



Configuring NAT

This chapter describes Network Address Translation. In routed firewall mode, the FWSM can perform NAT between each network.



Note

In transparent firewall mode, the FWSM does not support NAT except to set connection limits. See the “[Configuring Connection Limits for Transparent Firewall Mode and Non-NAT Configurations](#)” section on page 7-5.

This chapter contains the following sections:

- [NAT Overview, page 12-1](#)
- [Configuring NAT Control, page 12-16](#)
- [Using Dynamic NAT and PAT, page 12-17](#)
- [Using Static NAT, page 12-26](#)
- [Using Static PAT, page 12-28](#)
- [Bypassing NAT, page 12-30](#)
- [NAT Examples, page 12-34](#)

NAT Overview

This section describes how NAT works on the FWSM, and includes the following topics:

- [Introduction to NAT, page 12-2](#)
- [NAT Control, page 12-3](#)
- [NAT Types, page 12-5](#)
- [Policy NAT, page 12-9](#)
- [NAT and Same Security Level Interfaces, page 12-12](#)
- [Order of NAT Commands Used to Match Real Addresses, page 12-13](#)
- [Maximum Number of NAT Statements, page 12-13](#)
- [Mapped Address Guidelines, page 12-13](#)
- [DNS and NAT, page 12-14](#)

Introduction to NAT

Address translation substitutes the real address in a packet with a mapped address that is routable on the destination network. NAT is comprised of two steps: the process in which a real address is translated into a mapped address, and then the process to undo translation for returning traffic.

The FWSM translates an address when a NAT rule matches the traffic. If no NAT rule matches, processing for the packet continues. The exception is when you enable NAT control. NAT control requires that packets traversing from a higher security interface (inside) to a lower security interface (outside) match a NAT rule, or else processing for the packet stops. (See the [“Security Level Overview” section on page 6-1](#) for more information about security levels, and see [“NAT Control” section on page 12-3](#) for more information about NAT control).

**Note**

In this document, all types of translation are generally referred to as NAT. When discussing NAT, the terms *inside* and *outside* are relative, and represent the security relationship between any two interfaces. The higher security level is inside and the lower security level is outside; for example, interface 1 is at 60 and interface 2 is at 50, so interface 1 is “inside” and interface 2 is “outside.”

Some of the benefits of NAT are as follows:

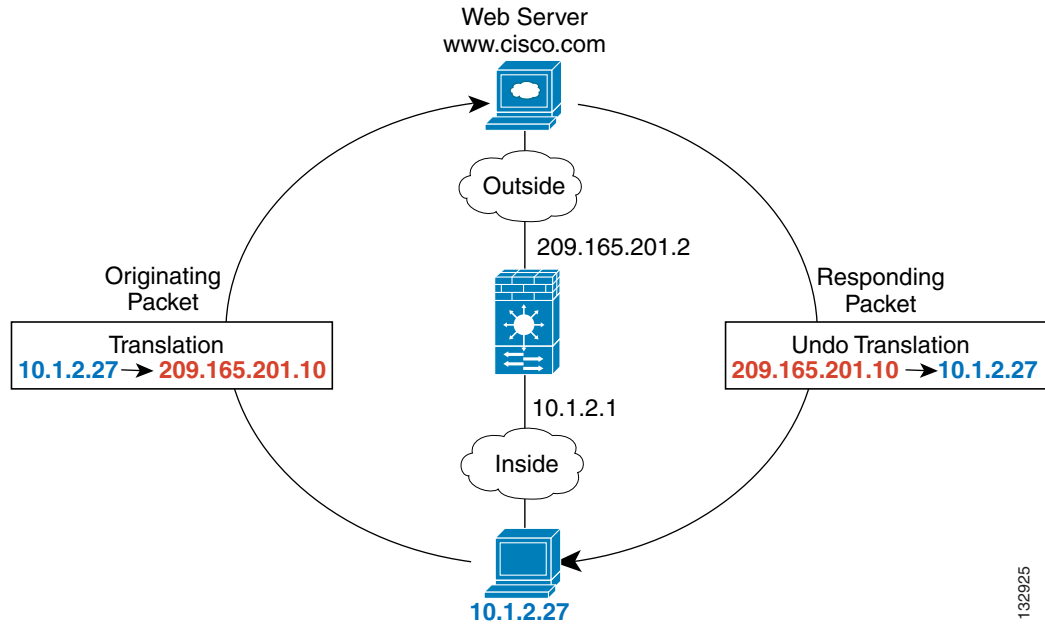
- You can use private addresses on your inside networks. Private addresses are not routable on the Internet. (See the [“Private Networks” section on page D-2](#) for more information.)
- NAT hides the real addresses from other networks, so attackers cannot learn the real address of a host.
- You can resolve IP routing problems such as overlapping addresses.

**Note**

See [Table 20-1 on page 20-5](#) for information about protocols that do not support NAT.

[Figure 12-1](#) shows a typical NAT scenario, with a private network on the inside. When the inside host at 10.1.2.27 sends a packet to a web server, the real source address, 10.1.2.27, of the packet is changed to a mapped address, 209.165.201.10. When the server responds, it sends the response to the mapped address, 209.165.201.10, and the FWSM receives the packet. The FWSM then undoes the translation of the mapped address, 209.165.201.10 back to the real address, 10.1.2.27 before sending it on to the host.

