL 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
```

Where to Go Next

Many features use ACLs. Apply the ACL to an interface. See the "Configuring Access Rules" section on page 1-7 for more information.

Feature History for Extended ACLs

Table 1-2 lists the release history for this feature.

Table 1-2 Feature History for Extended ACLs

Feature Name	Releases	Feature Information
Extended ACLs	7.0(1)	ACLs are used to control network access or to specify traffic for many features to act upon. An extended access control list is made up of one or more access control entries (ACE) in which you can specify the line number to insert the ACE, the source and destination addresses, and, depending upon the ACE type, the protocol, the ports (for TCP or UDP), or the IPCMP type (for ICMP). We introduced the following command: **access-list extended**.
Real IP addresses	8.3(1)	When using NAT or PAT, mapped addresses and ports are no longer required in an ACL for several features. You should now always use the real, untranslated addresses and ports for these features. Using the real address and port means that if the NAT configuration changes, you do not need to change the ACLs. See the "Features That Use Real IP Addresses" section on page 1-2 for more information.
Support for Identity Firewall	8.4(2)	You can now use identity firewall users and groups for the source and destination. You can use an identity firewall ACL with access rules, AAA rules, and for VPN authentication. We modified the following commands: **access-list extended**.
Support for TrustSec	9.0(1)	You can now use TrustSec security groups for the source and destination. You can use an identity firewall ACL with access rules. We modified the following commands: **access-list extended**.

Table 1-2 Feature History for Extended ACLs (continued)

Feature Name	Releases	Feature Information
Unified ACL for IPv4 and IPv6	9.0(1)	ACLs now support IPv4 and IPv6 addresses. You can even specify a mix of IPv4 and IPv6 addresses for the source and destination. The **any** keyword was changed to represent IPv4 and IPv6 traffic. The **any4** and **any6** keywords were added to represent IPv4-only and IPv6-only traffic, respectively. The IPv6-specific ACLs are deprecated. Existing IPv6 ACLs are migrated to extended ACLs. See the release notes for more information about migration. We modified the following commands: **access-list extended**, **access-list webtype**. We removed the following commands: **ipv6 access-list**, **ipv6 access-list webtype**, **ipv6-vpn-filter**
Extended ACLand object enhancement to filter ICMP traffic by ICMP code	9.0(1)	ICMP traffic can now be permitted/denied based on ICMP code. We introduced or modified the following commands: **access-list extended**, **service-object**, **service**.