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 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 4, "Information About NAT."

Information About Network Object NAT

When a packet enters the ASA, 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 How NAT is Implemented, page 4-13.

Network object NAT rules are added to section 2 of the NAT rules table. For more information about NAT ordering, see NAT Rule Order, page 4-18.

Licensing Requirements for Network Object NAT

The following table shows the licensing requirements for this feature:

Model	License Requirement
ASAv	Standard or Premium License.
All other 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 separate 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 general operations configuration guide.

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, page 5-2 section.

Guidelines and Limitations

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.
- In transparent mode, translating between IPv4 and IPv6 networks is not supported. Translating between two IPv6 networks, or between two IPv4 networks is supported.

IPv6 Guidelines

- Supports IPv6. See also the NAT and IPv6, page 4-13.
- For routed mode, you can also translate between IPv4 and IPv6.
- For transparent mode, translating between IPv4 and IPv6 networks is not supported. Translating between two IPv6 networks, or between two IPv4 networks is supported.
- For transparent mode, a PAT pool is not supported for IPv6.
- For static NAT, you can specify an IPv6 subnet up to /64. Larger subnets are not supported.
- When using FTP with NAT46, when an IPv4 FTP client connects to an IPv6 FTP server, the client must use either the extended passive mode (EPSV) or extended port mode (EPRT); PASV and PORT commands are not supported with IPv6.

Additional Guidelines

- You can only define a single NAT rule for a given object; if you want to configure multiple NAT rules for an object, you need to create multiple objects with different names 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 configuration 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.
- You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address.
- You can use the same mapped object or group in multiple NAT rules.
- 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.
- For application inspection limitations with NAT or PAT, see Default Settings and NAT Limitations, page 7-4 in Chapter 7, “Getting Started with Application Layer Protocol Inspection.”

Default Settings

- (Routed mode) The default real and mapped interface is Any, which applies the rule to all interfaces.
- The default behavior for identity NAT has proxy ARP enabled, matching other static NAT rules. You can disable proxy ARP if desired. See Routing NAT Packets, page 4-19 for more information.
- If you specify an optional interface, then the ASA uses the NAT configuration to determine the egress interface, but you have the option to always use a route lookup instead. See Routing NAT Packets, page 4-19 for more information.

Configuring Network Object NAT

This section describes how to configure network object NAT and includes the following topics:

Adding Network Objects for Mapped Addresses

For dynamic NAT, you must use an object or group for the mapped addresses. Other NAT types have the option of using inline addresses, or you can create an object or group according to this section. For more information about configuring a network object or group, see the general operations configuration guide.

Guidelines

- A network object group can contain objects and/or inline addresses of either IPv4 or IPv6 addresses. The group cannot contain both IPv4 and IPv6 addresses; it must contain one type only.
- See Guidelines and Limitations, page 5-2 for information about disallowed mapped IP addresses.
- Dynamic NAT:
  - You cannot use an inline address; you must configure a network object or group.
  - The object or group cannot contain a subnet; the object must define a range; the group can include hosts and ranges.
  - If a mapped network object contains both ranges and host IP addresses, then the ranges are used for dynamic NAT, and then the host IP addresses are used as a PAT fallback.
- Dynamic PAT (Hide):
  - Instead of using an object, you can optionally configure an inline host address or specify the interface address.
  - If you use an object, the object or group cannot contain a subnet; the object must define a host, or for a PAT pool, a range; the group (for a PAT pool) can include hosts and ranges.
- Static NAT or Static NAT with port translation:
  - Instead of using an object, you can configure an inline address or specify the interface address (for static NAT-with-port-translation).
  - If you use an object, the object or group can contain a host, range, or subnet.
- Identity NAT
  - Instead of using an object, you can configure an inline address.
  - If you use an object, the object must match the real addresses you want to translate.

Detailed Steps

Command	Purpose

object network *obj_name* {**host** *ip_address* | **range** *ip_address_1 ip_address_2* | **subnet** *subnet_address netmask*}
 	
Example:
`hostname(config)# object network TEST`
`hostname(config-network-object)# range 10.1.1.1 10.1.1.70`
	Adds a network object, either IPv4 or IPv6.

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 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`
	Adds a network object group, either IPv4 or IPv6.

Configuring Dynamic NAT

This section describes how to configure network object NAT for dynamic NAT. For more information, see Dynamic NAT, page 4-7.

Detailed Steps

	Command	Purpose
Step 1	`Create a network object or group for the mapped addresses.`	See Adding Network Objects for Mapped Addresses, page 5-4.
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) (either IPv4 or IPv6) that you want to translate.
Step 4	**nat** [(*real_ifc*,*mapped_ifc*)] **dynamic** *mapped_obj* [**interface** [**ipv6**]] [**dns**] **Example:** `hostname(config-network-object)# nat (inside,outside) dynamic MAPPED_IPS interface`	Configures **dynamic NAT** for the object IP addresses. **Note** You can only define a single NAT rule for a given object. See Additional Guidelines, page 5-3. See the following guidelines: • Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, 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. • Mapped IP address—Specify the mapped IP address as: – An existing network object (see Step 1). – An existing network object group (see Step 1). • Interface PAT fallback—(Optional) The **interface** keyword enables interface PAT fallback. After the mapped IP addresses are used up, then the IP address of the mapped interface is used. If you specify **ipv6**, then the IPv6 address of the interface is used. For this option, you must configure a specific interface for the *mapped_ifc*. (You cannot specify **interface** in transparent mode). • DNS—(Optional) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See DNS and NAT, page 4-28 for more information.

