CHAPTER 28

Configuring Network Object NAT

All NAT rules that are configured as a parameter of a network object are considered to be *network object NAT* rules. Network object NAT is a quick and easy way to configure NAT for a network object, which can be a single IP address, a range of addresses, or a subnet. After you configure the network object, you can then identify the mapped address for that object.

This chapter describes how to configure network object NAT, and it includes the following sections:

Note For detailed information about how NAT works, see Chapter 27, "Information About NAT."

Information About Network Object NAT

When a packet enters the adaptive security appliance, both the source and destination IP addresses are checked against the network object NAT rules. The source and destination address in the packet can be translated by separate rules if separate matches are made. These rules are not tied to each other; different combinations of rules can be used depending on the traffic.

Because the rules are never paired, you cannot specify that a source address should be translated to A when going to destination X, but be translated to B when going to destination Y. Use twice NAT for that kind of functionality (twice NAT lets you identify the source and destination address in a single rule).

For detailed information about the differences between twice NAT and network object NAT, see the "How NAT is Implemented" section on page 27-15.

Network object NAT rules are added to section 2 of the NAT rules table. For more information about NAT ordering, see the "NAT Rule Order" section on page 27-19.

Licensing Requirements for Network Object NAT

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Prerequisites for Network Object NAT

Depending on the configuration, you can configure the mapped address inline if desired or you can create a network object or network object group for the mapped address (the **object network** or **object-group network** command). Network object groups are particularly useful for creating a mapped address pool with discontinous IP address ranges or multiple hosts or subnets. To create a network object or group, see the "Configuring Objects and Groups" section on page 11-1.

For specific guidelines for objects and groups, see the configuration section for the NAT type you want to configure. See also the "Guidelines and Limitations" section.

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 mode.
- In transparent mode, you must specify the real and mapped interfaces; you cannot use **any**.
- In transparent mode, you cannot configure interface PAT, because the transparent mode interfaces do not have IP addresses. You also cannot use the management IP address as a mapped address.

IPv6 Guidelines

Does not support IPv6.

Additional Guidelines

- You can only define a single NAT rule for a given object; if you want to configure multiple NAT rules, you need to create multiple objects that specify the same IP address, for example, **object network obj-10.10.10.1-01**, **object network obj-10.10.10.1-02**, and so on.

- 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.

Note If you remove a dynamic NAT or PAT rule, and then add a new rule with mapped addresses that overlap the addresses in the removed rule, then the new rule will not be used until all connections associated with the removed rule time out or are cleared using the **clear xlate** command. This safeguard ensures that the same address is not assigned to multiple hosts.

- Objects and object groups used in NAT cannot be undefined; they must include IP addresses.
- The mapped IP address pool cannot include:
  - The mapped interface IP address. If you specify **any** interface for the rule, then all interface IP addresses are disallowed. For interface PAT (routed mode only), use the **interface** keyword instead of the IP address.
  - (Transparent mode) The management IP address.
  - (Dynamic NAT) The standby interface IP address when VPN is enabled.
  - Existing VPN pool addresses.

Configuring Network Object NAT

This section describes how to configure network object NAT to create rules for dynamic NAT, dynamic PAT, static NAT, static NAT with port translation, and identity NAT. This section includes the following topics:

Configuring Dynamic NAT

This section describes how to configure a dynamic NAT rule using network object NAT. For more information, see the "Dynamic NAT" section on page 27-8.

Detailed Steps

	Command	Purpose

Step 1	Network object:
	object network *obj_name*
	  range *ip_address_1 ip_address_2*
	
	Network object group:
	object-group network *grp_name*
	  {**network-object** {**object** *net_obj_name* |
	  host *ip_address*} |
	  group-object *grp_obj_name*}
	
	Example:
	`hostname(config)# object network TEST`
	`hostname(config-network-object)# range 10.1.1.1 10.1.1.70`
	
	`hostname(config)# object network TEST2`
	`hostname(config-network-object)# range 10.1.2.1 10.1.2.70`
	
	`hostname(config-network-object)# object-group network MAPPED_IPS`
	`hostname(config-network)# network-object object TEST`
	`hostname(config-network)# network-object object TEST2`
	`hostname(config-network)# network-object host 10.1.2.79`
		To specify the mapped addresses (that you want to translate to), configure a network object or network object group. A network object group can contain objects and/or inline addresses.
		
		Note The object or group cannot contain a subnet. You can share this mapped object across different dynamic NAT rules, if desired.
		
		See the "Guidelines and Limitations" section on page 28-2 for information about disallowed mapped IP addresses.
		
		For more information, see the "Configuring Objects" section on page 11-3.

Step 2	**object network** *obj_name* **Example:** `hostname(config)# object network my-host-obj1`	Configures a network object for which you want to configure NAT, or enters object network configuration mode for an existing network object.

