Configuring Access Rules

This chapter describes how to control network access through the ASA using access rules and includes the following sections:

Note You use access rules to control network access in both routed and transparent firewall modes. In transparent mode, you can use both access rules (for Layer 3 traffic) and EtherType rules (for Layer 2 traffic).

To access the ASA interface for management access, you do not also need an access rule allowing the host IP address. You only need to configure management access according to the general operations configuration guide.

Information About Access Rules

You create an access rule by applying an extended or EtherType ACL to an interface or globally for all interfaces.You can use access rules in routed and transparent firewall mode to control IP traffic. An access rule permits or denies traffic based on the protocol, a source and destination IP address or network, and optionally the source and destination ports.

For transparent mode only, an EtherType rule controls network access for non-IP traffic. An EtherType rule permits or denies traffic based on the EtherType.

This section includes the following topics:

General Information About Rules

This section describes information for both access rules and EtherType rules, and it includes the following topics:

Implicit Permits

For routed mode, the following types of traffic are allowed through by default:

- Unicast IPv4 traffic from a higher security interface to a lower security interface.
- Unicast IPv6 traffic from a higher security interface to a lower security interface.

For transparent mode, the following types of traffic are allowed through by default:

- Unicast IPv4 traffic from a higher security interface to a lower security interface.
- Unicast IPv6 traffic from a higher security interface to a lower security interface.
- ARPs in both directions.

Note ARP traffic can be controlled by ARP inspection, but cannot be controlled by an access rule.

- BPDUs in both directions.

For other traffic, you need to use either an extended access rule (IPv4 and IPv6) or an EtherType rule (non-IPv4/IPv6).

Information About Interface Access Rules and Global Access Rules

You can apply an access rule to a specific interface, or you can apply an access rule globally to all interfaces. You can configure global access rules in conjunction with interface access rules, in which case, the specific interface access rules are always processed before the general global access rules.

Note Global access rules apply only to inbound traffic. See the "Inbound and Outbound Rules" section on page 6-3.

Using Access Rules and EtherType Rules on the Same Interface

You can apply one access rule and one EtherType rule to each direction of an interface.

Implicit Deny

ACLs have an implicit deny at the end of the list, so unless you explicitly permit it, traffic cannot pass. For example, if you want to allow all users to access a network through the ASA except for particular addresses, then you need to deny the particular addresses and then permit all others.

For EtherType ACLs, the implicit deny at the end of the ACL does not affect IP traffic or ARPs; for example, if you allow EtherType 8037, the implicit deny at the end of the ACL does not now block any IP traffic that you previously allowed with an extended ACL (or implicitly allowed from a high security interface to a low security interface). However, if you explicitly deny all traffic with an EtherType ACE, then IP and ARP traffic is denied.

If you configure a global access rule, then the implicit deny comes *after* the global rule is processed. See the following order of operations:

1. Interface access rule.
2. Global access rule.
3. Implicit deny.

Inbound and Outbound Rules

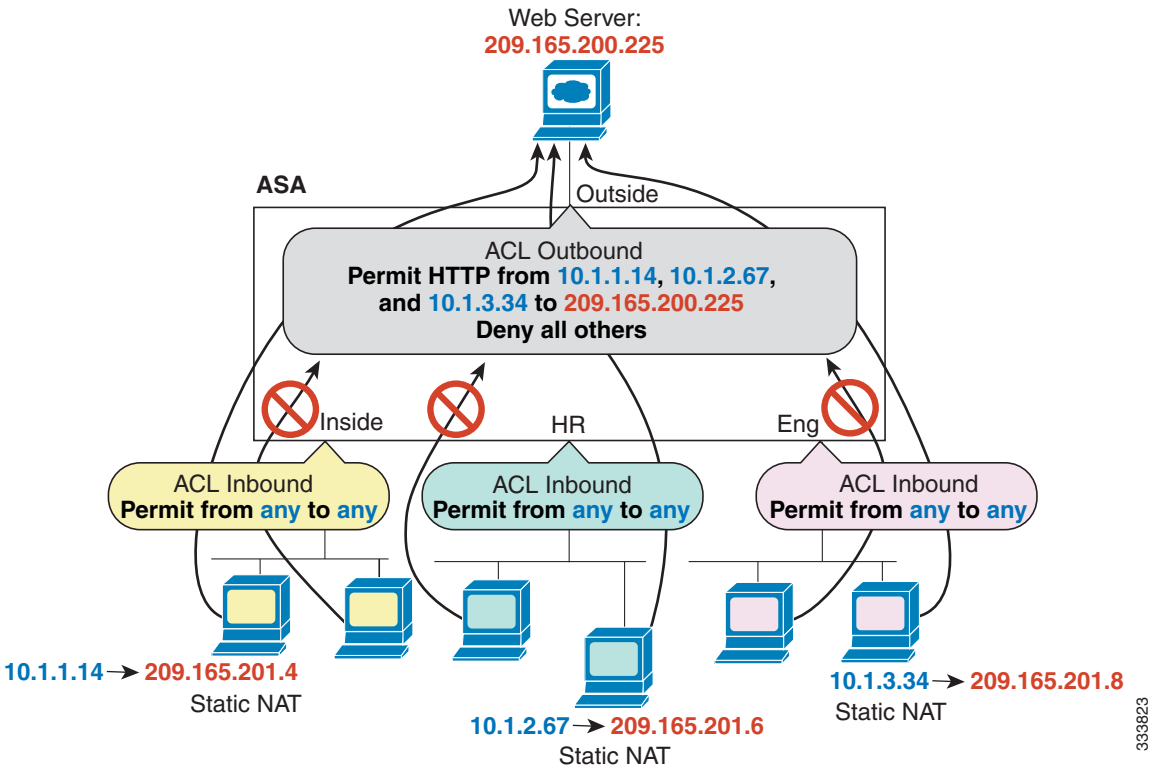
The ASA supports two types of ACLs:

- Inbound—Inbound access rules apply to traffic as it enters an interface. Global access rules are always inbound.
- Outbound—Outbound ACLs apply to traffic as it exits an interface.

Note "Inbound" and "outbound" refer to the application of an ACL on an interface, either to traffic entering the ASA on an interface or traffic exiting the ASA on an interface. These terms do not refer to the movement of traffic from a lower security interface to a higher security interface, commonly known as inbound, or from a higher to lower interface, commonly known as outbound.

An outbound ACL is useful, for example, if you want to allow only certain hosts on the inside networks to access a web server on the outside network. Rather than creating multiple inbound ACLs to restrict access, you can create a single outbound ACL that allows only the specified hosts. (See Figure 6-1.) The outbound ACL prevents any other hosts from reaching the outside network.

Figure 6-1 Outbound ACL



See the following commands for this example:

```
ciscoasa(config)# access-list OUTSIDE extended permit tcp host 10.1.1.14
host 209.165.200.225 eq www
ciscoasa(config)# access-list OUTSIDE extended permit tcp host 10.1.2.67
host 209.165.200.225 eq www
ciscoasa(config)# access-list OUTSIDE extended permit tcp host 10.1.3.34
host 209.165.200.225 eq www
ciscoasa(config)# access-group OUTSIDE out interface outside
```

Transactional-Commit Model

The ASA rule-engine supports a new feature for rule updation called the Transactional-Commit Model. When this feature is enabled, a rule update is applied after the rule compilation is completed; without affecting the rule matching performance. With the legacy model, rule updates take effect immediately but rule matching slows down during the rule compilation period. This feature is useful to prevent potential packet drops during large compilation of rules under high traffic conditions. This feature is also useful to reduce the rule compilation time under two specific patterns of configurations:

- Preventing packet drops while compiling large rules during high traffic rates.
- Reducing rule compilation time while updating a large number of similar rules.

Guidelines and Limitations

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines and Limitations

Evaluate the following alternatives before using the transactional commit model:

- While using large rules, try to optimize the number of rules by using the Object Group Search setting in Advanced Access Rule Configuration settings.
- Perform an incremental rule update instead of a bulk rule update. If a bulk update is necessary perform the bulk update during the maintenance window, when traffic is low.

Information About Extended Access Rules

This section describes information about extended access rules and includes the following topics:

Access Rules for Returning Traffic

For TCP and UDP connections for both routed and transparent mode, you do not need an access rule to allow returning traffic because the ASA allows all returning traffic for established, bidirectional connections.

For connectionless protocols such as ICMP, however, the ASA establishes unidirectional sessions, so you either need access rules to allow ICMP in both directions (by applying ACLs to the source and destination interfaces), or you need to enable the ICMP inspection engine. The ICMP inspection engine treats ICMP sessions as bidirectional connections. To control ping, specify **echo-reply** (**0**) (ASA to host) or **echo** (**8**) (host to ASA).

Allowing Broadcast and Multicast Traffic through the Transparent Firewall Using Access Rules

In routed firewall mode, broadcast and multicast traffic is blocked even if you allow it in an access rule, including unsupported dynamic routing protocols and DHCP (unless you configure DHCP relay). Transparent firewall mode can allow any IP traffic through.