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 used 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
```

The following example configures dynamic NAT with dynamic PAT backup to translate IPv6 hosts to IPv4. Hosts on inside network 2001:DB8::/96 are mapped first to the IPv4_NAT_RANGE pool (209.165.201.1 to 209.165.201.30). After all addresses in the IPv4_NAT_RANGE pool are allocated, dynamic PAT is performed using the IPv4_PAT address (209.165.201.31). In the event that the PAT translations are also used up, dynamic PAT is performed using the outside interface address.

```
hostname(config)# object network IPv4_NAT_RANGE
hostname(config-network-object)# range 209.165.201.1 209.165.201.30

hostname(config-network-object)# object network IPv4_PAT
hostname(config-network-object)# host 209.165.201.31

hostname(config-network-object)# object-group network IPv4_GROUP
hostname(config-network-object)# network-object object IPv4_NAT_RANGE
hostname(config-network-object)# network-object object IPv4_PAT

hostname(config-network-object)# object network my_net_obj5
hostname(config-network-object)# subnet 2001:DB8::/96
hostname(config-network-object)# nat (inside,outside) dynamic IPv4_GROUP interface
```

Configuring Dynamic PAT (Hide)

This section describes how to configure network object NAT for dynamic PAT (hide). For more information, see Dynamic PAT, page 4-8.

Guidelines

For a PAT pool:

- If available, the real source port number is used for the mapped port. However, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool that can be used. (8.4(3) and later, not including 8.5(1) or 8.6(1)) If you have a lot of traffic that uses the lower port ranges, you can now specify a flat range of ports to be used instead of the three unequal-sized tiers: either 1024 to 65535, or 1 to 65535.
- If you use the same PAT pool object in two separate rules, then be sure to specify the same options for each rule. For example, if one rule specifies extended PAT and a flat range, then the other rule must also specify extended PAT and a flat range.

For extended PAT for a PAT pool:

- Many application inspections do not support extended PAT. See Default Settings and NAT Limitations, page 7-4 in Chapter 7, “Getting Started with Application Layer Protocol Inspection,” for a complete list of unsupported inspections.

- If you enable extended PAT for a dynamic PAT rule, then you cannot also use an address in the PAT pool as the PAT address in a separate static NAT-with-port-translation rule. For example, if the PAT pool includes 10.1.1.1, then you cannot create a static NAT-with-port-translation rule using 10.1.1.1 as the PAT address.
- If you use a PAT pool and specify an interface for fallback, you cannot specify extended PAT.
- For VoIP deployments that use ICE or TURN, do not use extended PAT. ICE and TURN rely on the PAT binding to be the same for all destinations.

For round robin for a PAT pool:

- If a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available. **Note**: This "stickiness" does not survive a failover. If the ASA fails over, then subsequent connections from a host may not use the initial IP address.
- Round robin, especially when combined with extended PAT, can consume a large amount of memory. Because NAT pools are created for every mapped protocol/IP address/port range, round robin results in a large number of concurrent NAT pools, which use memory. Extended PAT results in an even larger number of concurrent NAT pools.

Detailed Steps

	Command	Purpose
Step 1	(Optional) Create a network object or group for the mapped addresses.	See Adding Network Objects for Mapped Addresses, page 5-4.
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.
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) (either IPv4 or IPv6) that you want to translate.

	Command	Purpose
Step 4	**nat** [(*real_ifc*,*mapped_ifc*)] **dynamic** {*mapped_inline_host_ip* | *mapped_obj* | **pat-pool** *mapped_obj* [**round-robin**] [**extended**] [**flat** [**include-reserve**]] | **interface** [**ipv6**]} [**interface** [**ipv6**]] [**dns**] **Example:** `hostname(config-network-object)# nat (any,outside) dynamic interface`	Configures **dynamic PAT** for the object IP addresses. You can only define a single NAT rule for a given object. See Additional Guidelines, page 5-3. See the following guidelines: • Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, 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. • 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). – **pat-pool**—An existing network object or group that contains multiple addresses. – **interface**—(Routed mode only) The IP address of the mapped interface is used as the mapped address. If you specify **ipv6**, then the IPv6 address of the interface is used. 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. • For a PAT pool, you can specify one or more of the following options: – Round robin—The **round-robin** keyword enables round-robin address allocation for a PAT pool. Without round robin, by default all ports for a PAT address will be allocated before the next PAT address is used. The round-robin method assigns an address/port from each PAT address in the pool before returning to use the first address again, and then the second address, and so on. (continued)

Command	Purpose
	(continued) – Extended PAT—The **extended** keyword enables extended PAT. Extended PAT uses 65535 ports per *service*, as opposed to per IP address, by including the destination address and port in the translation information. Normally, the destination port and address are not considered when creating PAT translations, so you are limited to 65535 ports per PAT address. For example, with extended PAT, you can create a translation of 10.1.1.1:1027 when going to 192.168.1.7:23 as well as a translation of 10.1.1.1:1027 when going to 192.168.1.7:80. – Flat range—The **flat** keyword enables use of the entire 1024 to 65535 port range when allocating ports. When choosing the mapped port number for a translation, the ASA uses the real source port number if it is available. However, without this option, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 1 to 511, 512 to 1023, and 1024 to 65535. To avoid running out of ports at the low ranges, configure this setting. To use the entire range of 1 to 65535, also specify the **include-reserve** keyword. • Interface PAT fallback—(Optional) The **interface** keyword enables interface PAT fallback when entered after a primary PAT address. After the primary PAT address(es) are used up, then the IP address of the mapped interface is used. If you specify **ipv6**, then the IPv6 address of the interface is used. For this option, you must configure a specific interface for the *mapped_ifc*. (You cannot specify **interface** in transparent mode). • DNS—(Optional) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See DNS and NAT, page 4-28 for more information.

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
```

The following example configures dynamic PAT with a PAT pool to translate the inside IPv6 network to an outside IPv4 network:

```
hostname(config)# object network IPv4_POOL
hostname(config-network-object)# range 203.0.113.1 203.0.113.254
hostname(config)# object network IPv6_INSIDE
hostname(config-network-object)# subnet 2001:DB8::/96
hostname(config-network-object)# nat (inside,outside) dynamic pat-pool IPv4_POOL
```

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 Static NAT, page 4-3.