Figure 12-1 NAT Example



132925

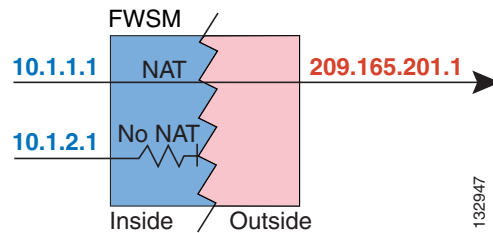
See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.1-209.165.201.15
```

NAT Control

NAT control requires that packets traversing from an inside interface to an outside interface match a NAT rule; for any host on the inside network to access a host on the outside network, you must configure NAT to translate the inside host address (see Figure 12-2).

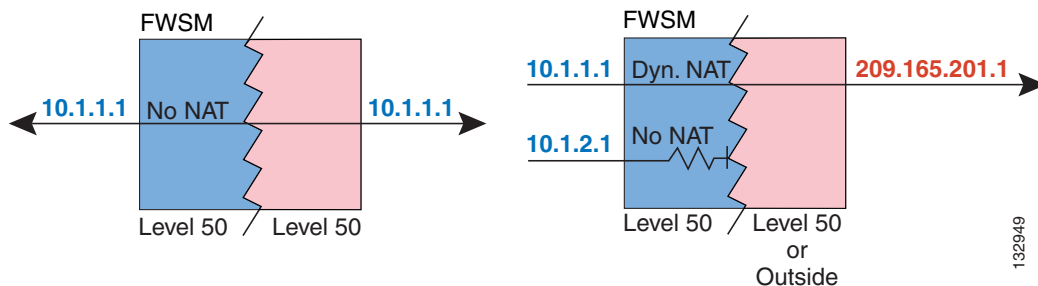
Figure 12-2 NAT Control and Outbound Traffic



132947

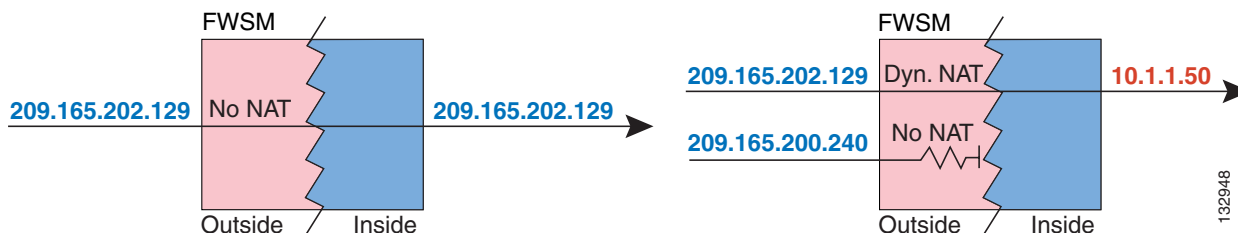
Interfaces at the same security level are not required to use NAT to communicate. However, if you configure dynamic NAT or PAT on a same security interface with NAT control enabled, then all traffic from the interface to a same security interface or an outside interface must match a NAT rule (see Figure 12-3).

Figure 12-3 NAT Control and Same Security Traffic



Similarly, if you enable outside dynamic NAT or PAT with NAT control, then all outside traffic must match a NAT rule when it accesses an inside interface (see Figure 12-4).

Figure 12-4 NAT Control and Inbound Traffic



Static NAT with NAT control does not cause these restrictions.

By default, NAT control is disabled, so you do not need to perform NAT on any networks unless you choose to perform NAT. If you upgraded from an earlier version of software, however, NAT control might be enabled on your system.



Note

Even if you do not configure NAT, the FWSM continues to create translation sessions for all traffic automatically. In this case, the translation is from the real address to the same real address. See the **show xlate** command to view translation sessions.

If you want the added security of NAT control but do not want to translate inside addresses in some cases, you can apply a NAT exemption or identity NAT rule on those addresses. (See the **“Bypassing NAT”** section on page 12-30 for more information).

To configure NAT control, see the **“Configuring NAT Control”** section on page 12-16.



Note

In multiple context mode, the packet classifier relies on the NAT configuration in some cases to assign packets to contexts. If you do not perform NAT because NAT control is disabled, then the classifier might require changes in your network configuration. See the **“How the FWSM Classifies Packets”** section on page 4-3 for more information about the relationship between the classifier and NAT.

NAT Types

This section describes the available NAT types. You can implement address translation as dynamic NAT, Port Address Translation (PAT; also known as NAT overloading), static NAT, or static PAT or as a mix of these types. You can also configure rules to bypass NAT, for example, if you enable NAT control but do not want to perform NAT. This section includes the following topics:

- [Dynamic NAT, page 12-5](#)
- [PAT, page 12-7](#)
- [Static NAT, page 12-7](#)
- [Static PAT, page 12-7](#)
- [Bypassing NAT when NAT Control is Enabled, page 12-8](#)

Dynamic NAT

Dynamic NAT translates a group of real addresses to a pool of mapped addresses that are routable on the destination network. The mapped pool can include fewer addresses than the real group. When a host you want to translate accesses the destination network, the FWSM assigns it an IP address from the mapped pool. The translation is added only when the real host initiates the connection. The translation is in place only for the duration of the connection, and a given user does not keep the same IP address after the translation times out (see the **timeout xlate** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*). Users on the destination network, therefore, cannot reliably initiate a connection to a host that uses dynamic NAT (even if the connection is allowed by an access list), and the FWSM rejects any attempt to connect to a real host address directly. See the following “[Static NAT](#)” or “[Static PAT](#)” sections for reliable access to hosts.

[Figure 12-5](#) shows a remote host attempting to connect to the real address. The connection is denied because the FWSM only allows returning connections to the mapped address.

Figure 12-5 Remote Host Attempts to Connect to the Real Address

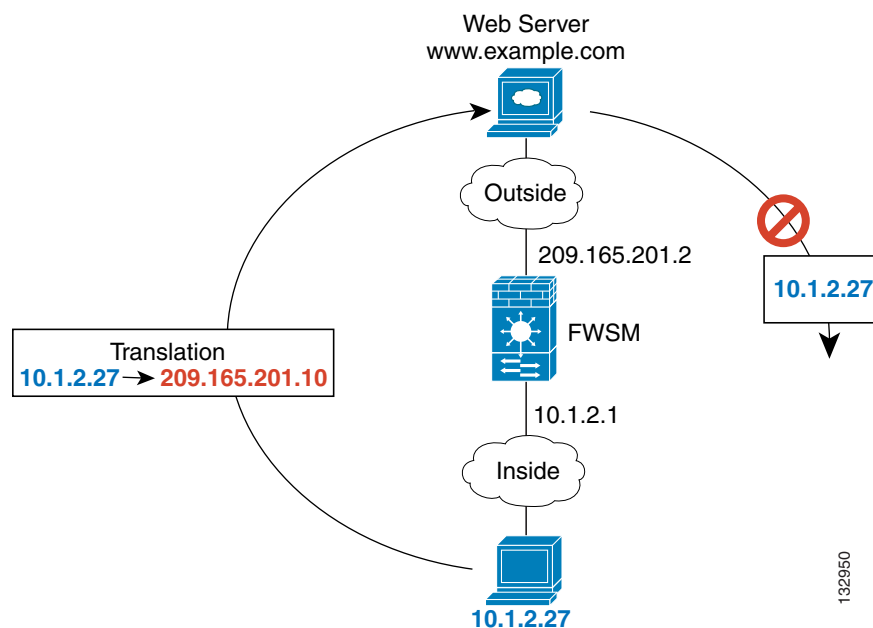
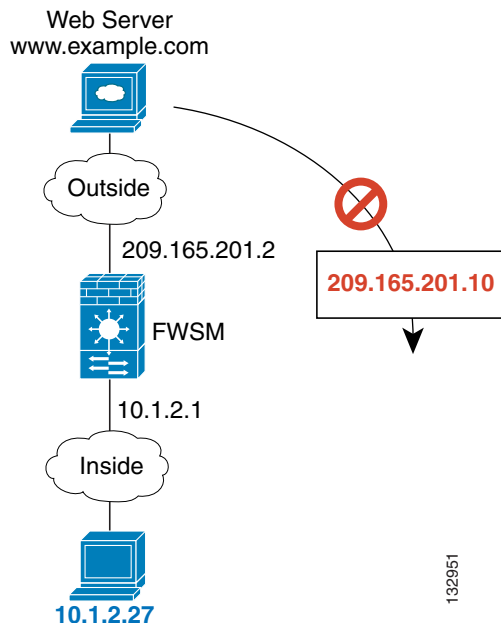


Figure 12-6 shows a remote host attempting to initiate a connection to a mapped address. This address is not currently in the translation table, so the FWSM drops the packet.

Figure 12-6 Remote Host Attempts to Initiate a Connection to a Mapped Address



Note

For the duration of the translation, a remote host can initiate a connection to the translated host if an access list allows it. Because the address is unpredictable, a connection to the host is unlikely. However in this case, you can rely on the security of the access list.

Dynamic NAT has these disadvantages:

- If the mapped pool has fewer addresses than the real group, you could run out of addresses if the amount of traffic is more than expected.
Use PAT if this event occurs often, because PAT provides over 64,000 translations using ports of a single address.
- You have to use a large number of routable addresses in the mapped pool; if the destination network requires registered addresses, such as the Internet, you might encounter a shortage of usable addresses.

The advantage of dynamic NAT is that some protocols cannot use PAT. For example, PAT does not work with IP protocols that do not have a port to overload, such as GRE version 0. PAT also does not work with some applications that have a data stream on one port and the control path on another and are not open standard, such as some multimedia applications. See the [“Application Inspection Engine Overview” section on page 20-2](#) for more information about NAT and PAT support.

PAT

PAT (also known as NAT overloading) translates multiple real addresses to a single mapped IP address. Specifically, the FWSM translates the real address and source port (real socket) to the mapped address and a unique port above 1024 (mapped socket). Each connection requires a separate translation, because the source port differs for each connection. For example, 10.1.1.1:1025 requires a separate translation from 10.1.1.1:1026.

After the connection expires, the port translation also expires after 30 seconds of inactivity. The timeout is not configurable. Users on the destination network cannot reliably initiate a connection to a host that uses PAT (even if the connection is allowed by an access list). Not only can you not predict the real or mapped port number of the host, but the FWSM does not create a translation at all unless the translated host is the initiator. See the following “[Static NAT](#)” or “[Static PAT](#)” sections for reliable access to hosts.

PAT lets you use a single mapped address, thus conserving routable addresses. You can even use the FWSM interface IP address as the PAT address. PAT does not work with some multimedia applications that have a data stream that is different from the control path. See the “[Application Inspection Engine Overview](#)” section on page 20-2 for more information about NAT and PAT support.

**Note**

For the duration of the translation, a remote host can initiate a connection to the translated host if an access list allows it. Because the port address (both real and mapped) is unpredictable, a connection to the host is unlikely. However in this case, you can rely on the security of the access list.

Static NAT

Static NAT creates a fixed translation of real address(es) to mapped address(es). With dynamic NAT and PAT, each host uses a different address or port for each subsequent translation. Because the mapped address is the same for each consecutive connection with static NAT, and a persistent translation rule exists, static NAT allows hosts on the destination network to initiate traffic to a translated host (if there is an access list that allows it).

The main difference between dynamic NAT and a range of addresses for static NAT is that static NAT allows a remote host to initiate a connection to a translated host (if there is an access list that allows it), while dynamic NAT does not. You also need an equal number of mapped addresses as real addresses with static NAT.

Static PAT

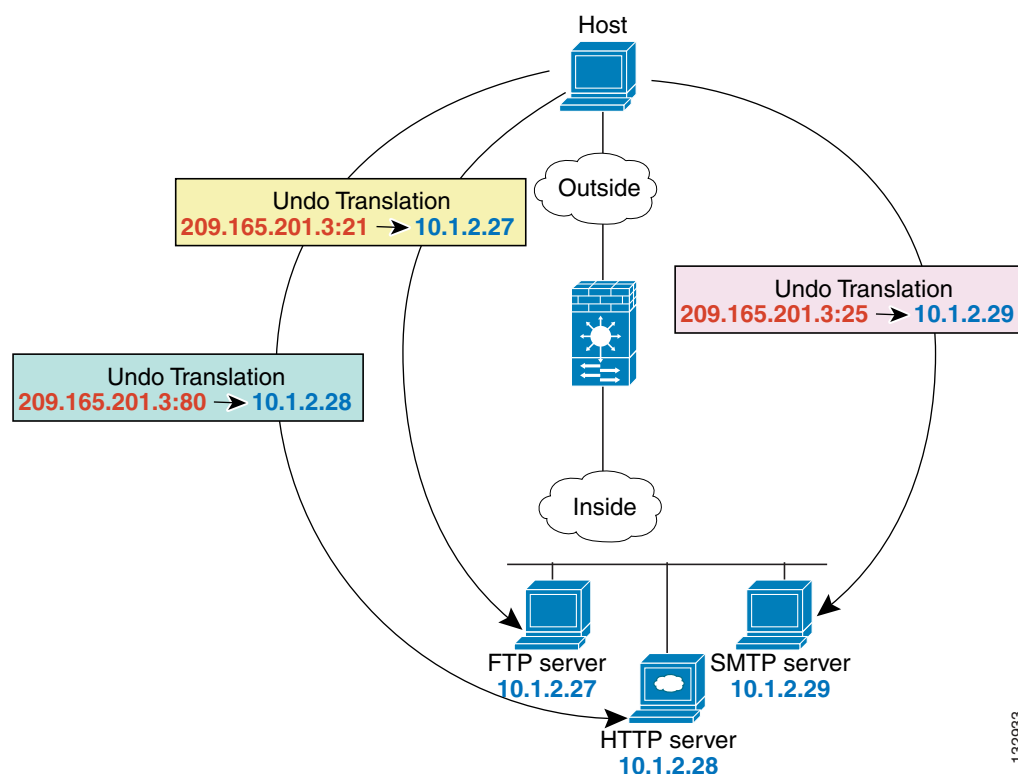
Static PAT is the same as static NAT, except it lets you specify the protocol (TCP or UDP) and port for the real and mapped addresses.

This feature lets you identify the same mapped address across many different static statements, so long as the port is different for each statement. However, you cannot use the same mapped address for multiple static NAT statements.

For applications that require application inspection for secondary channels (FTP, VoIP, etc.), the FWSM automatically translates the secondary ports.

For example, if you want to provide a single address for remote users to access FTP, HTTP, and SMTP, but these are all actually different servers on the real network, you can specify static PAT statements for each server that uses the same mapped IP address, but different ports (see Figure 12-7).

Figure 12-7 Static PAT



See the following commands for this example:

```
hostname(config)# static (inside,outside) tcp 209.165.201.3 ftp 10.1.2.27 ftp netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 209.165.201.3 http 10.1.2.28 http netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 209.165.201.3 smtp 10.1.2.29 smtp netmask
255.255.255.255
```

You can also use static PAT to translate a well-known port to a non-standard port or vice versa. For example, if your inside web servers use port 8080, you can allow outside users to connect to port 80, and then undo translation to the original port 8080. Similarly, if you want to provide extra security, you can tell your web users to connect to non-standard port 6785, and then undo translation to port 80.

Bypassing NAT when NAT Control is Enabled

If you enable NAT control, then inside hosts must match a NAT rule when accessing outside hosts. If you do not want to perform NAT for some hosts, then you can bypass NAT for those hosts (alternatively, you can disable NAT control). You might want to bypass NAT, for example, if you are using an application that does not support NAT (see the “[Application Inspection Engine Overview](#)” section on page 20-2 for information about inspection engines that do not support NAT).

You can configure traffic to bypass NAT using one of three methods. All methods achieve compatibility with inspection engines. However, each method offers slightly different capabilities, as follows:

- Identity NAT (**nat 0** command)—When you configure identity NAT (which is similar to dynamic NAT), you do not limit translation for a host on specific interfaces; you must use identity NAT for connections through all interfaces. Therefore, you cannot choose to perform normal translation on real addresses when you access interface A, but use identity NAT when accessing interface B. Regular dynamic NAT, on the other hand, lets you specify a particular interface on which to translate the addresses. Make sure that the real addresses for which you use identity NAT are routable on all networks that are available according to your access lists.

For identity NAT, even though the mapped address is the same as the real address, you cannot initiate a connection from the outside to the inside (even if the interface access list allows it). Use static identity NAT or NAT exemption for this functionality.

- Static identity NAT (**static** command)—Static identity NAT lets you specify the interface on which you want to allow the real addresses to appear, so you can use identity NAT when you access interface A, and use regular translation when you access interface B. Static identity NAT also lets you use policy NAT, which identifies the real and destination addresses when determining the real addresses to translate (see the “Policy NAT” section on page 12-9 for more information about policy NAT). For example, you can use static identity NAT for an inside address when it accesses the outside interface and the destination is server A, but use a normal translation when accessing the outside server B.
- NAT exemption (**nat 0 access-list** command)—NAT exemption allows both translated and remote hosts to initiate connections. Like identity NAT, you do not limit translation for a host on specific interfaces; you must use NAT exemption for connections through all interfaces. However, NAT exemption does let you specify the real and destination addresses when determining the real addresses to translate (similar to policy NAT), so you have greater control using NAT exemption. However unlike policy NAT, NAT exemption does not consider the ports in the access list.

Policy NAT

Policy NAT lets you identify real addresses for address translation by specifying the source and destination addresses in an extended access list. You can also optionally specify the source and destination ports. Regular NAT can only consider the real addresses. For example, you can use translate the real address to mapped address A when it accesses server A, but translate the real address to mapped address B when it accesses server B.

For applications that require application inspection for secondary channels (FTP, VoIP, etc.), the policy specified in the policy NAT statement should include the secondary ports. Or, when the ports cannot be predicted, the policy should specify only the IP addresses for the secondary channel. This way, the FWSM translates the secondary ports.

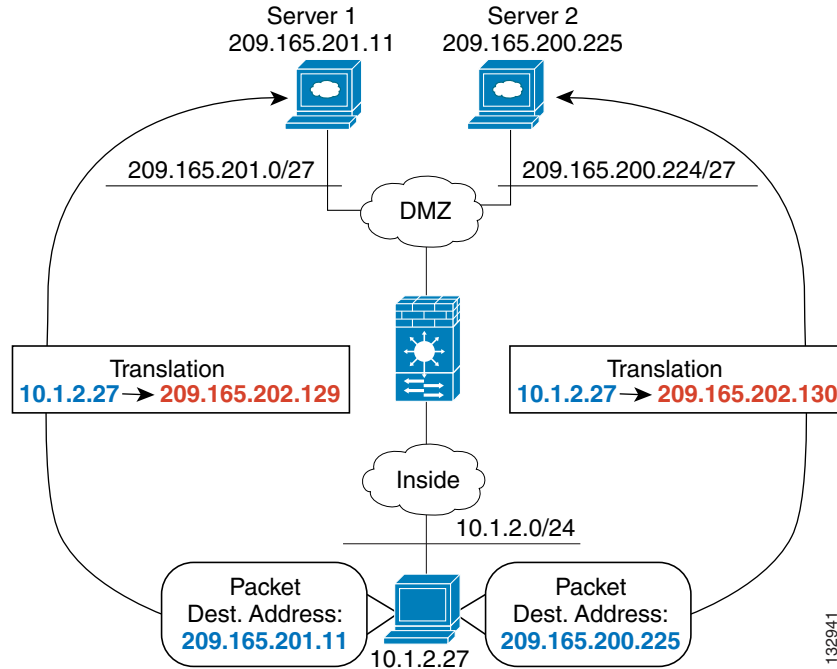


Note

All types of NAT support policy NAT except for NAT exemption. NAT exemption uses an access list to identify the real addresses, but differs from policy NAT in that the ports are not considered. See the “Bypassing NAT” section on page 12-30 for other differences. You can accomplish the same result as NAT exemption using static identity NAT, which does support policy NAT.

Figure 12-8 shows a host on the 10.1.2.0/24 network accessing two different servers. When the host accesses the server at 209.165.201.11, the real address is translated to 209.165.202.129. When the host accesses the server at 209.165.200.225, the real address is translated to 209.165.202.130 so that the host appears to be on the same network as the servers, which can help with routing.

Figure 12-8 Policy NAT with Different Destination Addresses

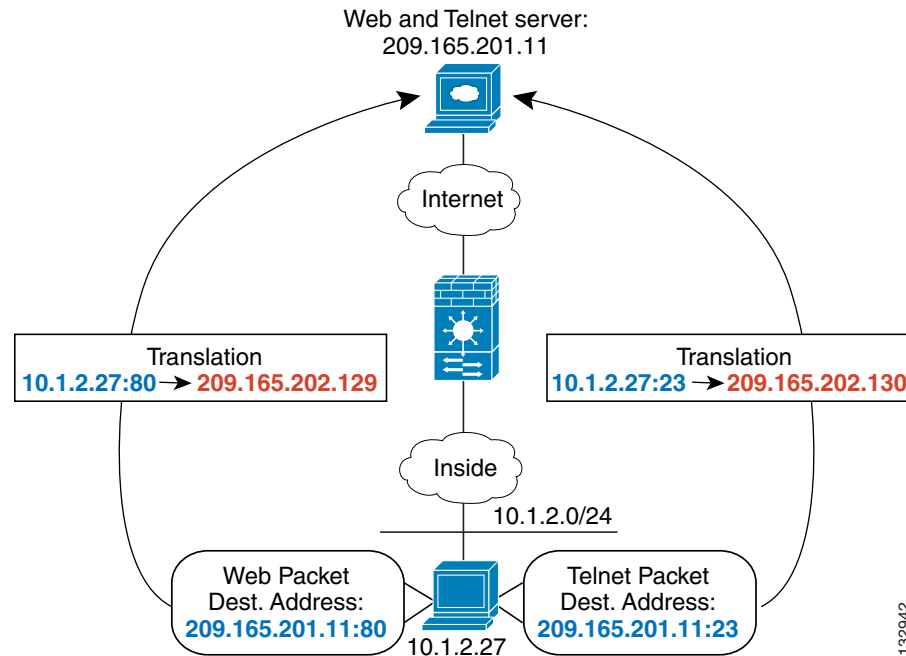


See the following commands for this example:

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
hostname(config)# access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
hostname(config)# nat (inside) 1 access-list NET1
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list NET2
hostname(config)# global (outside) 2 209.165.202.130
```

Figure 12-9 shows the use of source and destination ports. The host on the 10.1.2.0/24 network accesses a single host for both web services and Telnet services. When the host accesses the server for web services, the real address is translated to 209.165.202.129. When the host accesses the same server for Telnet services, the real address is translated to 209.165.202.130.

Figure 12-9 Policy NAT with Different Destination Ports



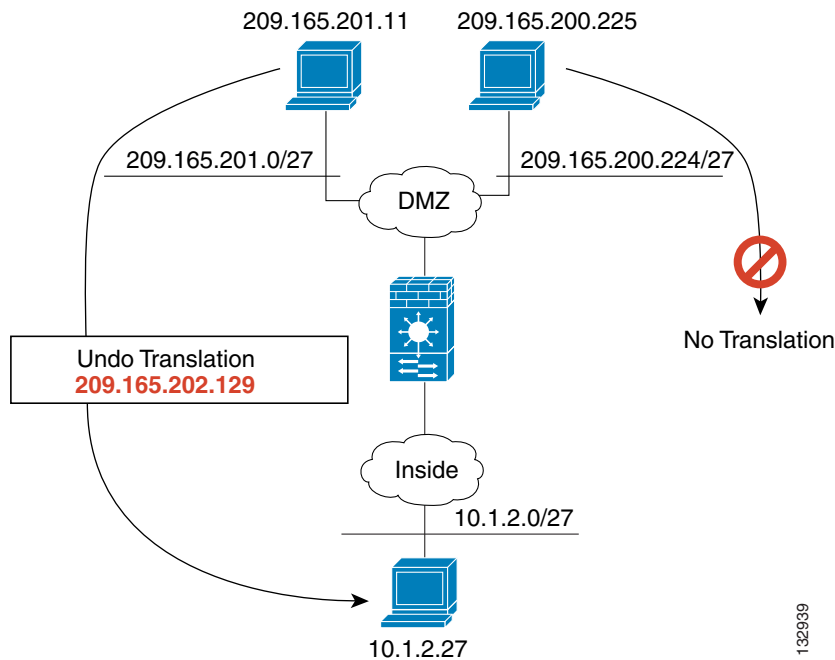
See the following commands for this example:

```
hostname(config)# access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 80
hostname(config)# access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 23
hostname(config)# nat (inside) 1 access-list WEB
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list TELNET
hostname(config)# global (outside) 2 209.165.202.130
```

For policy static NAT (and for NAT exemption, which also uses an access list to identify traffic), both translated and remote hosts can originate traffic. For traffic originated on the translated network, the NAT access list specifies the real addresses and the *destination* addresses, but for traffic originated on the remote network, the access list identifies the real addresses and the *source* addresses of remote hosts who are allowed to connect to the host using this translation.

Figure 12-10 shows a remote host connecting to a translated host. The translated host has a policy static NAT translation that translates the real address only for traffic to and from the 209.165.201.0/27 network. A translation does not exist for the 209.165.200.224/27 network, so the translated host cannot connect to that network, nor can a host on that network connect to the translated host.

Figure 12-10 Policy Static NAT with Destination Address Translation



See the following commands for this example:

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.224 209.165.201.0
255.255.255.224
hostname(config)# static (inside,outside) 209.165.202.129 access-list NET1
```



Note

For policy static NAT, in undoing the translation, the ACL in the **static** command is not used. If the destination address in the packet matches the mapped address in the static rule, the static rule is used to untranslate the address.



Note

Policy NAT does not support SQL*Net, but it is supported by regular NAT. See the [“Application Inspection Engine Overview”](#) section on page 20-2 for information about NAT support for other protocols.

NAT and Same Security Level Interfaces

NAT is not required between same security level interfaces even if you enable NAT control. You can optionally configure NAT if desired.

See the [“Allowing Communication Between Interfaces on the Same Security Level”](#) section on page 6-5 to enable same security communication.

**Note**

The FWSM does not support VoIP inspection engines when you configure NAT on same security interfaces. These inspection engines include Skinny, SIP, and H.323. See the [“Application Inspection Engine Overview”](#) section on page 20-2 for supported inspection engines.

Order of NAT Commands Used to Match Real Addresses

The FWSM matches real addresses to NAT commands in the following order:

1. NAT exemption (**nat 0 access-list**)—In order, until the first match. Identity NAT is not included in this category; it is included in the regular static NAT or regular NAT category. We do not recommend overlapping addresses in NAT exemption statements because unexpected results can occur.
2. Static NAT and Static PAT (regular and policy) (**static**)—Best match. Static identity NAT is included in this category. In the case of overlapping addresses in **static** statements, a warning will be displayed, but they are supported. The order of the **static** commands does not matter; the **static** statement that best matches the real address is used.
3. Policy dynamic NAT (**nat access-list**)—In order, until the first match. Overlapping addresses are allowed.
4. Regular dynamic NAT (**nat**)—Best match. Regular identity NAT is included in this category. The order of the NAT commands does not matter; the NAT statement that best matches the real address is used. For example, you can create a general statement to translate all addresses (0.0.0.0) on an interface. If you want to translate a subset of your network (10.1.1.1) to a different address, then you can create a statement to translate only 10.1.1.1. When 10.1.1.1 makes a connection, the specific statement for 10.1.1.1 is used because it matches the real address best. We do not recommend using overlapping statements; they use more memory and can slow the performance of the FWSM.

Maximum Number of NAT Statements

The FWSM supports the following numbers of **nat**, **global**, and **static** commands divided between all contexts or in single mode:

- **nat** command—2 K
- **global** command—4 K
- **static** command—2 K

The FWSM also supports up to 3942 ACEs in access lists used for policy NAT for single mode, and 7272 ACEs for multiple mode.

Mapped Address Guidelines

When you translate the real address to a mapped address, you can use the following mapped addresses:

- Addresses on the same network as the mapped interface.

If you use addresses on the same network as the mapped interface (through which traffic exits the FWSM), the FWSM uses proxy ARP to answer any requests for mapped addresses, and thus intercepts traffic destined for a real address. This solution simplifies routing, because the FWSM does not have to be the gateway for any additional networks. However, this approach does put a limit on the number of available addresses used for translations.

For PAT, you can even use the IP address of the mapped interface.

- Addresses on a unique network.

If you need more addresses than are available on the mapped interface network, you can identify addresses on a different subnet. The FWSM uses proxy ARP to answer any requests for mapped addresses, and thus intercepts traffic destined for a real address. If you use OSPF, and you advertise routes on the mapped interface, then the FWSM advertises the mapped addresses. If the mapped interface is passive (not advertising routes) or you are using static routing, then you need to add a static route on the upstream router that sends traffic destined for the mapped addresses to the FWSM.

DNS and NAT

You might need to configure the FWSM to modify DNS replies by replacing the address in the reply with an address that matches the NAT configuration. You can configure DNS modification when you configure each translation.

For example, a DNS server is accessible from the outside interface. A server, ftp.example.com, is on the inside interface. You configure the FWSM to statically translate the ftp.example.com real address (10.1.3.14) to a mapped address (209.165.201.10) that is visible on the outside network (see [Figure 12-11](#)). In this case, you want to enable DNS reply modification on this static statement so that inside users who have access to ftp.example.com using the real address receive the real address from the DNS server, and not the mapped address.

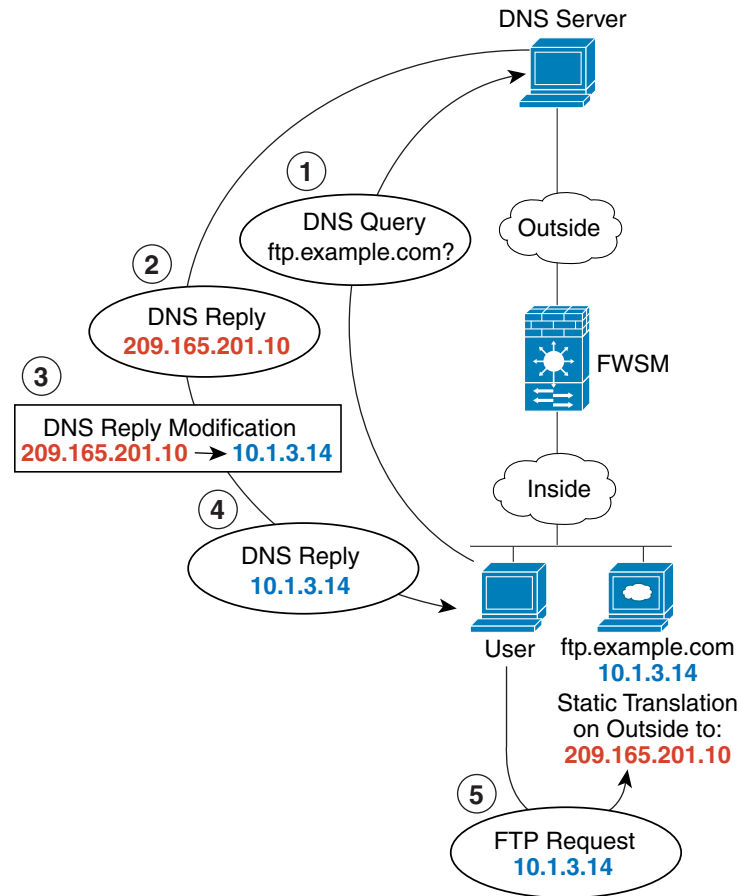
When an inside host sends a DNS request for the address of ftp.example.com, the DNS server replies with the mapped address (209.165.201.10). The FWSM refers to the static statement for the inside server and translates the address inside the DNS reply to 10.1.3.14. If you do not enable DNS reply modification, then the inside host attempts to send traffic to 209.165.201.10 instead of accessing ftp.example.com directly.



Note

A route needs to exist for the real IP address embedded in the DNS query response or the FWSM will not NAT it. The necessary route can be learned via static routing or by any other routing protocol, such as RIP or OSPF.

Figure 12-11 DNS Reply Modification



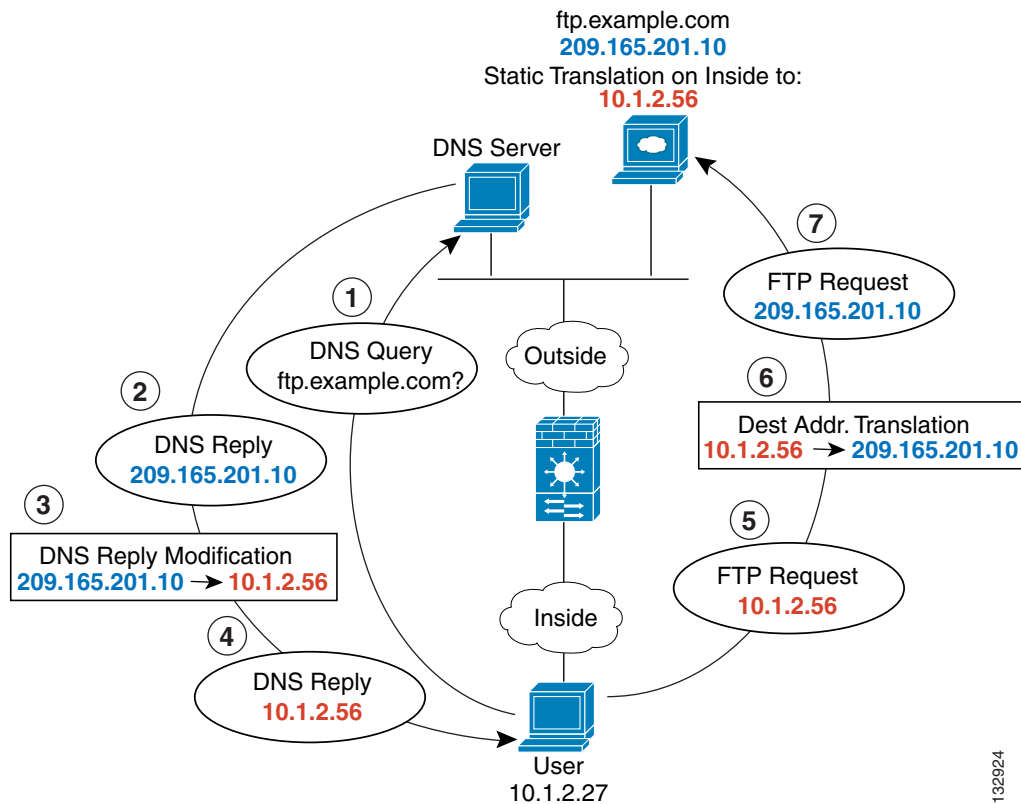
132946

See the following command for this example:

```
hostname(config)# static (inside,outside) 209.165.201.10 10.1.3.14 netmask 255.255.255.255
dns
```

Figure 12-12 shows a web server and DNS server on the outside. The FWSM has a static translation for the outside server. In this case, when an inside user requests the address for ftp.example.com from the DNS server, the DNS server responds with the real address, 209.165.201.10. Because you want inside users to use the mapped address for ftp.example.com (10.1.2.56) you need to configure DNS reply modification for the static translation.

Figure 12-12 DNS Reply Modification Using Outside NAT



See the following command for this example:

```
hostname(config)# static (outside,inside) 10.1.2.56 209.165.201.10 netmask 255.255.255.255 dns
```

Configuring NAT Control

NAT control requires that packets traversing from an inside interface to an outside interface match a NAT rule. See the “NAT Control” section on page 12-3 for more information.

To enable NAT control, enter the following command:

```
hostname(config)# nat-control
```

To disable NAT control, enter the **no** form of the command.

Using Dynamic NAT and PAT

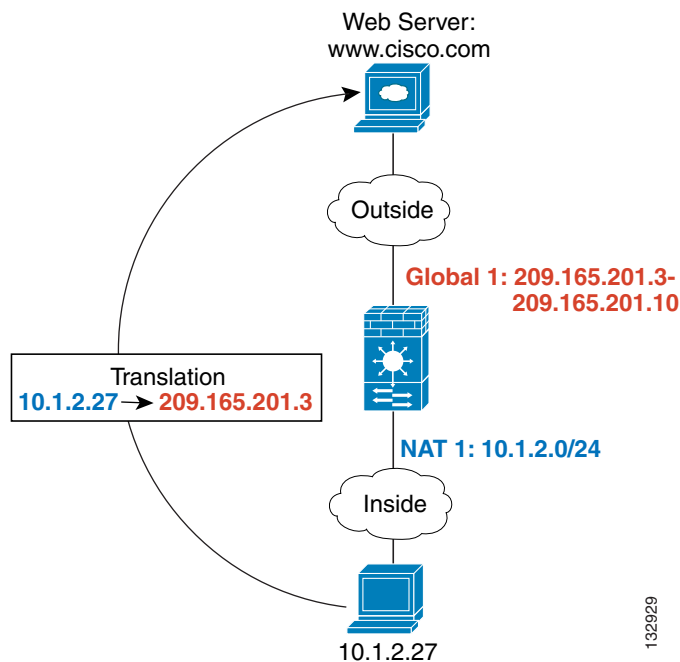
This section describes how to configure dynamic NAT and PAT, and includes the following topics:

- [Dynamic NAT and PAT Implementation, page 12-17](#)
- [Configuring Dynamic NAT or PAT, page 12-23](#)

Dynamic NAT and PAT Implementation

For dynamic NAT and PAT, you first configure a **nat** command identifying the real addresses on a given interface that you want to translate. Then you configure a separate **global** command to specify the mapped addresses when exiting another interface (in the case of PAT, this is one address). Each **nat** command matches a **global** command by comparing the NAT ID, a number that you assign to each command (see [Figure 12-13](#)).

Figure 12-13 *nat and global ID Matching*

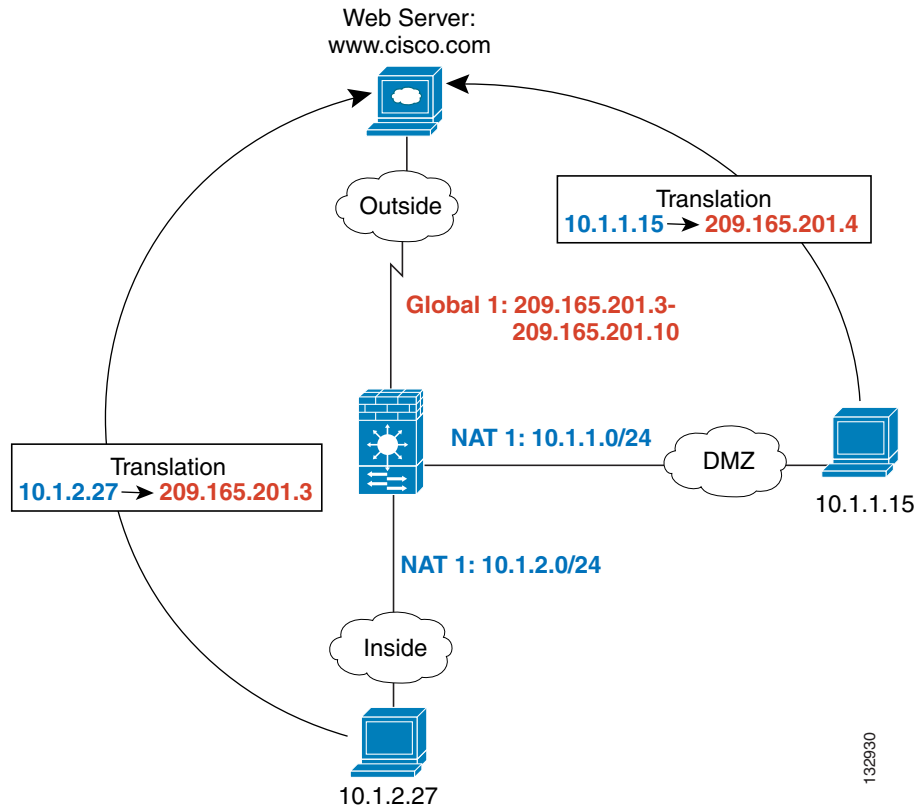


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
```

You can enter a **nat** command for each interface using the same NAT ID; they all use the same **global** command when traffic exits a given interface. For example, you can configure **nat** commands for Inside and DMZ interfaces, both on NAT ID 1. Then you configure a **global** command on the Outside interface that is also on ID 1. Traffic from the Inside interface and the DMZ interface share a mapped pool or a PAT address when exiting the Outside interface (see Figure 12-14).