	Command	Purpose
Step 3	{**host** *ip_address* | **subnet** *subnet_address netmask* | **range** *ip_address_1 ip_address_2*} **Example:** `hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0`	If you are creating a new network object, defines the real IP address(es) that you want to translate.
Step 4	**nat** [(*real_ifc*,*mapped_ifc*)] **dynamic** *mapped_obj* [**interface**] [**dns**] **Example:** `hostname(config-network-object)# nat (inside,outside) dynamic MAPPED_IPS interface`	Configures dynamic NAT for the object IP addresses. See the following guidelines: • Interfaces—If you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword **any** for one or both of the interfaces. Be sure to include the parentheses in your command. • Mapped IP address—Specify the mapped IP address as: – An existing network object (see Step 1). – An existing network object group (see Step 1). **Note** You can share this mapped object across different dynamic NAT rules, if desired. • Interface PAT fallback—If you specify a mapped object or group followed by the **interface** keyword, then the IP address of the mapped interface is only used if all of the other mapped addresses are already allocated. For this option, you must configure a specific interface for the *mapped_ifc*. (You cannot specify **interface** in transparent mode). • DNS—The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See the "DNS and NAT" section on page 27-21 for more information. **Note** You can only define a single NAT rule for a given object. See the "Additional Guidelines" section on page 28-2.

Examples

The following example configures dynamic NAT that hides 192.168.2.0 network behind a range of outside addresses 10.2.2.1 through 10.2.2.10:

```
hostname(config)# object network my-range-obj
hostname(config-network-object)# range 10.2.2.1 10.2.2.10
hostname(config)# object network my-inside-net
hostname(config-network-object)# subnet 192.168.2.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) dynamic my-range-obj
```

The following example configures dynamic NAT with dynamic PAT backup. Hosts on inside network 10.76.11.0 are mapped first to the nat-range1 pool (10.10.10.10-10.10.10.20). After all addresses in the nat-range1 pool are allocated, dynamic PAT is performed using the pat-ip1 address (10.10.10.21). In the unlikely event that the PAT translations are also use up, dynamic PAT is performed using the outside interface address.

```
hostname(config)# object network nat-range1
hostname(config-network-object)# range 10.10.10.10 10.10.10.20
```

```
hostname(config-network-object)# object network pat-ip1
hostname(config-network-object)# host 10.10.10.21

hostname(config-network-object)# object-group network nat-pat-grp
hostname(config-network-object)# network-object object nat-range1
hostname(config-network-object)# network-object object pat-ip1

hostname(config-network-object)# object network my_net_obj5
hostname(config-network-object)# subnet 10.76.11.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) dynamic nat-pat-grp interface
```

Configuring Dynamic PAT (Hide)

This section describes how to configure a dynamic PAT (hide) rule using network object NAT. For more information, see the "Dynamic PAT" section on page 27-10.

Detailed Steps

	Command	Purpose

Step 1	(Optional)
	object network *obj_name*
	host *ip_address*
	
	Example:
	`hostname(config)# object network MAPPED_IP`
	`hostname(config-network-object)# host 10.1.1.1`
		To specify the mapped address (that you want to translate to), configure a network object. Alternatively, you can enter the IP address as an inline value for the **nat** command.
		Note You can share this mapped object across different dynamic PAT rules, if desired.
		See the "Guidelines and Limitations" section on page 28-2 for information about disallowed mapped IP addresses.
		For more information, see the "Configuring Objects" section on page 11-3.

Step 2	**object network** *obj_name* **Example:** `hostname(config)# object network my-host-obj1`	Configures a network object for which you want to configure NAT, or enters object network configuration mode for an existing network object.

	Command	Purpose
Step 3	{**host** *ip_address* | **subnet** *subnet_address netmask* | **range** *ip_address_1 ip_address_2*} **Example:** `hostname(config-network-object)# range 10.1.1.1 10.1.1.90`	If you are creating a new network object, defines the real IP address(es) that you want to translate.
Step 4	**nat** [(*real_ifc*,*mapped_ifc*)] **dynamic** {*mapped_inline_host_ip* | *mapped_obj* | **interface**} [**dns**] **Example:** `hostname(config-network-object)# nat (any,outside) dynamic interface`	Configures dynamic PAT for the object IP addresses. See the following guidelines: • Interfaces—If you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword **any** for one or both of the interfaces. Be sure to include the parentheses in your command. • Mapped IP address—You can specify the mapped IP address as: – An inline host address. – An existing network object that is defined as a host address (see Step 1). – **interface**—(Routed mode only) The IP address of the mapped interface is used as the mapped address. For this option, you must configure a specific interface for the *mapped_ifc*. You must use this keyword when you want to use the interface IP address; you cannot enter it inline or as an object. **Note** You can share this mapped IP address across different dynamic PAT rules, if desired. • DNS—The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See the "DNS and NAT" section on page 27-21 for more information. **Note** You can only define a single NAT rule for a given object. See the "Additional Guidelines" section on page 28-2.