Detailed Steps

	Command	Purpose
Step 1	(Optional) Create a network object or group for the mapped addresses.	See Adding Network Objects for Mapped Addresses, page 5-4.
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) (IPv4 or IPv6) that you want to translate.

Command	Purpose
Step 4 **nat** [(*real_ifc*,*mapped_ifc*)] **static** {*mapped_inline_ip* | *mapped_obj* | **interface** [**ipv6**]} [**net-to-net**] [**dns** | **service** {**tcp** | **udp**} *real_port mapped_port*] [**no-proxy-arp**] **Example:** `hostname(config-network-object)# nat (inside,outside) static MAPPED_IPS service tcp 80 8080`	Configures **static NAT** for the object IP addresses. You can only define a single NAT rule for a given object. • Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, 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. • 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 or group (see Step 1). – **interface**—(Static NAT-with-port-translation only; routed mode) For this option, you must configure a specific interface for the *mapped_ifc*. If you specify **ipv6**, then the IPv6 address of the interface is used. Be sure to also configure the **service** keyword. 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. See Static NAT, page 4-3. • Net-to-net—(Optional) For NAT 46, specify **net-to-net** to translate the first IPv4 address to the first IPv6 address, the second to the second, and so on. Without this option, the IPv4-embedded method is used. For a one-to-one translation, you must use this keyword. • DNS—(Optional) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See DNS and NAT, page 4-28. This option is not available if you specify the **service** keyword. • Port translation—(Static NAT-with-port-translation only) 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**). • No Proxy ARP—(Optional) Specify **no-proxy-arp** to disable proxy ARP for incoming packets to the mapped IP addresses. See Mapped Addresses and Routing, page 4-20 for more information.

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 10.2.2.2 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
```

The following example maps an inside IPv4 network to an outside IPv6 network.

```
hostname(config)# object network inside_v4_v6
hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) static 2001:DB8::/96
```

The following example maps an inside IPv6 network to an outside IPv6 network.

```
hostname(config)# object network inside_v6
hostname(config-network-object)# subnet 2001:DB8:AAAA::/96
hostname(config-network-object)# nat (inside,outside) static 2001:DB8:BBBB::/96
```

Configuring Identity NAT

This section describes how to configure an identity NAT rule using network object NAT. For more information, see Identity NAT, page 4-10.

Detailed Steps

	Command	Purpose
Step 1	(Optional) Create a network object for the mapped addresses.	The object must include the same addresses that you want to translate. See Adding Network Objects for Mapped Addresses, page 5-4.
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. This network object has a different name from the mapped network object (see Step 1) even though they both contain the same IP addresses.

	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) (IPv4 or IPv6) to which you want to perform identity NAT. If you configured a network object for the mapped addresses in Step 1, then these addresses must match.
Step 4	**nat** [(*real_ifc*,*mapped_ifc*)] **static** {*mapped_inline_ip* | *mapped_obj*} [**no-proxy-arp**] [**route-lookup**] **Example:** `hostname(config-network-object)# nat (inside,outside) static MAPPED_IPS`	Configures **identity NAT** for the object IP addresses. **Note** You can only define a single NAT rule for a given object. See Additional Guidelines, page 5-3. See the following guidelines: • Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, 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. • Mapped IP addresses—Be sure to configure the same IP address for both the mapped and real address. Use one of the following: – Network object—Including the same IP address as the real object (see Step 1). – 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 10.1.1.1 as the mapped address, then the mapped range will include 10.1.1.1 through 10.1.1.6. • No Proxy ARP—Specify **no-proxy-arp** to disable proxy ARP for incoming packets to the mapped IP addresses. See Mapped Addresses and Routing, page 4-20 for more information. • Route lookup—(Routed mode only; interface(s) specified) Specify **route-lookup** to determine the egress interface using a route lookup instead of using the interface specified in the NAT command. See Determining the Egress Interface, page 4-22 for more information.

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
```

Configuring Per-Session PAT Rules

By default, all TCP PAT traffic and all UDP DNS traffic uses per-session PAT. To use multi-session PAT for traffic, you can configure per-session PAT rules: a permit rule uses per-session PAT, and a deny rule uses multi-session PAT. For more information about per-session vs. multi-session PAT, see Per-Session PAT vs. Multi-Session PAT, page 4-9.

Defaults

By default, the following rules are installed:

```
xlate per-session permit tcp any4 any4
xlate per-session permit tcp any4 any6
xlate per-session permit tcp any6 any4
xlate per-session permit tcp any6 any6
xlate per-session permit udp any4 any4 eq domain
xlate per-session permit udp any4 any6 eq domain
xlate per-session permit udp any6 any4 eq domain
xlate per-session permit udp any6 any6 eq domain
```

Note You cannot remove these rules, and they always exist after any manually-created rules. Because rules are evaluated in order, you can override the default rules. For example, to completely negate these rules, you could add the following:

```
xlate per-session deny tcp any4 any4
xlate per-session deny tcp any4 any6
xlate per-session deny tcp any6 any4
xlate per-session deny tcp any6 any6
xlate per-session deny udp any4 any4 eq domain
xlate per-session deny udp any4 any6 eq domain
xlate per-session deny udp any6 any4 eq domain
xlate per-session deny udp any6 any6 eq domain
```