Figure 12-14 *nat Commands on Multiple Interfaces*

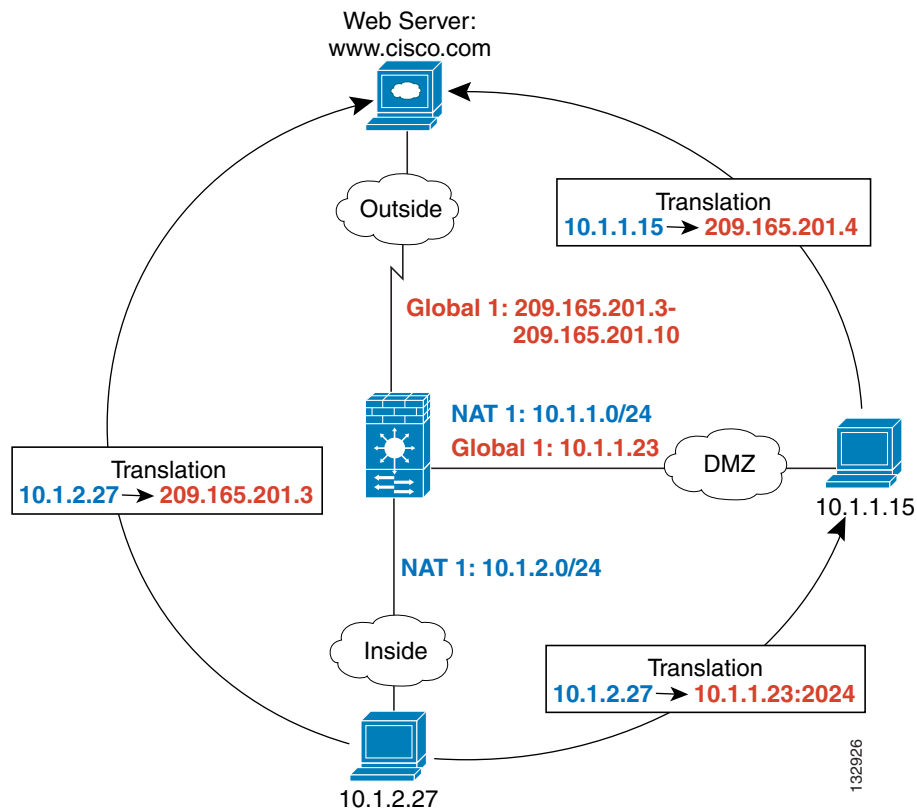


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
```

You can also enter a **global** command for each interface using the same NAT ID. If you enter a **global** command for the Outside and DMZ interfaces on ID 1, then the Inside **nat** command identifies traffic to be translated when going to both the Outside and the DMZ interfaces. Similarly, if you also enter a **nat** command for the DMZ interface on ID 1, then the **global** command on the Outside interface is also used for DMZ traffic. (See Figure 12-15).

Figure 12-15 *global and nat Commands on Multiple Interfaces*

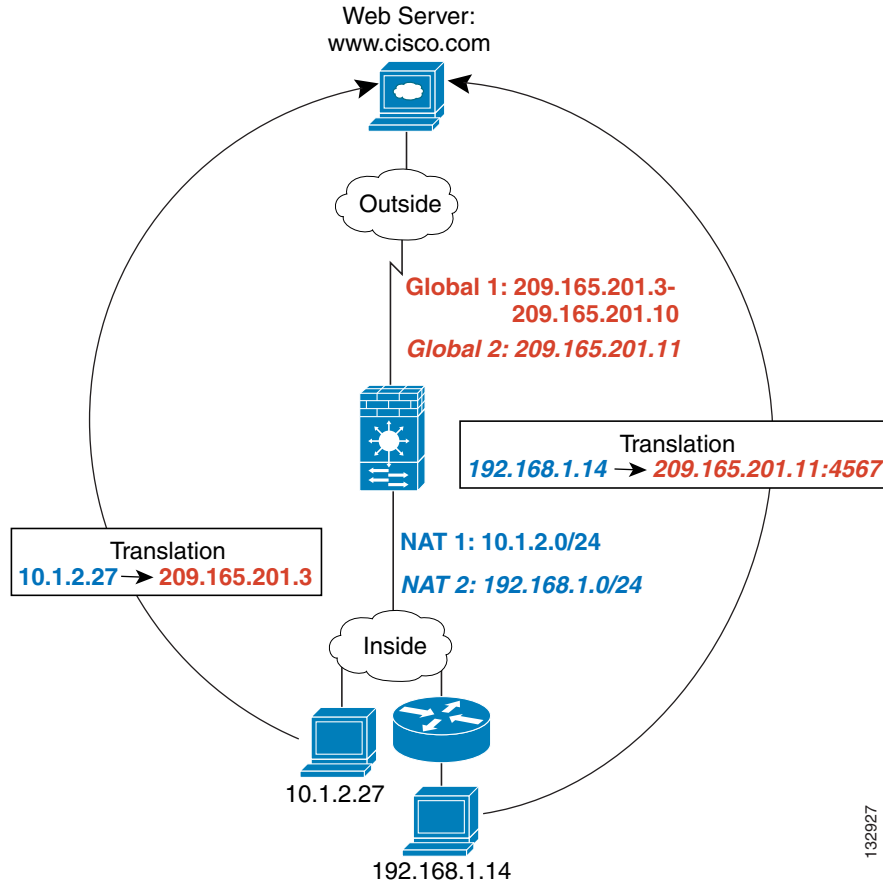


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
hostname(config)# global (dmz) 1 10.1.1.23
```

If you use different NAT IDs, you can identify different sets of real addresses to have different mapped addresses. For example, on the Inside interface, you can have two **nat** commands on two different NAT IDs. On the Outside interface, you configure two **global** commands for these two IDs. Then, when traffic from Inside network A exits the Outside interface, the IP addresses are translated to pool A addresses; while traffic from Inside network B are translated to pool B addresses (see Figure 12-16). If you use policy NAT, you can specify the same real addresses for multiple **nat** commands, as long as the destination addresses and ports are unique in each access list.

Figure 12-16 Different NAT IDs

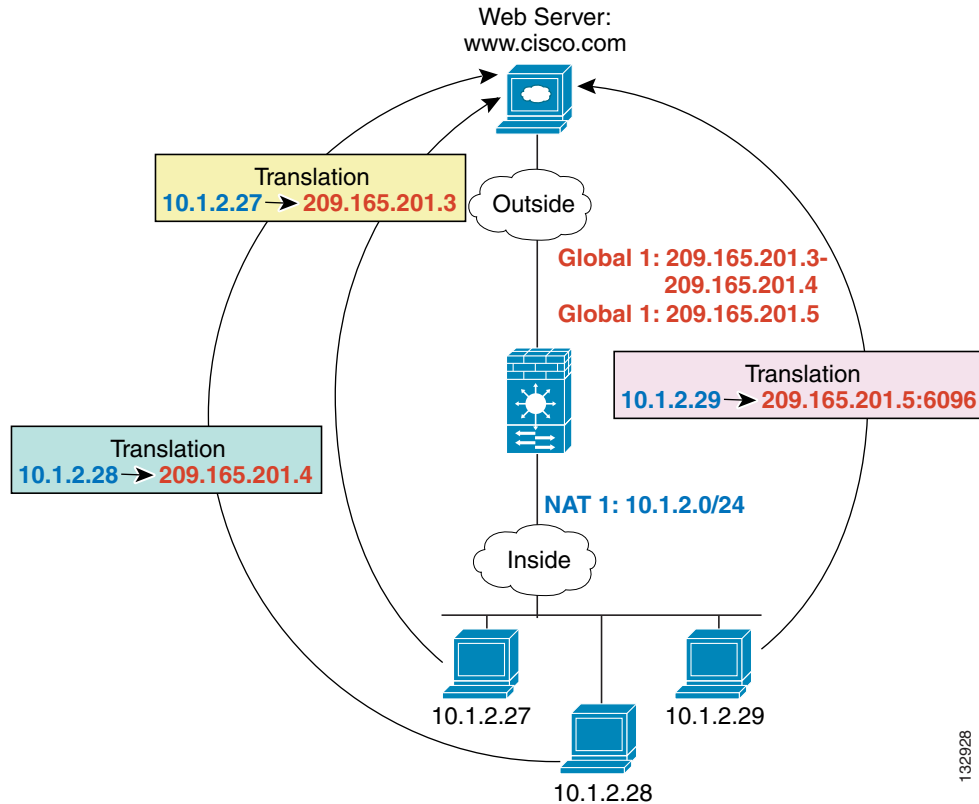


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# nat (inside) 2 192.168.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
hostname(config)# global (outside) 2 209.165.201.11
```

You can enter multiple **global** commands for one interface using the same NAT ID; the FWSM uses the dynamic NAT **global** commands first, in the order they are in the configuration, and then uses the PAT **global** commands in order. You might want to enter both a dynamic NAT **global** command and a PAT **global** command if you need to use dynamic NAT for a particular application, but want to have a backup PAT statement in case all the dynamic NAT addresses are depleted. Similarly, you might enter two PAT statements if you need more than the approximately 64,000 PAT sessions that a single PAT mapped statement supports (see [Figure 12-17](#)).

Figure 12-17 NAT and PAT Together



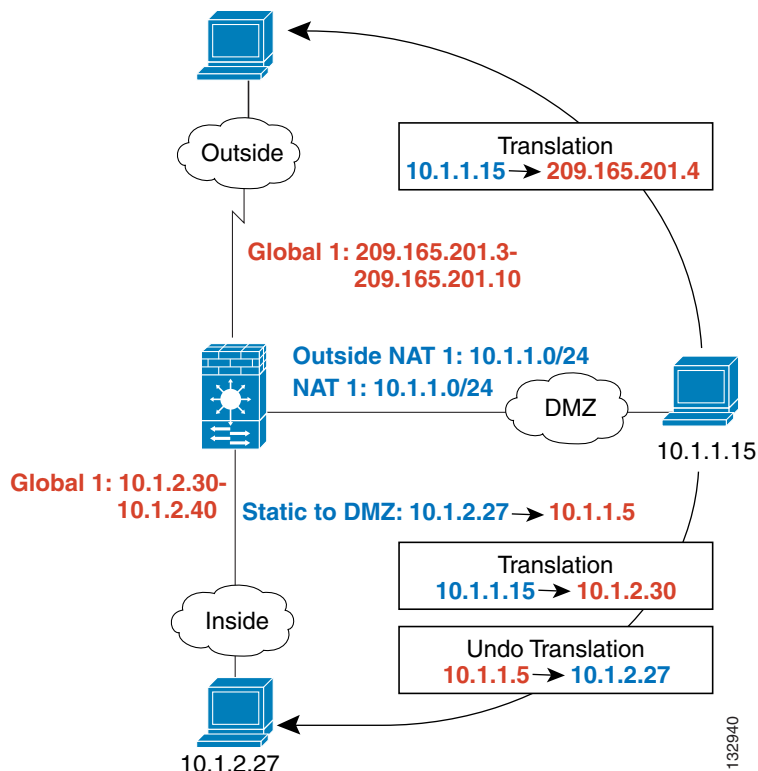
132928

See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.4
hostname(config)# global (outside) 1 209.165.201.5
```

For outside NAT, you need to identify the **nat** command for outside NAT (the **outside** keyword). If you also want to translate the same traffic when it accesses an inside interface (for example, traffic on a DMZ is translated when accessing the Inside and the Outside interfaces), then you must configure a separate **nat** command without the **outside** option. In this case, you can identify the same addresses in both statements and use the same NAT ID (see Figure 12-18). Note that for outside NAT (DMZ interface to Inside interface), the inside host uses a **static** command to allow outside access, so both the source and destination addresses are translated.

