



CHAPTER 11

Permitting or Denying Network Access

This chapter describes how to control network access through the FWSM using access lists. To create an extended access lists or an EtherType access list, see [Chapter 10, “Identifying Traffic with Access Lists.”](#)



Note

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

To access the FWSM interface for management access, you do not also need an access list allowing the host IP address. You only need to configure management access according to [Chapter 21, “Configuring Management Access.”](#)

This chapter includes the following sections:

- [Inbound and Outbound Access List Overview, page 11-1](#)
- [Applying an Access List to an Interface, page 11-4](#)

Inbound and Outbound Access List Overview

Traffic flowing across an interface in the FWSM can be controlled in two ways. Traffic that enters the FWSM can be controlled by attaching an inbound access list to the source interface. Traffic that exits the FWSM can be controlled by attaching an outbound access list to the destination interface. To allow any traffic to enter the FWSM, you must attach an inbound access list to an interface; otherwise, the FWSM automatically drops all traffic that enters that interface. By default, traffic can exit the FWSM on any interface unless you restrict it using an outbound access list, which adds restrictions to those already configured in the inbound access list.

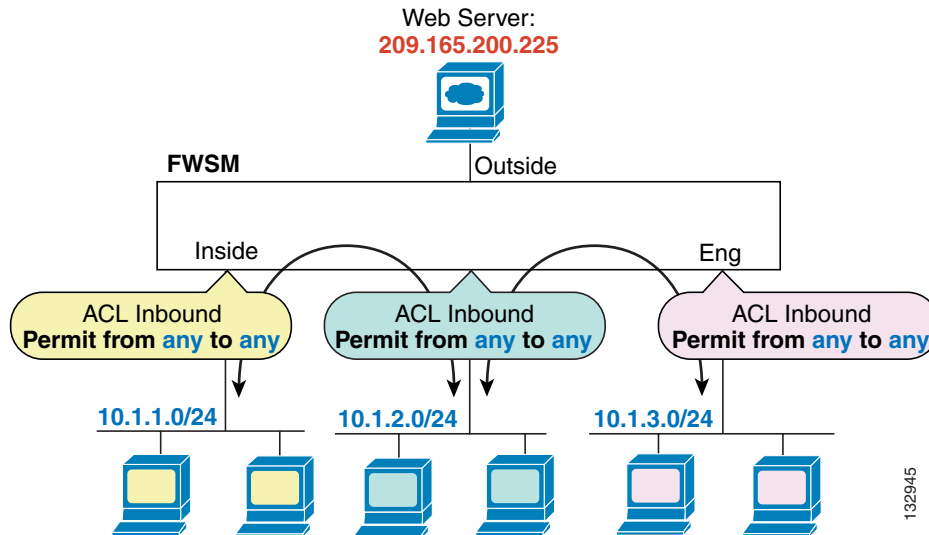


Note

“Inbound” and “outbound” refer to the application of an access list on an interface, either to traffic entering the FWSM on an interface or traffic exiting the FWSM 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.

You might want to use an outbound access list to simplify your access list configuration. For example, if you want to allow three inside networks on three different interfaces to access each other, you can create a simple inbound access list that allows all traffic on each inside interface (see [Figure 11-1](#)).

Figure 11-1 Inbound Access Lists



See the following commands for this example:

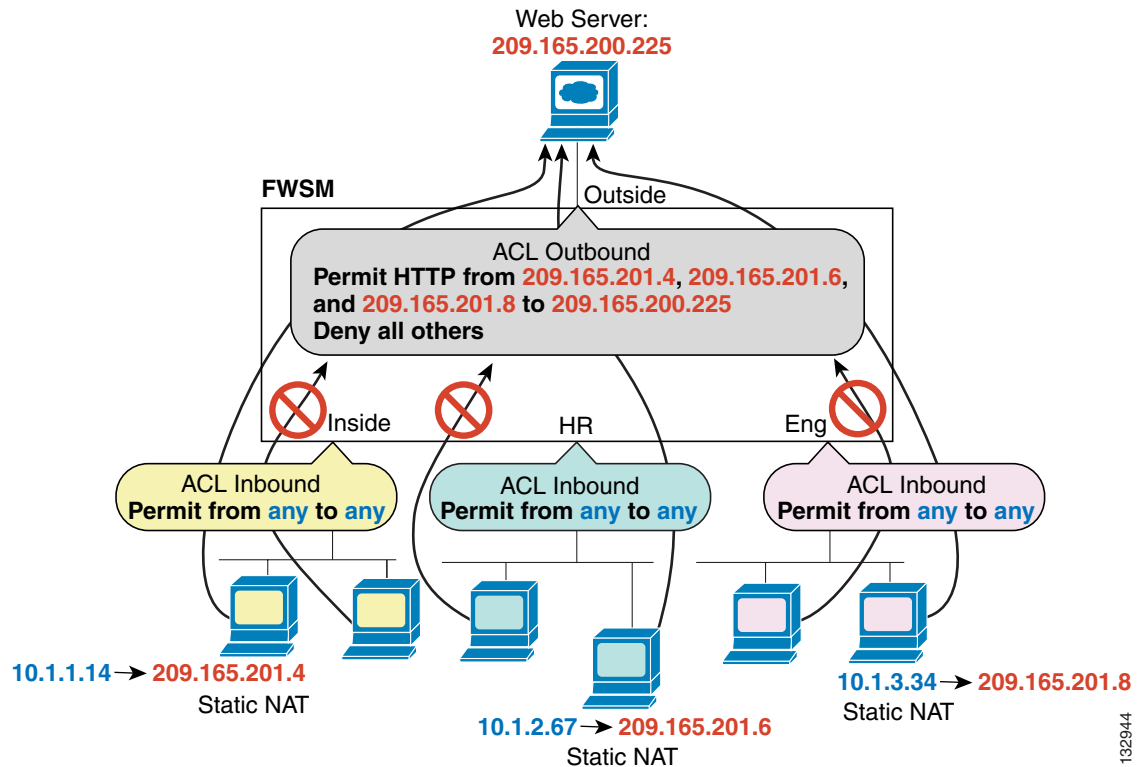
```
hostname(config)# access-list INSIDE extended permit ip any any
hostname(config)# access-group INSIDE in interface inside

hostname(config)# access-list HR extended permit ip any any
hostname(config)# access-group HR in interface hr

hostname(config)# access-list ENG extended permit ip any any
hostname(config)# access-group ENG in interface eng
```

Then, if you want to allow only certain hosts on the inside networks to access a web server on the outside network, you can create a more restrictive access list that allows only the specified hosts and apply it to the outbound direction of the outside interface (see Figure 11-1). See the “IP Addresses Used for Access Lists When You Use NAT” section on page 10-3 for information about NAT and IP addresses. The outbound access list prevents any other hosts from reaching the outside network.

Figure 11-2 Outbound Access List



See the following commands for this example:

```
hostname(config)# access-list INSIDE extended permit ip any any
hostname(config)# access-group INSIDE in interface inside

hostname(config)# access-list HR extended permit ip any any
hostname(config)# access-group HR in interface hr

hostname(config)# access-list ENG extended permit ip any any
hostname(config)# access-group ENG in interface eng

hostname(config)# access-list OUTSIDE extended permit tcp host 209.165.201.4
host 209.165.200.225 eq www
hostname(config)# access-list OUTSIDE extended permit tcp host 209.165.201.6
host 209.165.200.225 eq www
hostname(config)# access-list OUTSIDE extended permit tcp host 209.165.201.8
host 209.165.200.225 eq www
hostname(config)# access-group OUTSIDE out interface outside
```

Applying an Access List to an Interface

To apply an extended access list to the inbound or outbound direction of an interface, enter the following command:

```
hostname(config)# access-group access_list_name {in | out} interface interface_name
[per-user-override]
```

You can apply one access list of each type (extended and EtherType) to both directions of the interface. See the [“Inbound and Outbound Access List Overview”](#) section on page 11-1 for more information about access list directions.

The **per-user-override** keyword allows dynamic access lists that are downloaded for user authorization to override the access list assigned to the interface. For example, if the interface access list denies all traffic from 10.0.0.0, but the dynamic access list permits all traffic from 10.0.0.0, then the dynamic access list overrides the interface access list for that user. See the [“Configuring RADIUS Authorization”](#) section for more information about per-user access lists. The **per-user-override** keyword is only available for inbound access lists.

For connectionless protocols, you need to apply the access list to the source and destination interfaces if you want traffic to pass in both directions. For example, you can allow BGP in an EtherType access list in transparent mode, and you need to apply the access list to both interfaces.

The following example illustrates the commands required to enable access to an inside web server with the IP address 209.165.201.12 (this IP address is the address visible on the outside interface after NAT):

```
hostname(config)# access-list ACL_OUT extended permit tcp any host 209.165.201.12 eq www
hostname(config)# access-group ACL_OUT in interface outside
```

You also need to configure NAT for the web server.

The following access lists allow all hosts to communicate between the inside and hr networks, but only specific hosts to access the outside network:

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

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

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

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

The following access list allows some EtherTypes through the FWSM, but denies all others:

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

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

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