Detailed Steps

Command	Purpose
xlate per-session {**permit** | **deny**} {**tcp** | **udp**} *source_ip* [*operator src_port*] *destination_ip operator dest_port* **Example:** `hostname(config)# xlate per-session deny tcp any4 209.165.201.3 eq 1720`	Creates a permit or deny rule. This rule is placed above the default rules, but below any other manually-created rules. Be sure to create your rules in the order you want them applied. For the source and destination IP addresses, you can configure the following: • **host** *ip_address*—Specifies an IPv4 host address. • *ip_address mask*—Specifies an IPv4 network address and subnet mask. • *ipv6-address/prefix-length*—Specifies an IPv6 host or network address and prefix. • **any4** and **any6**—**any4** specifies only IPv4 traffic; and **any6** specifies any6 traffic. The *operator* matches the port numbers used by the source or destination. The permitted operators are as follows: • lt—less than • gt—greater than • eq—equal to • neq—not equal to • range—an inclusive range of values. When you use this operator, specify two port numbers, for example: `range 100 200`

Examples

The following example creates a deny rule for H.323 traffic, so that it uses multi-session PAT:

```
hostname(config)# xlate per-session deny tcp any4 209.165.201.7 eq 1720
hostname(config)# xlate per-session deny udp any4 209.165.201.7 range 1718 1719
```

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.

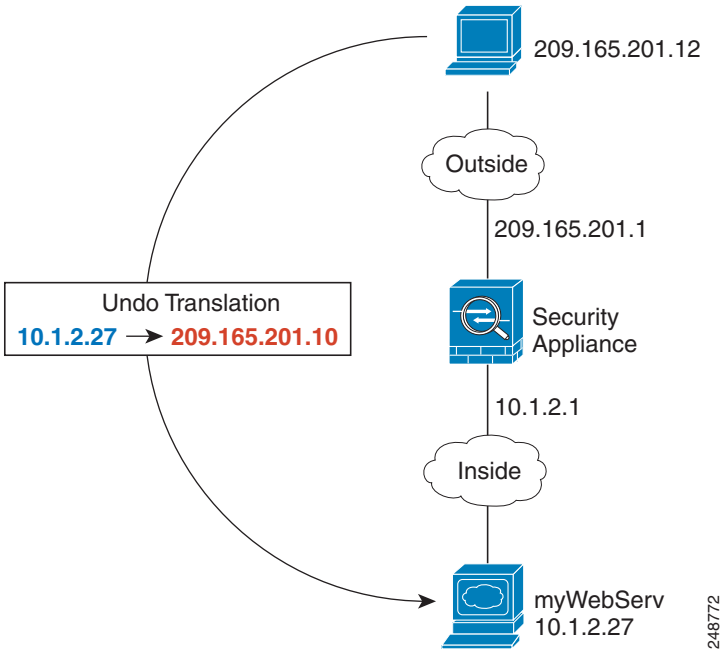
Configuration Examples for Network Object NAT

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 5-1).

Figure 5-1 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
```

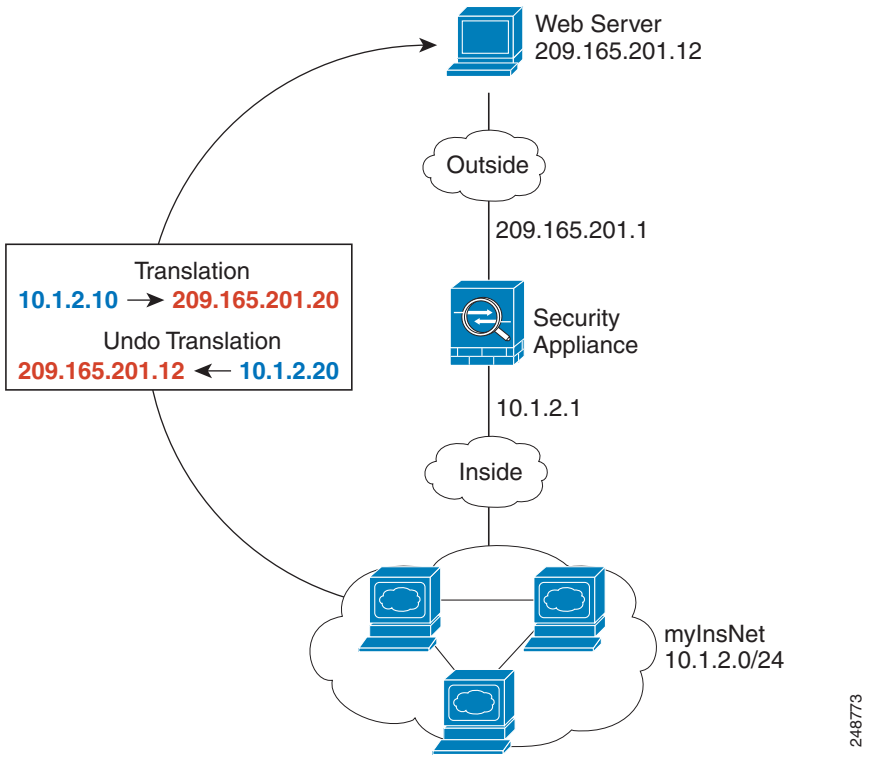
Step 3 Configure static NAT for the object:

```
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 5-2).