Figure 12-18 Outside NAT and Inside NAT Combined



See the following commands for this example:

```
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0 outside
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
hostname(config)# static (inside,dmz) 10.1.1.5 10.1.2.27 netmask 255.255.255.255
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.4
hostname(config)# global (inside) 1 10.1.2.30-1-10.1.2.40
```

When you specify a group of IP address(es) in a **nat** command, then you must perform NAT on that group of addresses when they access any lower or same security level interface; you must apply a **global** command with the same NAT ID on each interface, or use a **static** command. NAT is not required for that group when it accesses a higher security interface, because to perform NAT from outside to inside, you must create a separate **nat** command using the **outside** keyword. If you do apply outside NAT, then the NAT requirements preceding come into effect for that group of addresses when they access all higher security interfaces. Traffic identified by a **static** command is not affected.

Configuring Dynamic NAT or PAT

This section describes how to configure dynamic NAT or dynamic PAT. The configuration for dynamic NAT and PAT are almost identical; for NAT you specify a range of mapped addresses, and for PAT you specify a single address.

Figure 12-19 shows a typical dynamic NAT scenario. Only translated hosts can create a NAT session, and responding traffic is allowed back. The mapped address is dynamically assigned from a pool defined by the **global** command.

Figure 12-19 Dynamic NAT

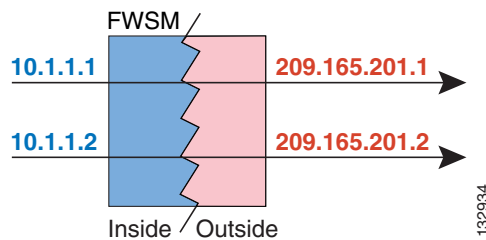
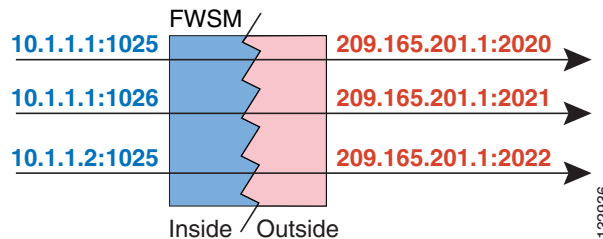


Figure 12-20 shows a typical dynamic PAT scenario. Only translated hosts can create a NAT session, and responding traffic is allowed back. The mapped address defined by the **global** command is the same for each translation, but the port is dynamically assigned.

Figure 12-20 Dynamic PAT



For more information about dynamic NAT, see the “Dynamic NAT” section on page 12-5. For more information about PAT, see the “PAT” section on page 12-7.



Note

If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections that use translations.

To configure dynamic NAT or PAT, perform the following steps:

Step 1 To identify the real addresses that you want to translate, enter one of the following commands:

- Policy NAT:

```
hostname(config)# nat (real_interface) nat_id access-list acl_name [dns] [outside]
[[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns] [norandomseq]
```

You can identify overlapping addresses in other **nat** commands. For example, you can identify 10.1.1.0 in one command, but 10.1.1.1 in another. The traffic is matched to a policy NAT command in order, until the first match, or for regular NAT, using the best match.

See the following description about options for this command:

- **access-list** *acl_name*—Identify the real addresses and destination addresses using an extended access list. Create the access list using the **access-list** command (see the “[Adding an Extended Access List](#)” section on page 10-6). This access list should include only **permit** ACEs. You can optionally specify the real and destination ports in the access list using the **eq** operator. Policy NAT does not consider the **inactive** or **time-range** keywords; all ACEs are considered to be active for policy NAT configuration.
- *nat_id*—An integer between 1 and 65535. The NAT ID should match a **global** command NAT ID. See the “[Dynamic NAT and PAT Implementation](#)” section on page 12-17 for more information about how NAT IDs are used. **0** is reserved for NAT exemption. (See the “[Configuring NAT Exemption](#)” section on page 12-33 for more information about NAT exemption.)
- **dns**—If your **nat** command includes the address of a host that has an entry in a DNS server, and the DNS server is on a different interface from a client, then the client and the DNS server need different addresses for the host; one needs the mapped address and one needs the real address. This option rewrites the address in the DNS reply to the client. The translated host needs to be on the same interface as either the client or the DNS server. Typically, hosts that need to allow access from other interfaces use a static translation, so this option is more likely to be used with the **static** command. (See the “[DNS and NAT](#)” section on page 12-14 for more information.)
- **outside**—If this interface is on a lower security level than the interface you identify by the matching **global** statement, then you must enter **outside** to identify the NAT instance as outside NAT.
- **tcp** *tcp_max_conns*—Sets the maximum number of simultaneous TCP connections for the entire subnet up to 65,536. The default is 0, which means the maximum connections.
- *emb_limit*—Sets the maximum number of embryonic connections per host up to 65,536. The default is 0, which means the maximum connections. You must enter the **tcp** *tcp_max_conns* before you enter the *emb_limit*. If you want to use the default value for *tcp_max_conns*, but change the *emb_limit*, then enter **0** for *tcp_max_conns*.

An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. Limiting the number of embryonic connections protects you from a DoS attack. The FWSM uses the embryonic limit to trigger TCP Intercept. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. TCP Intercept uses the SYN cookies algorithm to prevent TCP SYN-flooding attacks. A SYN-flooding attack consists of a series of SYN packets usually originating from spoofed IP addresses. The constant flood of SYN packets keeps the server SYN queue full, which prevents it from servicing connection requests. When the embryonic connection threshold of a connection is crossed, the FWSM acts as a proxy for the server and generates a SYN-ACK response to the client's SYN request. When the FWSM receives an ACK back from the client, it can then authenticate the client and allow the connection to the server.

- **udp udp_max_conns**—Sets the maximum number of simultaneous UDP connections for the entire subnet up to 65,536. The default is 0, which means the maximum connections.
- **norandomseq**—Disables TCP Initial Sequence Number (ISN) randomization. TCP initial sequence number randomization can be disabled if another in-line firewall is also randomizing the initial sequence numbers, because there is no need for both firewalls to be performing this action. However, leaving ISN randomization enabled on both firewalls does not affect the traffic. Each TCP connection has two ISNs: one generated by the client and one generated by the server. The security appliance randomizes the ISN of the TCP SYN passing in the outbound direction. If the connection is between two interfaces with the same security level, then the ISN will be randomized in the SYN in both directions. Randomizing the ISN of the protected host prevents an attacker from predicting the next ISN for a new connection and potentially hijacking the new session.

**Note**

You can alternatively set connection limits (but not embryonic connection limits) using the Modular Policy Framework. See the “[Configuring Connection Limits and Timeouts](#)” section on page 19-3 for more information. You can only set embryonic connection limits using NAT. If you configure these settings for the same traffic using both methods, then the FWSM uses the lower limit. For TCP sequence randomization, if it is disabled using either method, then the FWSM disables TCP sequence randomization.

- Regular NAT:

```
hostname(config)# nat (real_interface) nat_id real_ip [mask [dns] [outside]
[ [tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns] [norandomseq]]
```

The *nat_id* is an integer between 1 and 2147483647. The NAT ID must match a **global** command NAT ID. See the “[Dynamic NAT and PAT Implementation](#)” section on page 12-17 for more information about how NAT IDs are used. 0 is reserved for identity NAT. See the “[Configuring Identity NAT](#)” section on page 12-30 for more information about identity NAT.

See the preceding policy NAT command for information about other options.

- Step 2** To identify the mapped address(es) to which you want to translate the real addresses when they exit a particular interface, enter the following command:

```
hostname(config)# global (mapped_interface) nat_id {mapped_ip[-mapped_ip]}
```

This NAT ID should match a **nat** command NAT ID. The matching **nat** command identifies the addresses that you want to translate when they exit this interface.

You can specify a single address (for PAT) or a range of addresses (for NAT). The range can go across subnet boundaries if desired. For example, you can specify the following “supernet”:

```
192.168.1.1-192.168.2.254
```

For example, to translate the 10.1.1.0/24 network on the inside interface, enter the following command:

```
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.1-209.165.201.30
```

To identify a pool of addresses for dynamic NAT as well as a PAT address for when the NAT pool is exhausted, enter the following commands:

```
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.5
hostname(config)# global (outside) 1 209.165.201.10-209.165.201.20
```

To translate the lower security DMZ network addresses so they appear to be on the same network as the inside network (10.1.1.0), for example, to simplify routing, enter the following commands:

```
hostname(config)# nat (dmz) 1 10.1.2.0 255.255.255.0 outside dns
hostname(config)# global (inside) 1 10.1.1.45
```

To identify a single real address with two different destination addresses using policy NAT, enter the following commands (see [Figure 12-8 on page 12-10](#) for a related figure):

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
hostname(config)# access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
hostname(config)# nat (inside) 1 access-list NET1 tcp 0 2000 udp 10000
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list NET2 tcp 1000 500 udp 2000
hostname(config)# global (outside) 2 209.165.202.130
```

To identify a single real address/destination address pair that use different ports using policy NAT, enter the following commands (see [Figure 12-9 on page 12-11](#) for a related figure):

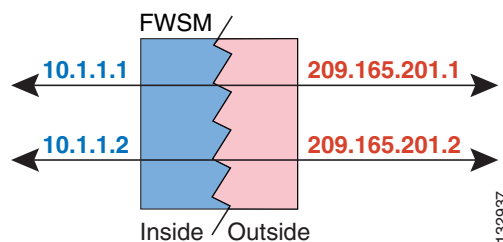
```
hostname(config)# access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 80
hostname(config)# access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 23
hostname(config)# nat (inside) 1 access-list WEB
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list TELNET
hostname(config)# global (outside) 2 209.165.202.130
```

Using Static NAT

This section describes how to configure a static translation.

[Figure 12-21](#) shows a typical static NAT scenario. The translation is always active so both translated and remote hosts can originate connections, and the mapped address is statically assigned by the **static** command.

Figure 12-21 Static NAT



You cannot use the same real or mapped address in multiple **static** commands between the same two interfaces. Do not use a mapped address in the **static** command that is also defined in a **global** command for the same mapped interface.

For more information about static NAT, see the “[Static NAT](#)” section on page 12-7.

**Note**

If you remove a **static** command, existing connections that use the translation are not affected. To remove these connections, enter the **clear local-host** or the **clear xlate** command. Static translations from the translation table can be removed using the **clear xlate** command; the translation table will be cleared and all current translations are deleted. The **clear xlate** command clears all connections, even when **xlate-bypass** is enabled and when a connection does not have an **xlate**.

For more information about these commands, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

To configure static NAT, enter one of the following commands.

- For policy static NAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) mapped_ip
access-list acl_name [dns] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
[norandomseq]
```

Create the access list using the **access-list** command (see the “[Adding an Extended Access List](#)” section on page 10-6). This access list should include only **permit** ACEs. The source subnet mask used in the access list is also used for the mapped addresses. You can also specify the real and destination ports in the access list using the **eq** operator. Policy NAT does not consider the **inactive** or **time-range** keywords; all ACEs are considered to be active for policy NAT configuration. See the “[Policy NAT](#)” section on page 12-9 for more information.