Note Because these special types of traffic are connectionless, you need to apply an access rule to both interfaces, so returning traffic is allowed through.

Table 6-1 lists common traffic types that you can allow through the transparent firewall.

Table 6-1 Transparent Firewall Special Traffic

Traffic Type	Protocol or Port	Notes
DHCP	UDP ports 67 and 68	If you enable the DHCP server, then the ASA does not pass DHCP packets.
EIGRP	Protocol 88	—
OSPF	Protocol 89	—
Multicast streams	The UDP ports vary depending on the application.	Multicast streams are always destined to a Class D address (224.0.0.0 to 239.x.x.x).
RIP (v1 or v2)	UDP port 520	—

Management Access Rules

You can configure access rules that control management traffic destined to the ASA. Access control rules for to-the-box management traffic (defined by such commands as **http**, **ssh**, or **telnet**) have higher precedence than an management access rule applied with the **control-plane** option. Therefore, such permitted management traffic will be allowed to come in even if explicitly denied by the to-the-box ACL.

Information About EtherType Rules

This section describes EtherType rules and includes the following topics:

Supported EtherTypes and Other Traffic

An EtherType rule controls the following:

- EtherType identified by a 16-bit hexadecimal number, including common types IPX and MPLS unicast or multicast.
- Ethernet V2 frames.
- BPDUs, which are permitted by default. BPDUs are SNAP-encapsulated, and the ASA is designed to specifically handle BPDUs.
- Trunk port (Cisco proprietary) BPDUs. Trunk BPDUs have VLAN information inside the payload, so the ASA modifies the payload with the outgoing VLAN if you allow BPDUs.
- IS-IS.

The following types of traffic are not supported:

- 802.3-formatted frames—These frames are not handled by the rule because they use a length field as opposed to a type field.

Access Rules for Returning Traffic

Because EtherTypes are connectionless, you need to apply the rule to both interfaces if you want traffic to pass in both directions.

Allowing MPLS

If you allow MPLS, ensure that Label Distribution Protocol and Tag Distribution Protocol TCP connections are established through the ASA by configuring both MPLS routers connected to the ASA to use the IP address on the ASA interface as the router-id for LDP or TDP sessions. (LDP and TDP allow MPLS routers to negotiate the labels (addresses) used to forward packets.)

On Cisco IOS routers, enter the appropriate command for your protocol, LDP or TDP. The *interface* is the interface connected to the ASA.

```
ciscoasa(config)# mpls ldp router-id interface force
```

Or

```
ciscoasa(config)# tag-switching tdp router-id interface force
```

Licensing Requirements for Access Rules

Model	License Requirement
All models	Base License.

Prerequisites

Before you can create an access rule, create the ACL. See the general operations configuration guide for more information.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall modes.

IPv6 Guidelines

Supports IPv6. The source and destination addresses can include any mix of IPv4 and IPv6 addresses.

Per-User ACL Guidelines

- The per-user ACL uses the value in the **timeout uauth** command, but it can be overridden by the AAA per-user session timeout value.
- If traffic is denied because of a per-user ACL, syslog message 109025 is logged. If traffic is permitted, no syslog message is generated. The **log** option in the per-user ACL has no effect.

Default Settings

See the "Implicit Permits" section on page 6-2.

Configuring Access Rules

To apply an access rule, perform the following steps.

Detailed Steps

Command	Purpose
access-group *access_list* {{**in** | **out**} **interface** *interface_name* [**per-user-override** | **control-plane**] | **global**} **Example:** `ciscoasa(config)# access-group outside_access in interface outside`	Binds an ACL to an interface or applies it globally. Specify the extended or EtherType ACL name. You can configure one **access-group** command per ACL type per interface. You cannot reference empty ACLs or ACLs that contain only a remark. For an interface-specific rule: • The **in** keyword applies the ACL to inbound traffic. The **out** keyword applies the ACL to the outbound traffic. • Specify the **interface** name. • The **per-user-override** keyword (for inbound ACLs only) allows dynamic user ACLs that are downloaded for user authorization to override the ACL assigned to the interface. For example, if the interface ACL denies all traffic from 10.0.0.0, but the dynamic ACL permits all traffic from 10.0.0.0, then the dynamic ACL overrides the interface ACL for that user. By default, VPN remote access traffic is not matched against interface ACLs. However, if you use the **no sysopt connection permit-vpn** command to turn off this bypass, the behavior depends on whether there is a **vpn-filter** applied in the group policy and whether you set the **per-user-override** option: – No **per-user-override**, no **vpn-filter**—Traffic is matched against the interface ACL. – No **per-user-override**, **vpn-filter**—Traffic is matched first against the interface ACL, then against the VPN filter. – **per-user-override**, **vpn-filter**—Traffic is matched against the VPN filter only. See Per-User ACL Guidelines, page 6-8. • The **control-plane** keyword specifies if the rule is for to-the-box traffic. For a global rule, specify the **global** keyword to apply the ACL to the inbound direction of all interfaces.