Figure 5-2 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
```

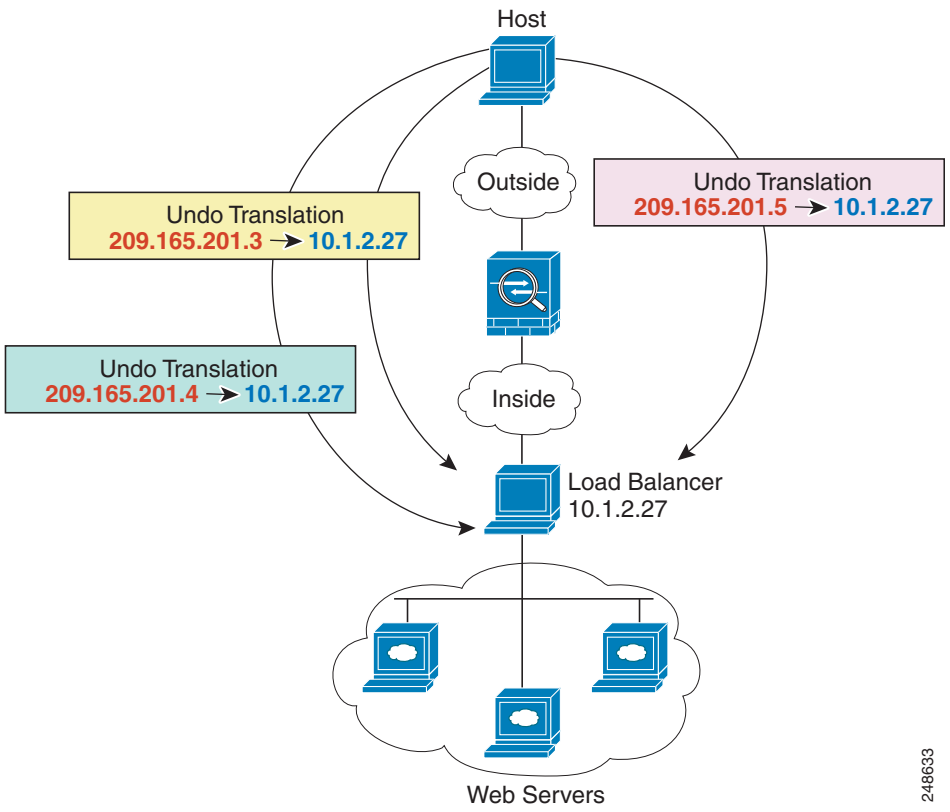
Step 6 Configure static NAT for the web server:

```
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 5-3).

Figure 5-3 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
```

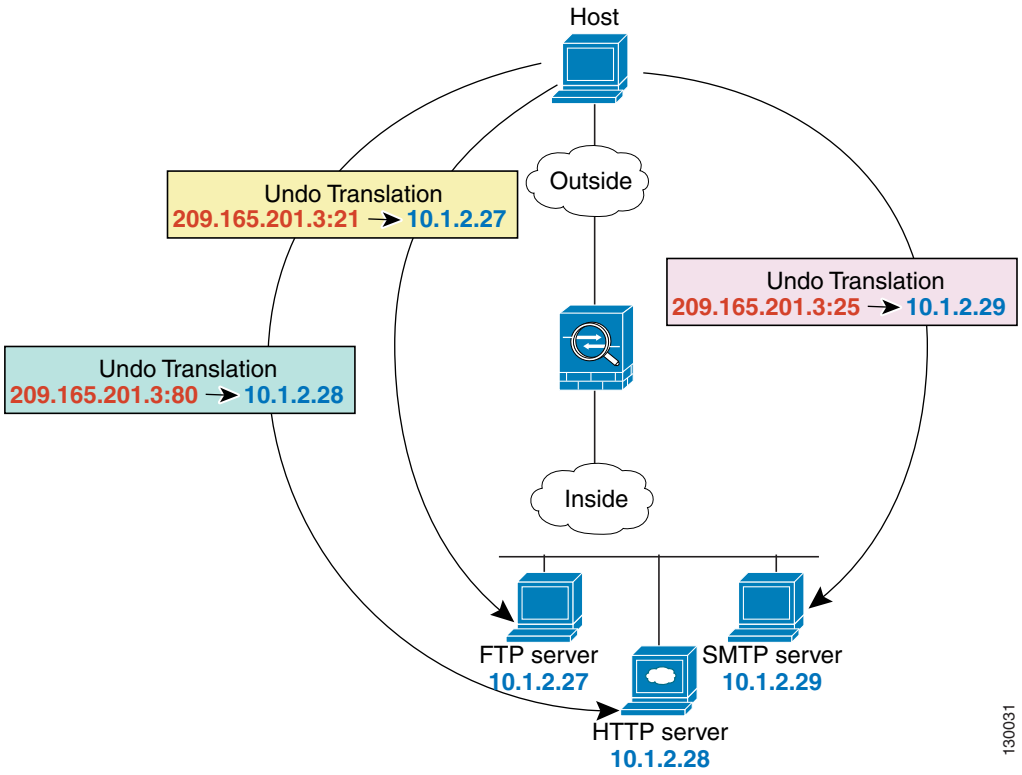
Step 4 Configure static NAT for the load balancer:

```
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 5-4.)