If you specify a network for translation (for example, 10.1.1.0 255.255.255.0), then the FWSM translates the .0 and .255 addresses. If you want to prevent access to these addresses, be sure to configure an access list to deny access.

See the “[Configuring Dynamic NAT or PAT](#)” section on page 12-23 for information about the other options.

- To configure regular static NAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) mapped_ip real_ip
[netmask mask] [dns] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
[norandomseq]
```

See the “[Configuring Dynamic NAT or PAT](#)” section on page 12-23 for information about the options.

For example, the following policy static NAT example shows a single real address that is translated to two mapped addresses depending on the destination address (see [Figure 12-8](#) on page 12-10 for a related figure):

```
hostname(config)# access-list NET1 permit ip host 10.1.2.27 209.165.201.0 255.255.255.224
hostname(config)# access-list NET2 permit ip host 10.1.2.27 209.165.200.224
255.255.255.224
hostname(config)# static (inside,outside) 209.165.202.129 access-list NET1
hostname(config)# static (inside,outside) 209.165.202.130 access-list NET2
```

The following command maps an inside IP address (10.1.1.3) to an outside IP address (209.165.201.12):

```
hostname(config)# static (inside,outside) 209.165.201.12 10.1.1.3 netmask 255.255.255.255
```

The following command maps the outside address (209.165.201.15) to an inside address (10.1.1.6):

```
hostname(config)# static (outside,inside) 10.1.1.6 209.165.201.15 netmask 255.255.255.255
```

The following command statically maps an entire subnet:

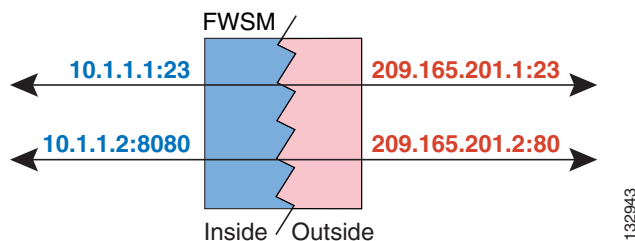
```
hostname(config)# static (inside,dmz) 10.1.1.0 10.1.2.0 netmask 255.255.255.0
```

Using Static PAT

This section describes how to configure a static port translation. Static PAT lets you translate the real IP address to a mapped IP address, as well as the real port to a mapped port. You can choose to translate the real port to the same port, which lets you translate only specific types of traffic, or you can take it further by translating to a different port.

Figure 12-22 shows a typical static PAT scenario. The translation is always active so that both translated and remote hosts can originate connections, and the mapped address and port is statically assigned by the **static** command.

Figure 12-22 Static PAT



For applications that require application inspection for secondary channels (FTP, VoIP, etc.), the FWSM automatically translates the secondary ports.

You cannot use the same real or mapped address in multiple **static** statements between the same two interfaces. Do not use a mapped address in the **static** command that is also defined in a **global** command for the same mapped interface.

For more information about static PAT, see the “Static PAT” section on page 12-7.



Note

If you remove a **static** command, existing connections that use the translation are not affected. To remove these connections, enter the **clear local-host** command.

Static translations from the translation table can be removed using the **clear xlate** command; the translation table will be cleared and all current translations are deleted.

To configure static PAT, enter one of the following commands.

- For policy static PAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) {tcp | udp} mapped_ip
mapped_port access-list acl_name [dns] [[tcp] tcp_max_conns [emb_limit]]
[udp udp_max_conns] [norandomseq]
```

Create the access list using the **access-list** command (see the “Adding an Extended Access List” section on page 10-6). The protocol in the access list must match the protocol you set in this command. For example, if you specify **tcp** in the **static** command, then you must specify **tcp** in the access list. Specify the port using the **eq** operator. This access list should include only **permit** ACEs.

The source subnet mask used in the access list is also used for the mapped addresses. Policy NAT does not consider the **inactive** or **time-range** keywords; all ACEs are considered to be active for policy NAT configuration.

If you specify a network for translation (for example, 10.1.1.0 255.255.255.0), then the FWSM translates the .0 and .255 addresses. If you want to prevent access to these addresses, be sure to configure an access list to deny access.

See the “[Configuring Dynamic NAT or PAT](#)” section on page 12-23 for information about the other options.

- To configure regular static PAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) {tcp | udp} mapped_ip
mapped_port real_ip real_port [netmask mask] [dns] [[tcp] tcp_max_conns [emb_limit]]
[udp udp_max_conns] [norandomseq]
```

See the “[Configuring Dynamic NAT or PAT](#)” section on page 12-23 for information about the options.

For example, for Telnet traffic initiated from hosts on the 10.1.3.0 network to the FWSM outside interface (10.1.2.14), you can redirect the traffic to the inside host at 10.1.1.15 by entering the following commands:

```
hostname(config)# access-list TELNET permit tcp host 10.1.1.15 eq telnet 10.1.3.0
255.255.255.0 eq telnet
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet access-list TELNET
```

For HTTP traffic initiated from hosts on the 10.1.3.0 network to the FWSM outside interface (10.1.2.14), you can redirect the traffic to the inside host at 10.1.1.15 by entering:

```
hostname(config)# access-list HTTP permit tcp host 10.1.1.15 eq http 10.1.3.0
255.255.255.0 eq http
hostname(config)# static (inside,outside) tcp 10.1.2.14 http access-list HTTP
```

To redirect Telnet traffic from the FWSM outside interface (10.1.2.14) to the inside host at 10.1.1.15, enter the following command:

```
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask
255.255.255.255
```

If you want to allow the preceding real Telnet server to initiate connections, though, then you need to provide additional translation. For example, to translate all other types of traffic, enter the following commands. The original **static** command provides translation for Telnet to the server, while the **nat** and **global** commands provide PAT for outbound connections from the server.

```
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask
255.255.255.255
hostname(config)# nat (inside) 1 10.1.1.15 255.255.255.255
hostname(config)# global (outside) 1 10.1.2.14
```

If you also have a separate translation for all inside traffic, and the inside hosts use a different mapped address from the Telnet server, you can still configure traffic initiated from the Telnet server to use the same mapped address as the **static** statement that allows Telnet traffic to the server. You need to create a more exclusive **nat** statement just for the Telnet server. Because **nat** statements are read for the best match, more exclusive **nat** statements are matched before general statements. The following example shows the Telnet **static** statement, the more exclusive **nat** statement for initiated traffic from the Telnet server, and the statement for other inside hosts, which uses a different mapped address.

```
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask
255.255.255.255
hostname(config)# nat (inside) 1 10.1.1.15 255.255.255.255
hostname(config)# global (outside) 1 10.1.2.14
```

```
hostname(config)# nat (inside) 2 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 2 10.1.2.78
```

To translate a well-known port (80) to another port (8080), enter the following command:

```
hostname(config)# static (inside,outside) tcp 10.1.2.45 80 10.1.1.16 8080 netmask
255.255.255.255
```

Bypassing NAT

This section describes how to bypass NAT. You might want to bypass NAT when you enable NAT control. You can bypass NAT using identity NAT, static identity NAT, or NAT exemption. See the “[Bypassing NAT when NAT Control is Enabled](#)” section on page 12-8 for more information about these methods. This section includes the following topics:

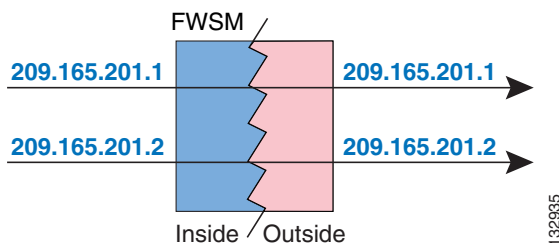
- [Configuring Identity NAT, page 12-30](#)
- [Configuring Static Identity NAT, page 12-32](#)
- [Configuring NAT Exemption, page 12-33](#)

Configuring Identity NAT

Identity NAT translates the real IP address to the same IP address. Only “translated” hosts can create NAT translations, and responding traffic is allowed back.

Figure 12-23 shows a typical identity NAT scenario.

Figure 12-23 Identity NAT



Note

If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections that use translations.

To configure identity NAT, enter the following command:

```
hostname(config)# nat (real_interface) 0 real_ip [mask [dns] [outside]
[[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns] [norandomseq]]
```

See the “[Configuring Dynamic NAT or PAT](#)” section on page 12-23 for information about the options.

For example, to use identity NAT for the inside 10.1.1.0/24 network, enter the following command:

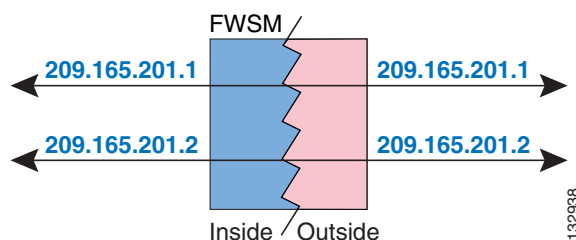
```
hostname(config)# nat (inside) 0 10.1.1.0 255.255.255.0
```


Configuring Static Identity NAT

Static identity NAT translates the real IP address to the same IP address. The translation is always active, and both “translated” and remote hosts can originate connections. Static identity NAT lets you use regular NAT or policy NAT. Policy NAT lets you identify the real and destination addresses when determining the real addresses to translate (see the “[Policy NAT](#)” section on page 12-9 for more information about policy NAT). For example, you can use policy static identity NAT for an inside address when it accesses the outside interface and the destination is server A, but use a normal translation when accessing the outside server B.

Figure 12-24 shows a typical static identity NAT scenario.

Figure 12-24 Static Identity NAT



Note

If you remove a **static** command, existing connections that use the translation are not affected. To remove these connections, enter the **clear local-host** command.

Static translations from the translation table can be removed using the **clear xlate** command; the translation table will be cleared and all current translations are deleted.

To configure static identity NAT, enter one of the following commands:

- To configure policy static identity NAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) real_ip access-list acl_id
[ dns ] [[ tcp ] tcp_max_conns [ emb_limit ] ] [ udp udp_max_conns ] [ norandomseq ]
```

Create the access list using the **access-list** command (see the “[Adding an Extended Access List](#)” section on page 10-6). This access list should include only **permit** ACEs. Make sure the source address in the access list matches the *real_ip* in this command. Policy NAT does not consider the **inactive** or **time-range** keywords; all ACEs are considered to be active for policy NAT configuration. See the “[Policy NAT](#)” section on page 12-9 for more information.

See the “[Configuring Dynamic NAT or PAT](#)” section on page 12-23 for information about the other options.