Examples

The following example configures dynamic PAT that hides the 192.168.2.0 network behind address 10.2.2.2:

```
hostname(config)# object network my-inside-net
hostname(config-network-object)# subnet 192.168.2.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) dynamic 10.2.2.2
```

The following example configures dynamic PAT that hides the 192.168.2.0 network behind the outside interface address:

```
hostname(config)# object network my-inside-net
hostname(config-network-object)# subnet 192.168.2.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) dynamic interface
```

Configuring Static NAT or Static NAT with Port Translation

This section describes how to configure a static NAT rule using network object NAT. For more information, see the "Static NAT" section on page 27-3.

Detailed Steps

	Command	Purpose
Step 1	(Optional) Network object: **object network** *obj_name* {**host** *ip_address* | **subnet** *subnet_address netmask* | **range** *ip_address_1 ip_address_2*} Network object group: **object-group network** *grp_name* {**network-object** {**object** *net_obj_name* | *subnet_address netmask* | **host** *ip_address*} | **group-object** *grp_obj_name*} **Example:** hostname(config)# object network MAPPED_IPS hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0	To specify the mapped addresses (that you want to translate to), configure a network object or network object group. A network object group can contain objects and/or inline addresses. See the "Guidelines and Limitations" section on page 28-2 for information about disallowed mapped IP addresses. For more information, see the "Configuring Objects" section on page 11-3.
Step 2	**object network** *obj_name* **Example:** hostname(config)# object network my-host-obj1	Configures a network object for which you want to configure NAT, or enters object network configuration mode for an existing network object.

	Command	Purpose
Step 3	{**host** *ip_address* | **subnet** *subnet_address netmask* | **range** *ip_address_1 ip_address_2*} **Example:** `hostname(config-network-object)# subnet 10.2.1.0 255.255.255.0`	If you are creating a new network object, defines the real IP address(es) that you want to translate.
Step 4	**nat** [(*real_ifc*,*mapped_ifc*)] **static** {*mapped_inline_ip* | *mapped_obj* | **interface**} [**dns** | **service** {**tcp** | **udp**} *real_port mapped_port*] **Example:** `hostname(config-network-object)# nat (inside,outside) static MAPPED_IPS service tcp 80 8080`	Configures static NAT for the object IP addresses. See the following guidelines: • Interfaces—If you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword **any** for one or both of the interfaces. Be sure to include the parentheses in your command. • Mapped IP Addresses—You can specify the mapped IP address as: – An inline IP address. The netmask or range for the mapped network is the same as that of the real network. For example, if the real network is a host, then this address will be a host address. In the case of a range, then the mapped addresses include the same number of addresses as the real range. For example, if the real address is defined as a range from 10.1.1.1 through 10.1.1.6, and you specify 172.20.1.1 as the mapped address, then the mapped range will include 172.20.1.1 through 172.20.1.6. – An existing network object (see Step 1). – An existing network object group (see Step 1). – **interface**—(Static NAT with port translation only) For this option, you must configure a specific interface for the *mapped_ifc*. Be sure to also configure the **service** keyword. (You cannot specify **interface** in transparent mode). Typically, you configure the same number of mapped addresses as real addresses for a one-to-one mapping. You can, however, have a mismatched number of addresses. For more information, see the "Static NAT" section on page 27-3. • DNS—The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See the "DNS and NAT" section on page 27-21 for more information. This option is not available if you specify the **service** keyword. • (Static NAT with port translation only) Port translation—Specify **tcp** or **udp** and the real and mapped ports. You can enter either a port number or a well-known port name (such as **ftp**). **Note** You can only define a single NAT rule for a given object. See the "Additional Guidelines" section on page 28-2.

Examples

The following example configures static NAT for the real host 10.1.1.1 on the inside to 10.2.2.2 on the outside with DNS rewrite enabled.

```
hostname(config)# object network my-host-obj1
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# nat (inside,outside) static 10.2.2.2 dns
```

The following example configures static NAT for the real host 10.1.1.1 on the inside to 2.2.2.2 on the outside using a mapped object.

```
hostname(config)# object network my-mapped-obj
hostname(config-network-object)# host 10.2.2.2

hostname(config-network-object)# object network my-host-obj1
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# nat (inside,outside) static my-mapped-obj
```

The following example configures static NAT with port translation for 10.1.1.1 at TCP port 21 to the outside interface at port 2121.

```
hostname(config)# object network my-ftp-server
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# nat (inside,outside) static interface service tcp 21 2121
```

Configuring Identity NAT

This section describes how to configure an identity NAT rule using network object NAT. For more information, see the "Identity NAT" section on page 27-11.