Figure 5-4 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:

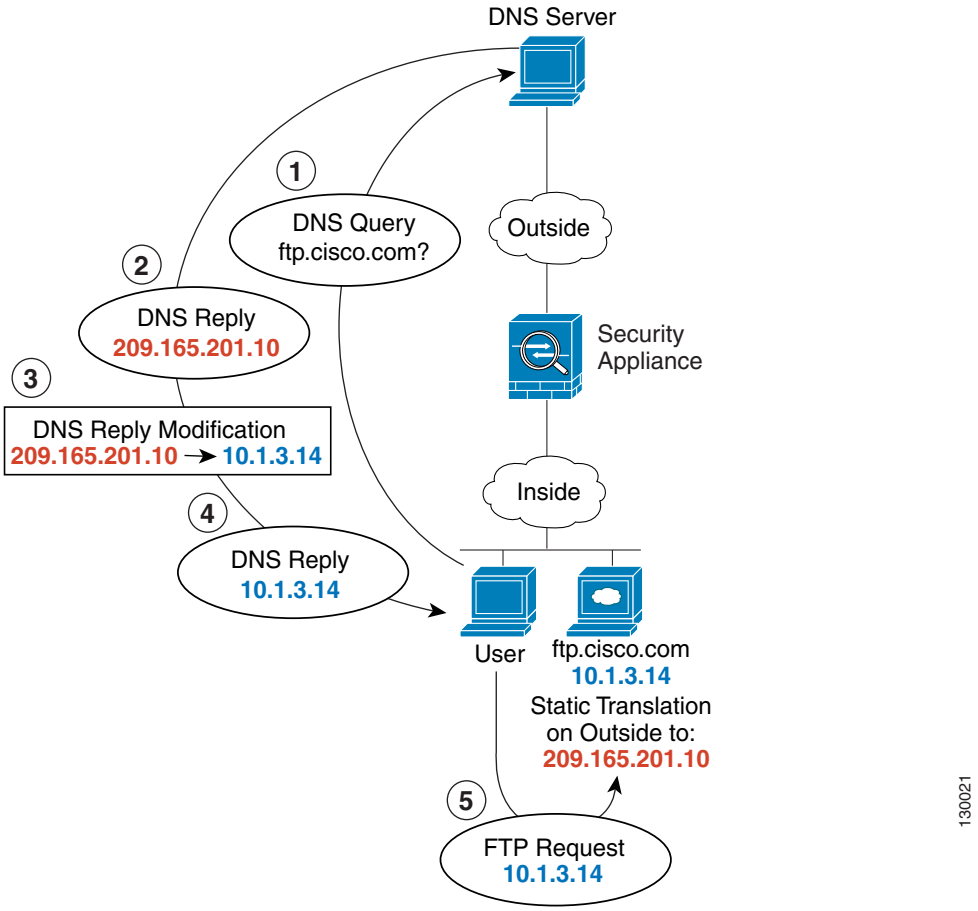
```
hostname(config-network-object)# host 10.1.2.29
hostname(config-network-object)# nat (inside,outside) static 209.165.201.3 service tcp
smtp smtp
```

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 ASA 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 5-5.) 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 ASA 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 5-5 DNS Reply Modification