- To configure regular static identity NAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) real_ip real_ip
[ netmask mask ] [ dns ] [[ tcp ] tcp_max_conns [ emb_limit ] ] [ udp udp_max_conns ]
[ norandomseq ]
```

Specify the same IP address for both *real_ip* arguments.

See the “[Configuring Dynamic NAT or PAT](#)” section on page 12-23 for information about the other options.

For example, the following command uses static identity NAT for an inside IP address (10.1.1.3) when accessed by the outside:

```
hostname(config)# static (inside,outside) 10.1.1.3 10.1.1.3 netmask 255.255.255.255
```

The following command uses static identity NAT for an outside address (209.165.201.15) when accessed by the inside:

```
hostname(config)# static (outside,inside) 209.165.201.15 209.165.201.15 netmask 255.255.255.255
```

The following command statically maps an entire subnet:

```
hostname(config)# static (inside,dmz) 10.1.2.0 10.1.2.0 netmask 255.255.255.0
```

The following static identity policy NAT example shows a single real address that uses identity NAT when accessing one destination address, and a translation when accessing another:

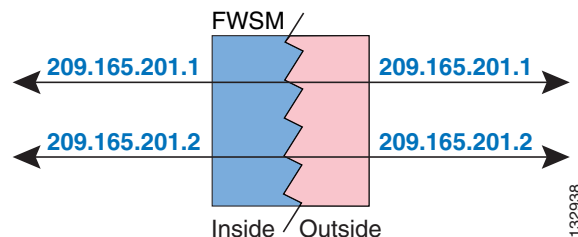
```
hostname(config)# access-list NET1 permit ip host 10.1.2.27 209.165.201.0 255.255.255.224
hostname(config)# access-list NET2 permit ip host 10.1.2.27 209.165.200.224 255.255.255.224
hostname(config)# static (inside,outside) 10.1.2.27 access-list NET1
hostname(config)# static (inside,outside) 209.165.202.130 access-list NET2
```

Configuring NAT Exemption

NAT exemption exempts addresses from translation and allows both real and remote hosts to originate connections. NAT exemption lets you specify the real and destination addresses when determining the real traffic to exempt (similar to policy NAT), so you have greater control using NAT exemption than identity NAT. However unlike policy NAT, NAT exemption does not consider the ports in the access list. Use static identity NAT to consider ports in the access list.

Figure 12-25 shows a typical NAT exemption scenario.

Figure 12-25 NAT Exemption



Note

If you remove a NAT exemption configuration, existing connections that use NAT exemption are not affected. To remove these connections, enter the **clear local-host** command.

To configure NAT exemption, enter the following command:

```
hostname(config)# nat (real_interface) 0 access-list acl_name [outside] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns] [norandomseq]
```

Create the extended access list using the **access-list extended** command (see the “[Adding an Extended Access List](#)” section on page 10-6). This access list can include both **permit** ACEs and **deny** ACEs. Do not specify the real and destination ports in the access list; NAT exemption does not consider the ports. NAT exemption also does not consider the **inactive** or **time-range** keywords; all ACEs are considered to be active for NAT exemption configuration.

See the “[Configuring Dynamic NAT or PAT](#)” section on page 12-23 for information about the other options.

By default, this command exempts traffic from inside to outside. If you want traffic from outside to inside to bypass NAT, then add an additional **nat** command and enter **outside** to identify the NAT instance as outside NAT. You might want to use outside NAT exemption if you configure dynamic NAT for the outside interface and want to exempt other traffic.

For example, to exempt an inside network when accessing any destination address, enter the following command:

```
hostname(config)# access-list EXEMPT permit ip 10.1.2.0 255.255.255.0 any
hostname(config)# nat (inside) 0 access-list EXEMPT
```

To use dynamic outside NAT for a DMZ network, and exempt another DMZ network, enter the following command:

```
hostname(config)# nat (dmz) 1 10.1.2.0 255.255.255.0 outside dns
hostname(config)# global (inside) 1 10.1.1.45
hostname(config)# access-list EXEMPT permit ip 10.1.3.0 255.255.255.0 any
hostname(config)# nat (dmz) 0 access-list EXEMPT outside
```

To exempt an inside address when accessing two different destination addresses, enter the following commands:

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
hostname(config)# nat (inside) 0 access-list NET1
```

NAT Examples

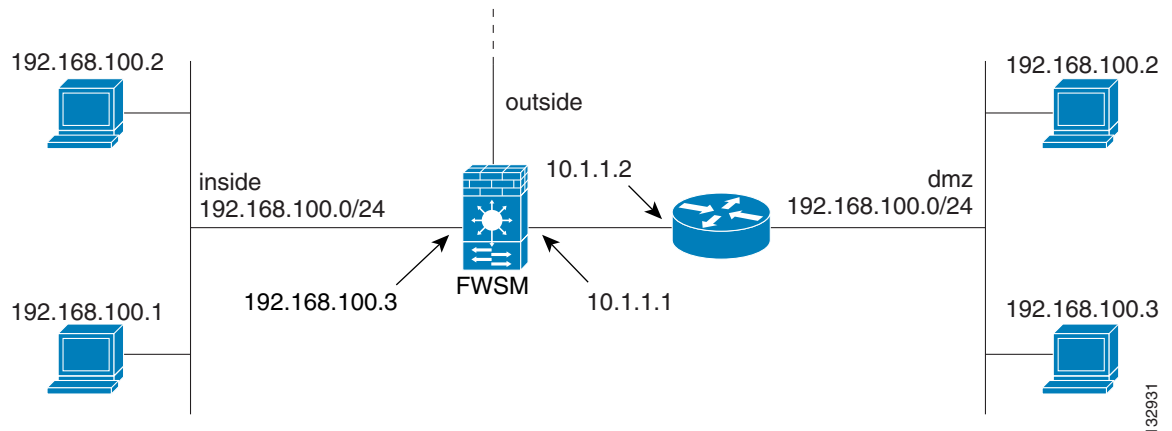
This section describes typical scenarios that use NAT solutions, and includes the following topics:

- [Overlapping Networks, page 12-35](#)
- [Redirecting Ports, page 12-36](#)

Overlapping Networks

In [Figure 12-26](#), the FWSM connects two private networks with overlapping address ranges.

Figure 12-26 Using Outside NAT with Overlapping Networks



Two networks use an overlapping address space (192.168.100.0/24), but hosts on each network must communicate (as allowed by access lists). Without NAT, when a host on the inside network tries to access a host on the overlapping DMZ network, the packet never makes it past the FWSM, which sees the packet as having a destination address on the inside network. Moreover, if the destination address is being used by another host on the inside network, that host receives the packet.

To solve this problem, use NAT to provide non-overlapping addresses. If you want to allow access in both directions, use static NAT for both networks. If you only want to allow the inside interface to access hosts on the DMZ, then you can use dynamic NAT for the inside addresses, and static NAT for the DMZ addresses you want to access. This example shows static NAT.

To configure static NAT for these two interfaces, perform the following steps. The 10.1.1.0/24 network on the DMZ is not translated.

-
- Step 1** Translate 192.168.100.0/24 on the inside to 10.1.2.0 /24 when it accesses the DMZ by entering the following command:
- ```
hostname(config)# static (inside,dmz) 10.1.2.0 192.168.100.0 netmask 255.255.255.0
```
- Step 2** Translate the 192.168.100.0/24 network on the DMZ to 10.1.3.0/24 when it accesses the inside by entering the following command:
- ```
hostname(config)# static (dmz,inside) 10.1.3.0 192.168.100.0 netmask 255.255.255.0
```
- Step 3** Configure the following static routes so that traffic to the DMZ network can be routed correctly by the FWSM:
- ```
hostname(config)# route dmz 192.168.100.128 255.255.255.128 10.1.1.2 1
hostname(config)# route dmz 192.168.100.0 255.255.255.128 10.1.1.2 1
```

The FWSM already has a connected route for the inside network. These static routes allow the FWSM to send traffic for the 192.168.100.0/24 network out the DMZ interface to the gateway router at 10.1.1.2. (You need to split the network into two because you cannot create a static route with the exact same network as a connected route.) Alternatively, you could use a more broad route for the DMZ traffic, such as a default route.

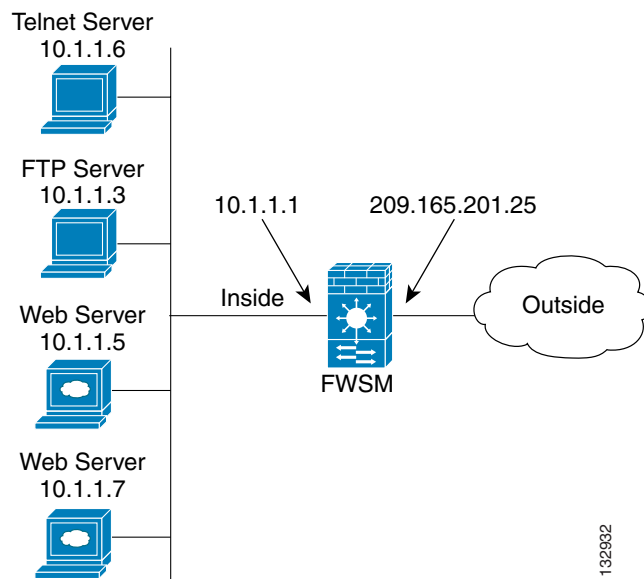
If host 192.168.100.2 on the DMZ network wants to initiate a connection to host 192.168.100.2 on the inside network, the following events occur:

1. The DMZ host 192.168.100.2 sends the packet to IP address 10.1.2.2.
2. When the FWSM receives this packet, the FWSM translates the source address from 192.168.100.2 to 10.1.3.2.
3. Then the FWSM translates the destination address from 10.1.2.2 to 192.168.100.2, and the packet is forwarded.

## Redirecting Ports

Figure 12-27 illustrates a typical network scenario in which the port redirection feature might be useful.

**Figure 12-27** Port Redirection Using Static PAT



In the configuration described in this section, port redirection occurs for hosts on external networks as follows:

- Telnet requests to IP address 209.165.201.5 are redirected to 10.1.1.6.
- FTP requests to IP address 209.165.201.5 are redirected to 10.1.1.3.
- HTTP request to FWSM outside IP address 209.165.201.25 are redirected to 10.1.1.5.
- HTTP port 8080 requests to PAT address 209.165.201.15 are redirected to 10.1.1.7 port 80.

To implement this scenario, perform the following steps:

---

**Step 1** Configure PAT for the inside network by entering the following commands:

```
hostname(config)# nat (inside) 1 0.0.0.0 0.0.0.0 0 0
hostname(config)# global (outside) 1 209.165.201.15
```

**Step 2** Redirect Telnet requests for 209.165.201.5 to 10.1.1.6 by entering the following command:

```
hostname(config)# static (inside,outside) tcp 209.165.201.5 telnet 10.1.1.6 telnet netmask
255.255.255.255
```

**Step 3** Redirect FTP requests for IP address 209.165.201.5 to 10.1.1.3 by entering the following command:

```
hostname(config)# static (inside,outside) tcp 209.165.201.5 ftp 10.1.1.3 ftp netmask
255.255.255.255
```

**Step 4** Redirect HTTP requests for the FWSM outside interface address to 10.1.1.5 by entering the following command:

```
hostname(config)# static (inside,outside) tcp interface www 10.1.1.5 www netmask
255.255.255.255
```

**Step 5** Redirect HTTP requests on port 8080 for PAT address 209.165.201.15 to 10.1.1.7 port 80 by entering the following command:

```
hostname(config)# static (inside,outside) tcp 209.165.201.15 8080 10.1.1.7 www netmask
255.255.255.255
```

---