Detailed Steps

	Command	Purpose

Step 1	(Optional)
	object network *obj_name*
	{**host** *ip_address* |
	subnet *subnet_address netmask* |
	range *ip_address_1 ip_address_2*}
	
	Example:
	`hostname(config)# object network MAPPED_IPS`
	`hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0`
		For the mapped addresses (which will be the same as the real addresses), configure a network object.
		
		For more information, see the "Configuring Objects" section on page 11-3.

Step 2	**object network** *obj_name* **Example:** `hostname(config)# object network my-host-obj1`	Configures a network object for which you want to perform identity NAT, or enters object network configuration mode for an existing network object.

	Command	Purpose
Step 3	{**host** *ip_address* | **subnet** *subnet_address netmask* | **range** *ip_address_1 ip_address_2*} **Example:** `hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0`	If you are creating a new network object, defines the real IP address(es) to which you want to perform identity NAT.
Step 4	**nat** [(*real_ifc*,*mapped_ifc*)] **static** {*mapped_inline_ip* | *mapped_obj*} **Example:** `hostname(config-network-object)# nat (inside,outside) static MAPPED_IPS`	Configures identity NAT for the object IP addresses. See the following guidelines: • If you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword **any** for one or both of the interfaces. Be sure to include the parentheses in your command. • Be sure to configure the same IP address for both the mapped and real address. Typically, you will enter the address as an inline address, but you can create a second network object with the same IP address defined and use that object (see Step 1). **Note** You can only define a single NAT rule for a given object. See the "Additional Guidelines" section on page 28-2.

Example

The following example maps a host address to itself using an inline mapped address:

```
hostname(config)# object network my-host-obj1
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# nat (inside,outside) static 10.1.1.1
```

The following example maps a host address to itself using a network object:

```
hostname(config)# object network my-host-obj1-identity
hostname(config-network-object)# host 10.1.1.1

hostname(config-network-object)# object network my-host-obj1
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# nat (inside,outside) static my-host-obj1-identity
```

Monitoring Network Object NAT

To monitor object NAT, enter one of the following commands:

Command	Purpose
show nat	Shows NAT statistics, including hits for each NAT rule.
show nat pool	Shows NAT pool statistics, including the addresses and ports allocated, and how many times they were allocated.

Command	Purpose

show running-config nat	Shows the NAT configuration.
	
	Note You cannot view the NAT configuration using the **show running-config object** command. You cannot reference objects or object groups that have not yet been created in **nat** commands. To avoid forward or circular references in **show** command output, the **show running-config** command shows the **object** command two times: first, where the IP address(es) are defined; and later, where the **nat** command is defined. This command output guarantees that objects are defined first, then object groups, and finally NAT. For example:
	
	`hostname# show running-config`
	`...`
	`object network obj1`
	`  range 192.168.49.1 192.150.49.100`
	`object network obj2`
	`  object 192.168.49.100`
	`object network network-1`
	`  subnet <network-1>`
	`object network network-2`
	`  subnet <network-2>`
	`object-group network pool`
	`  network-object object obj1`
	`  network-object object obj2`
	`...`
	`object network network-1`
	`  nat (inside,outside) dynamic pool`
	`object network network-2`
	`  nat (inside,outside) dynamic pool`

show xlate	Shows current NAT session information.

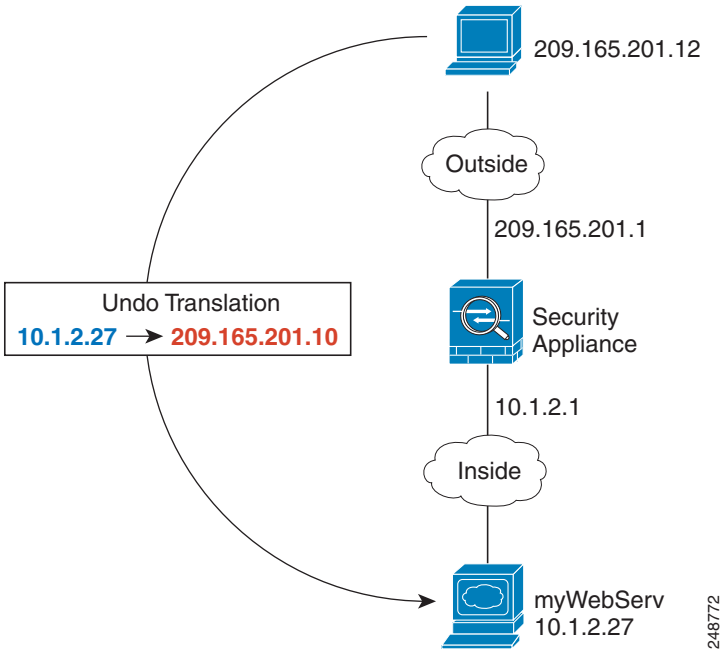
Configuration Examples for Network Object NAT

This section includes the following configuration examples:

Providing Access to an Inside Web Server (Static NAT)

The following example performs static NAT for an inside web server. The real address is on a private network, so a public address is required. Static NAT is necessary so hosts can initiate traffic to the web server at a fixed address. (See Figure 28-1).