Examples

The following example shows how to use the **access-group** command:

```
hostname(config)# access-list outside_access permit tcp any host 209.165.201.3 eq 80
hostname(config)# access-group outside_access interface outside
```

The **access-list** command lets any host access the global address using port 80. The **access-group** command specifies that the **access-list** command applies to traffic entering the outside interface.

Monitoring Access Rules

To monitor network access, enter the following command:

Command	Purpose
show running-config access-group	Displays the current ACL bound to the interfaces.

Configuration Examples for Permitting or Denying Network Access

This section includes typical configuration examples for permitting or denying network access.

The following example adds a network object for inside server 1, performs static NAT for the server, and enables access to from the outside for inside server 1.

```
ciscoasa(config)# object network inside-server1
ciscoasa(config)# host 10.1.1.1
ciscoasa(config)# nat (inside,outside) static 209.165.201.12

ciscoasa(config)# access-list outside_access extended permit tcp any object inside-server1
eq www
ciscoasa(config)# access-group outside_access in interface outside
```

The following example allows all hosts to communicate between the **inside** and **hr** networks but only specific hosts to access the outside network:

```
ciscoasa(config)# access-list ANY extended permit ip any any
ciscoasa(config)# access-list OUT extended permit ip host 209.168.200.3 any
ciscoasa(config)# access-list OUT extended permit ip host 209.168.200.4 any

ciscoasa(config)# access-group ANY in interface inside
ciscoasa(config)# access-group ANY in interface hr
ciscoasa(config)# access-group OUT out interface outside
```

For example, the following sample ACL allows common EtherTypes originating on the inside interface:

```
ciscoasa(config)# access-list ETHER ethertype permit ipx
ciscoasa(config)# access-list ETHER ethertype permit mpls-unicast
ciscoasa(config)# access-group ETHER in interface inside
```

The following example allows some EtherTypes through the ASA, but it denies all others:

```
ciscoasa(config)# access-list ETHER ethertype permit 0x1234
ciscoasa(config)# access-list ETHER ethertype permit mpls-unicast
ciscoasa(config)# access-group ETHER in interface inside
ciscoasa(config)# access-group ETHER in interface outside
```

The following example denies traffic with EtherType 0x1256 but allows all others on both interfaces:

```
ciscoasa(config)# access-list nonIP ethertype deny 1256
ciscoasa(config)# access-list nonIP ethertype permit any
ciscoasa(config)# access-group ETHER in interface inside
ciscoasa(config)# access-group ETHER in interface outside
```

The following example uses object groups to permit specific traffic on the inside interface:

```
!
hostname (config)# object-group service myaclog
```

```
hostname (config-service)# service-object tcp source range 2000 3000
hostname (config-service)# service-object tcp source range 3000 3010 destinatio$
hostname (config-service)# service-object ipsec
hostname (config-service)# service-object udp destination range 1002 1006
hostname (config-service)# service-object icmp echo

ciscoasa(config)# access-list outsideacl extended permit object-group myaclog interface
inside any
```

Feature History for Access Rules

Table 6-2 lists each feature change and the platform release in which it was implemented.

Table 6-2 Feature History for Access Rules

Feature Name	Platform Releases	Feature Information
Interface access rules	7.0(1)	Controlling network access through the ASA using ACLs. We introduced the following command: **access-group**.
Global access rules	8.3(1)	Global access rules were introduced. We modified the following command: **access-group.**
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**.
EtherType ACL support for IS-IS traffic	8.4(5), 9.1(2)	In transparent firewall mode, the ASA can now pass IS-IS traffic using an EtherType ACL. We modified the following command: **access-list ethertype** {**permit** | **deny**} **is-is**.
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 6-2 Feature History for Access Rules (continued)

Feature Name	Platform 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**.
Transactional Commit Model on Rule Engine for Access groups	9.1(5)	When enabled, a rule update is applied after the rule compilation is completed; without affecting the rule matching performance. We introduced the following commands: **asp rule-engine transactional-commit**, **show running-config asp rule-engine transactional-commit**, **clear configure asp rule-engine transactional-commit**.