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

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

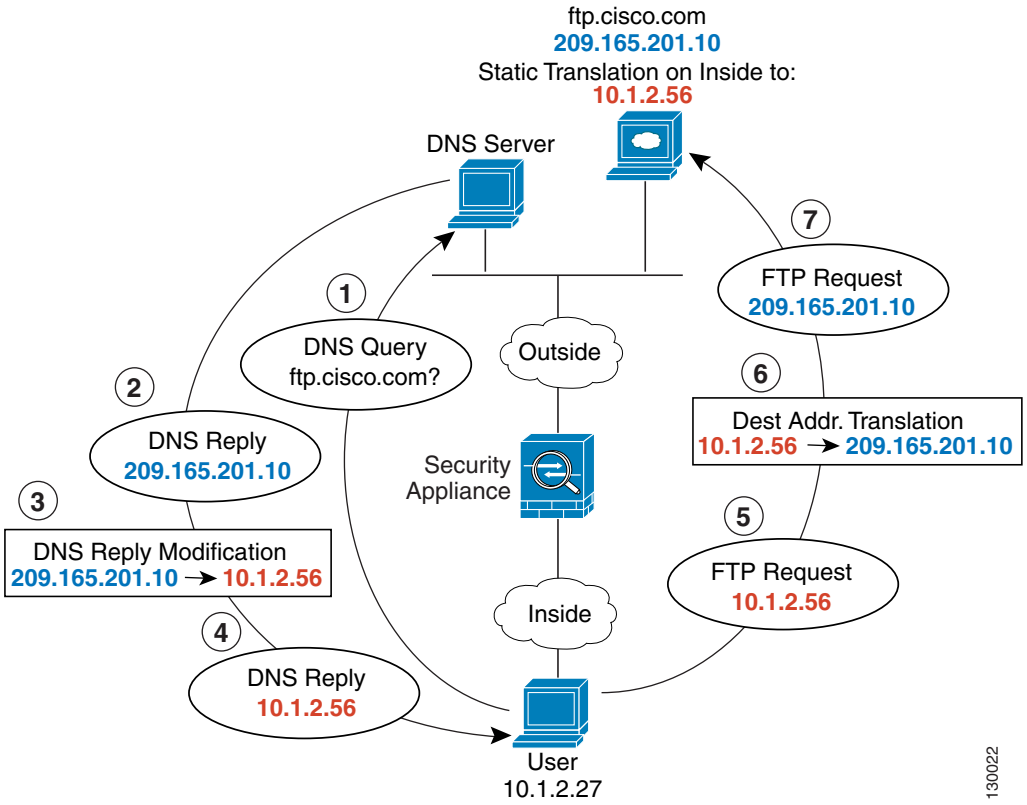
Step 2 Define the FTP server address, and configure static NAT with DNS modification:

```
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 FTP Server on Mapped Interface, FTP Server is Translated (Static NAT with DNS Modification)

Figure 5-6 shows an FTP server and DNS server on the outside. The ASA 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, 209.165.201.10. 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 5-6 DNS Reply Modification Using Outside NAT



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

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

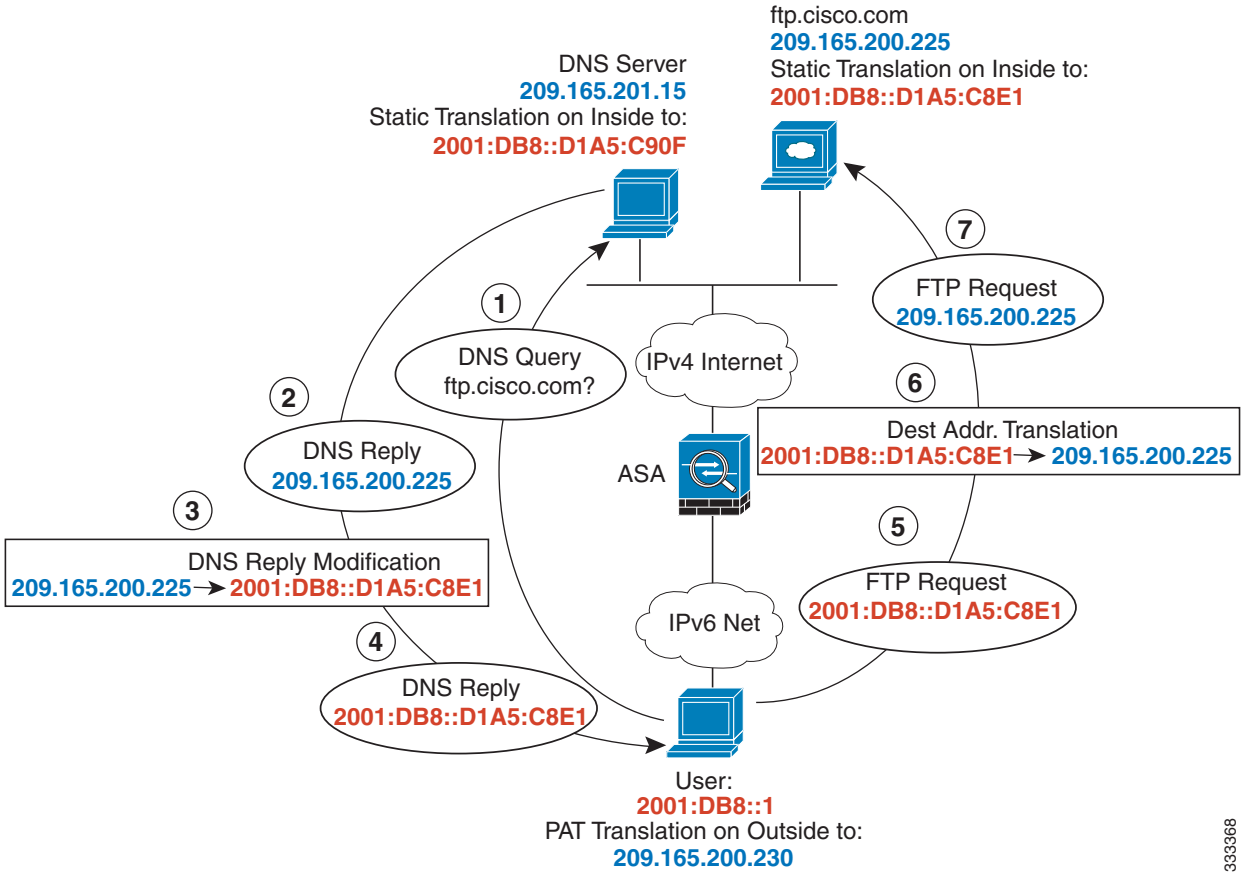
Step 2 Define the FTP server address, and configure static NAT with DNS modification:

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

IPv4 DNS Server and FTP Server on Mapped Interface, IPv6 Host on Real Interface (Static NAT64 with DNS64 Modification)

Figure 5-6 shows an FTP server and DNS server on the outside IPv4 network. The ASA has a static translation for the outside server. In this case, when an inside IPv6 user requests the address for ftp.cisco.com from the DNS server, the DNS server responds with the real address, 209.165.200.225. Because you want inside users to use the mapped address for ftp.cisco.com (2001:DB8::D1A5:C8E1) you need to configure DNS reply modification for the static translation. This example also includes a static NAT translation for the DNS server, and a PAT rule for the inside IPv6 hosts.

Figure 5-7 DNS Reply Modification Using Outside NAT



Step 1 Configure static NAT with DNS modification for the FTP server.

a. Create a network object for the FTP server address.

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

b. Define the FTP server address, and configure static NAT with DNS modification and, because this is a one-to-one translation, configure the net-to-net method for NAT46.

```
hostname(config-network-object)# host 209.165.200.225
hostname(config-network-object)# nat (outside,inside) static 2001:DB8::D1A5:C8E1/128
net-to-net dns
```

Step 2 Configure NAT for the DNS server.

a. Create a network object for the DNS server address.

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

b. Define the DNS server address, and configure static NAT using the net-to-net method.

```
hostname(config-network-object)# host 209.165.201.15
hostname(config-network-object)# nat (outside,inside) static 2001:DB8::D1A5:C90F/128
net-to-net
```

Step 3 Configure an IPv4 PAT pool for translating the inside IPv6 network.

```
hostname(config)# object network IPv4_POOL
hostname(config-network-object)# range 203.0.113.1 203.0.113.254
```

Step 4 Configure PAT for the inside IPv6 network.

a. Create a network object for the inside IPv6 network.

```
hostname(config)# object network IPv6_INSIDE
```

b. Define the IPv6 network address, and configure dynamic NAT using a PAT pool.