Figure 28-1 Static NAT for an Inside Web Server



Step 1 Create a network object for the internal web server:

```
hostname(config)# object network myWebServ
```

Step 2 Define the web server address:

```
hostname(config-network-object)# host 10.1.2.27
```

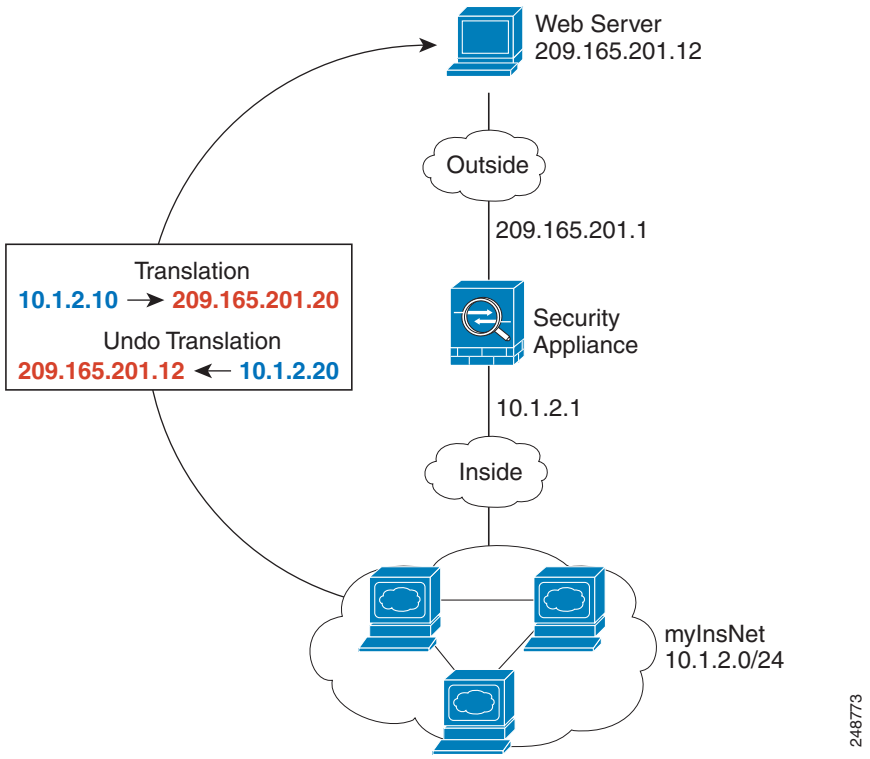
Step 3 Configure static NAT for the object:

```
hostname(config-network-object)# nat (inside,outside) static 209.165.201.10
```

NAT for Inside Hosts (Dynamic NAT) and NAT for an Outside Web Server (Static NAT)

The following example configures dynamic NAT for inside users on a private network when they access the outside. Also, when inside users connect to an outside web server, that web server address is translated to an address that appears to be on the inside network. (See Figure 28-2).

Figure 28-2 Dynamic NAT for Inside, Static NAT for Outside Web Server



Step 1 Create a network object for the dynamic NAT pool to which you want to translate the inside addresses:

```
hostname(config)# object network myNatPool
hostname(config-network-object)# range 209.165.201.20 209.165.201.30
```

Step 2 Create a network object for the inside network:

```
hostname(config)# object network myInsNet
hostname(config-network-object)# subnet 10.1.2.0 255.255.255.0
```

Step 3 Enable dynamic NAT for the inside network:

```
hostname(config-network-object)# nat (inside,outside) dynamic myNatPool
```

Step 4 Create a network object for the outside web server:

```
hostname(config)# object network myWebServ
```

Step 5 Define the web server address:

```
hostname(config-network-object)# host 209.165.201.12
```

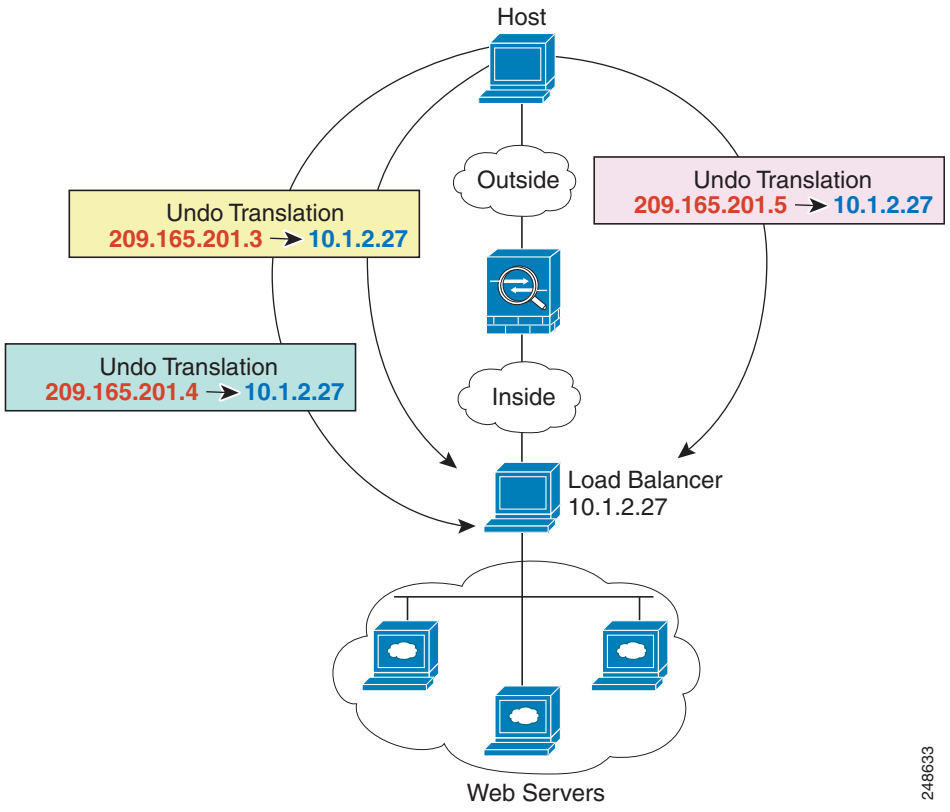
Step 6 Configure static NAT for the web server:

```
hostname(config-network-object)# nat (outside,inside) static 10.1.2.20
```

Inside Load Balancer with Multiple Mapped Addresses (Static NAT, One-to-Many)

The following example shows an inside load balancer that is translated to multiple IP addresses. When an outside host accesses one of the mapped IP addresses, it is untranslated to the single load balancer address. Depending on the URL requested, it redirects traffic to the correct web server. (See Figure 28-3).

Figure 28-3 Static NAT with One-to-Many for an Inside Load Balancer



Step 1 Create a network object for the addresses to which you want to map the load balancer:

```
hostname(config)# object network myPublicIPs
hostname(config-network-object)# range 209.165.201.3 209.265.201.8
```

Step 2 Create a network object for the load balancer:

```
hostname(config)# object network myLBHost
```

Step 3 Define the load balancer address:

```
hostname(config-network-object)# host 10.1.2.27
```

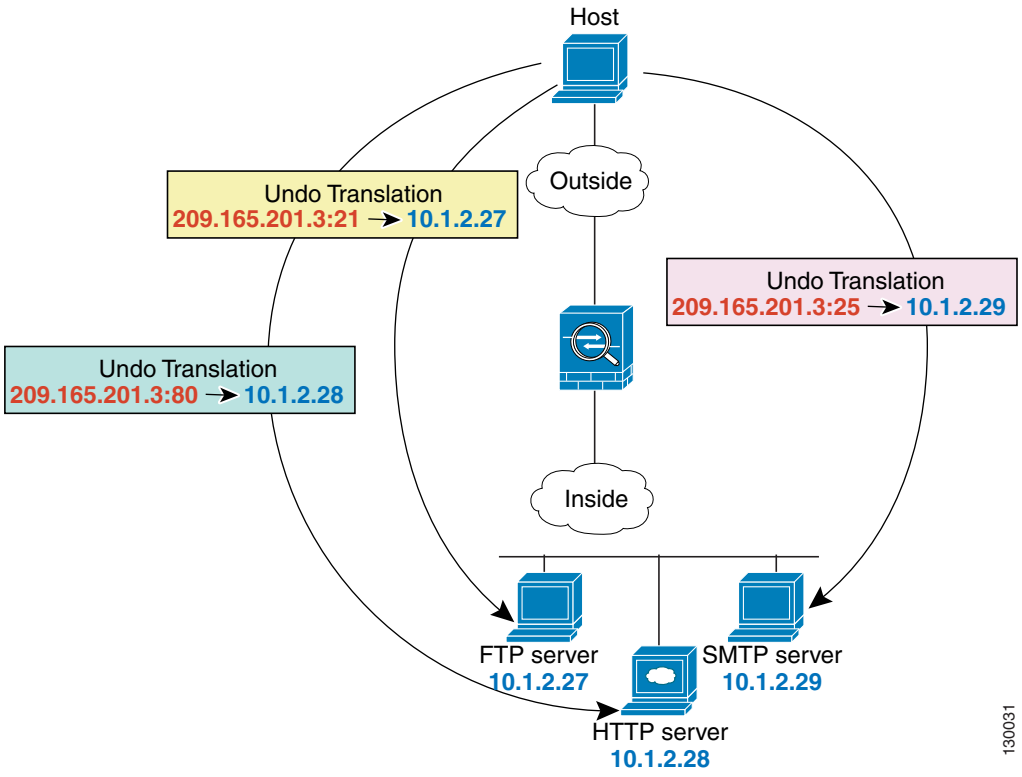
Step 4 Configure static NAT for the load balancer:

```
hostname(config-network-object)# nat (inside,outside) static myPublicIPs
```

Single Address for FTP, HTTP, and SMTP (Static NAT with Port Translation)

The following static NAT with port translation example provides a single address for remote users to access FTP, HTTP, and SMTP. These servers are actually different devices on the real network, but for each server, you can specify static NAT with port translation rules that use the same mapped IP address, but different ports. (See Figure 28-4.)