```
hostname(config-network-object)# subnet 2001:DB8::/96
hostname(config-network-object)# nat (inside,outside) dynamic pat-pool IPv4_POOL
```

Feature History for Network Object NAT

Table 5-1 lists each feature change and the platform release in which it was implemented.

Table 5-1 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). We introduced or modified the following commands: **nat** (object network configuration mode), **show nat**, **show xlate**, **show nat pool**.
Identity NAT configurable proxy ARP and route lookup	8.4(2)/8.5(1)	In earlier releases for identity NAT, proxy ARP was disabled, and a route lookup was always used to determine the egress interface. You could not configure these settings. In 8.4(2) and later, the default behavior for identity NAT was changed to match the behavior of other static NAT configurations: proxy ARP is enabled, and the NAT configuration determines the egress interface (if specified) by default. You can leave these settings as is, or you can enable or disable them discretely. Note that you can now also disable proxy ARP for regular static NAT. When upgrading to 8.4(2) from 8.3(1), 8.3(2), and 8.4(1), all identity NAT configurations will now include the **no-proxy-arp** and **route-lookup** keywords, to maintain existing functionality. We modified the following command: **nat static** [**no-proxy-arp**] [**route-lookup**].
PAT pool and round robin address assignment	8.4(2)/8.5(1)	You can now specify a pool of PAT addresses instead of a single address. You can also optionally enable round-robin assignment of PAT addresses instead of first using all ports on a PAT address before using the next address in the pool. These features help prevent a large number of connections from a single PAT address from appearing to be part of a DoS attack and makes configuration of large numbers of PAT addresses easy. We modifed the following command: **nat dynamic** [**pat-pool** *mapped_object* [**round-robin**]].
Round robin PAT pool allocation uses the same IP address for existing hosts	8.4(3)	When using a PAT pool with round robin allocation, if a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available. We did not modify any commands. *This feature is not available in 8.5(1) or 8.6(1).*

Table 5-1 Feature History for Network Object NAT (continued)

Feature Name	Platform Releases	Feature Information
Flat range of PAT ports for a PAT pool	8.4(3)	If available, the real source port number is used for the mapped port. However, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool. If you have a lot of traffic that uses the lower port ranges, when using a PAT pool, you can now specify a flat range of ports to be used instead of the three unequal-sized tiers: either 1024 to 65535, or 1 to 65535. We modifed the following command: **nat dynamic** [**pat-pool** *mapped_object* [**flat** [**include-reserve**]]]. *This feature is not available in 8.5(1) or 8.6(1).*
Extended PAT for a PAT pool	8.4(3)	Each PAT IP address allows up to 65535 ports. If 65535 ports do not provide enough translations, you can now enable extended PAT for a PAT pool. Extended PAT uses 65535 ports per *service*, as opposed to per IP address, by including the destination address and port in the translation information. We modifed the following command: **nat dynamic** [**pat-pool** *mapped_object* [**extended**]]. *This feature is not available in 8.5(1) or 8.6(1).*

Table 5-1 Feature History for Network Object NAT (continued)

Feature Name	Platform Releases	Feature Information
Automatic NAT rules to translate a VPN peer's local IP address back to the peer's real IP address	8.4(3)	In rare situations, you might want to use a VPN peer's real IP address on the inside network instead of an assigned local IP address. Normally with VPN, the peer is given an assigned local IP address to access the inside network. However, you might want to translate the local IP address back to the peer's real public IP address if, for example, your inside servers and network security is based on the peer's real IP address. You can enable this feature on one interface per tunnel group. Object NAT rules are dynamically added and deleted when the VPN session is established or disconnected. You can view the rules using the **show nat** command. **Note** Because of routing issues, we do not recommend using this feature unless you know you need this feature; contact Cisco TAC to confirm feature compatibility with your network. See the following limitations: • Only supports Cisco IPsec and AnyConnect Client. • Return traffic to the public IP addresses must be routed back to the ASA so the NAT policy and VPN policy can be applied. • Does not support load-balancing (because of routing issues). • Does not support roaming (public IP changing). We introduced the following command: **nat-assigned-to-public-ip** *interface* (tunnel-group general-attributes configuration mode).
NAT support for IPv6	9.0(1)	NAT now supports IPv6 traffic, as well as translating between IPv4 and IPv6. Translating between IPv4 and IPv6 is not supported in transparent mode. We modified the following commands: **nat** (object network configuration mode), **show nat**, **show nat pool**, **show xlate**.

Table 5-1 Feature History for Network Object NAT (continued)

Feature Name	Platform Releases	Feature Information
NAT support for reverse DNS lookups	9.0(1)	NAT now supports translation of the DNS PTR record for reverse DNS lookups when using IPv4 NAT, IPv6 NAT, and NAT64 with DNS inspection enabled for the NAT rule.
Per-session PAT	9.0(1)	The per-session PAT feature improves the scalability of PAT and, for clustering, allows each member unit to own PAT connections; multi-session PAT connections have to be forwarded to and owned by the master unit. At the end of a per-session PAT session, the ASA sends a reset and immediately removes the xlate. This reset causes the end node to immediately release the connection, avoiding the TIME_WAIT state. Multi-session PAT, on the other hand, uses the PAT timeout, by default 30 seconds. For "hit-and-run" traffic, such as HTTP or HTTPS, the per-session feature can dramatically increase the connection rate supported by one address. Without the per-session feature, the maximum connection rate for one address for an IP protocol is approximately 2000 per second. With the per-session feature, the connection rate for one address for an IP protocol is 65535/*average-lifetime*. By default, all TCP traffic and UDP DNS traffic use a per-session PAT xlate. For traffic that requires multi-session PAT, such as H.323, SIP, or Skinny, you can disable per-session PAT by creating a per-session deny rule. We introduced the following commands: **xlate per-session**, **show nat pool**.