Figure 28-4 Static NAT with Port Translation



Step 1 Create a network object for the FTP server address:

```
hostname(config)# object network FTP_SERVER
```

Step 2 Define the FTP server address, and configure static NAT with identity port translation for the FTP server:

```
hostname(config-network-object)# host 10.1.2.27
hostname(config-network-object)# nat (inside,outside) static 209.165.201.3 service tcp ftp
ftp
```

Step 3 Create a network object for the HTTP server address:

```
hostname(config)# object network HTTP_SERVER
```

Step 4 Define the HTTP server address, and configure static NAT with identity port translation for the HTTP server:

```
hostname(config-network-object)# host 10.1.2.28
hostname(config-network-object)# nat (inside,outside) static 209.165.201.3 service tcp
http http
```

Step 5 Create a network object for the SMTP server address:

```
hostname(config)# object network SMTP_SERVER
```

Step 6 Define the SMTP server address, and configure static NAT with identity port translation for the SMTP server:

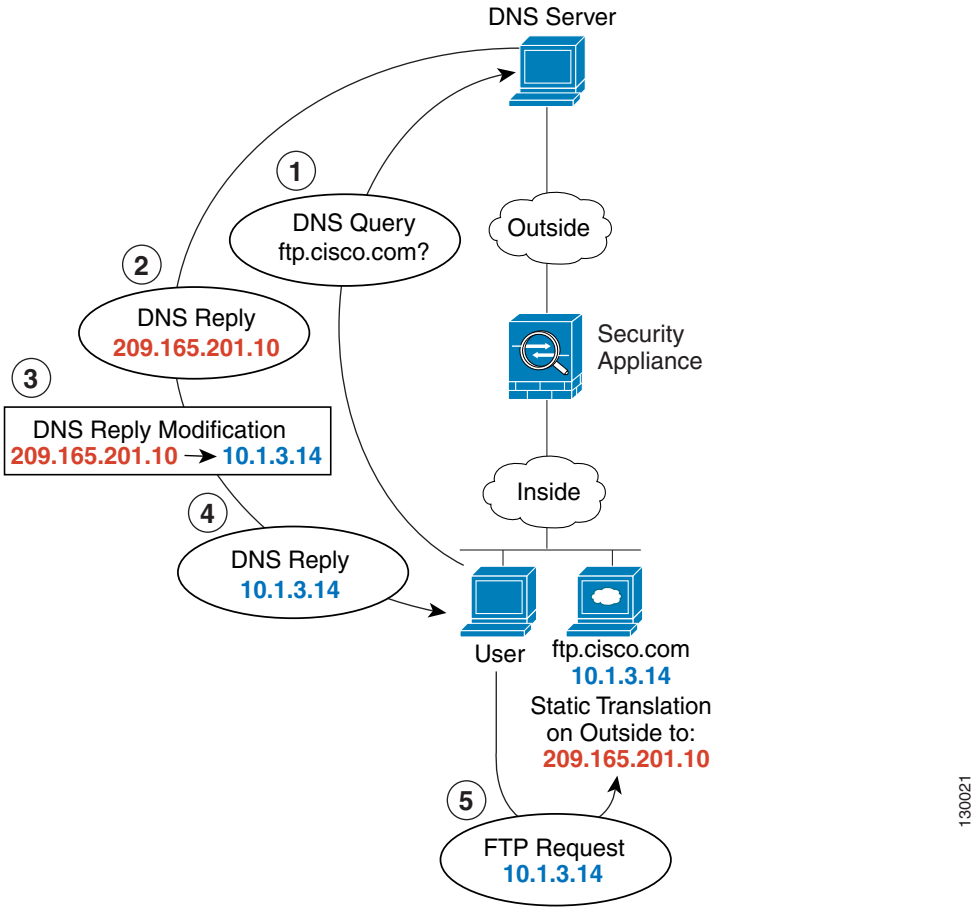
```
hostname(config-network-object)# host 10.1.2.29
hostname(config-network-object)# nat (inside,outside) static 209.165.201.3 service tcp
smtp smtp
```

DNS Server on Mapped Interface, Web Server on Real Interface (Static NAT with DNS Modification)

For example, a DNS server is accessible from the outside interface. A server, ftp.cisco.com, is on the inside interface. You configure the adaptive security appliance to statically translate the ftp.cisco.com real address (10.1.3.14) to a mapped address (209.165.201.10) that is visible on the outside network. (See Figure 28-5.) In this case, you want to enable DNS reply modification on this static rule so that inside users who have access to ftp.cisco.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.cisco.com, the DNS server replies with the mapped address (209.165.201.10). The adaptive security appliance refers to the static rule 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.cisco.com directly.

Figure 28-5 DNS Reply Modification



Step 1 Create a network object for the FTP server address:

```
hostname(config)# object network FTP_SERVER
```

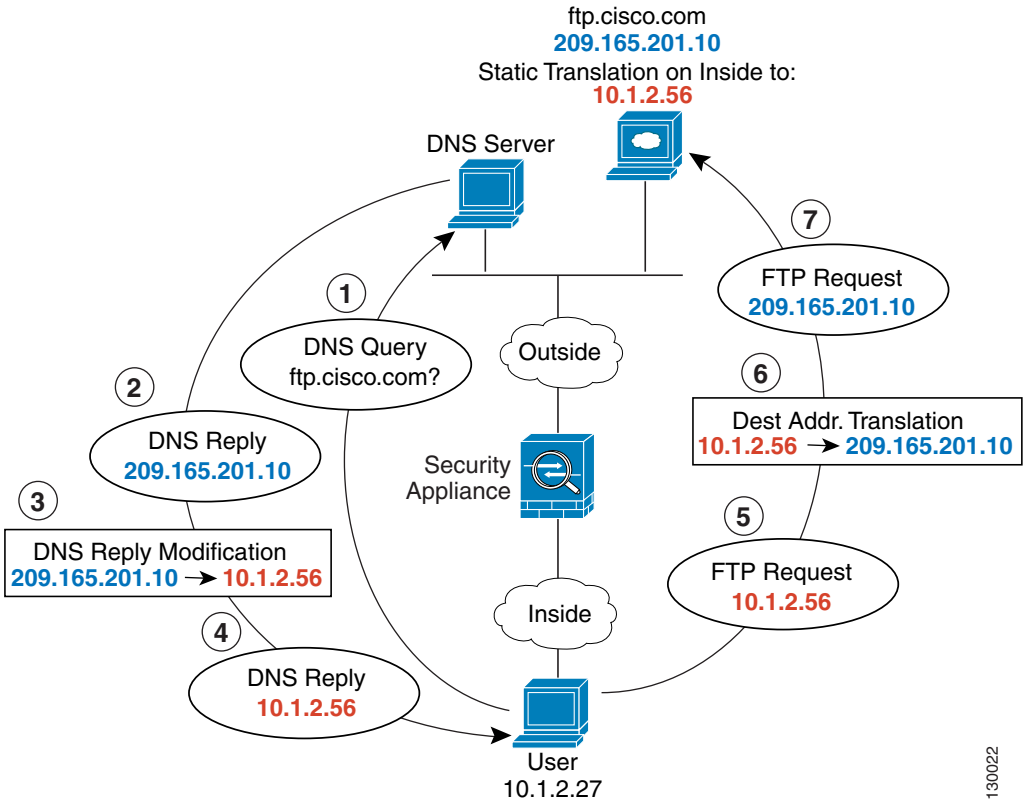
Step 2 Define the FTP server address, and configure static NAT with DNS modification:

```
hostname(config-network-object)# host 10.1.3.14
hostname(config-network-object)# nat (inside,outside) static 209.165.201.10 dns
```

DNS Server and Web Server on Mapped Interface, Web Server is Translated (Static NAT with DNS Modification)

Figure 28-6 shows a web server and DNS server on the outside. The adaptive security appliance has a static translation for the outside server. In this case, when an inside user requests the address for ftp.cisco.com from the DNS server, the DNS server responds with the real address, 209.165.20.10. Because you want inside users to use the mapped address for ftp.cisco.com (10.1.2.56) you need to configure DNS reply modification for the static translation.

Figure 28-6 DNS Reply Modification Using Outside NAT



Step 1 Create a network object for the FTP server address:

```
hostname(config)# object network FTP_SERVER
```

Step 2 Define the FTP server address, and configure static NAT with DNS modification:

```
hostname(config-network-object)# host 209.165.201.10
hostname(config-network-object)# nat (outside,inside) static 10.1.2.56 dns
```

Feature History for Network Object NAT

Table 28-1 lists each feature change and the platform release in which it was implemented.

Table 28-1 Feature History for Network Object NAT

Feature Name	Platform Releases	Feature Information
Network Object NAT	8.3(1)	Configures NAT for a network object IP address(es). The following commands were introduced or modified: **nat** (object network configuration mode), **show nat**, **show xlate**, **show nat pool**.