CHAPTER 31

Configuring Twice NAT

Twice NAT lets you identify both the source and destination address in a single rule. This chapter shows you how to configure twice NAT and includes the following sections:

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

Information About Twice NAT

Twice NAT lets you identify both the source and destination address in a single rule. Specifying both the source and destination addresses lets you 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, for example.

Note For static NAT, the rule is bidirectional, so be aware that "source" and "destination" are used in commands and descriptions throughout this guide even though a given connection might originate at the "destination" address. For example, if you configure static NAT with port address translation, and specify the source address as a Telnet server, and you want all traffic going to that Telnet server to have the port translated from 2323 to 23, then in the command, you must specify the *source* ports to be translated (real: 23, mapped: 2323). You specify the source ports because you specified the Telnet server address as the source address.

The destination address is optional. If you specify the destination address, you can either map it to itself (identity NAT), or you can map it to a different address. The destination mapping is always a static mapping.

Twice NAT also lets you use service objects for static NAT-with-port-translation; network object NAT only accepts inline definition.

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

Twice NAT rules are added to section 1 of the NAT rules table, or if specified, section 3. For more information about NAT ordering, see the "NAT Rule Order" section on page 27-19.

Licensing Requirements for Twice NAT

Model	License Requirement
All models	Base License.

Prerequisites for Twice NAT

- For both the real and mapped addresses, configure network objects or network object groups (the **object network** or **object-group network** command). Network object groups are particularly useful for creating a mapped address pool with discontinuous IP address ranges or multiple hosts or subnets. To create a network object or group, see the "Configuring Objects and Groups" section on page 13-1.
- For static NAT-with-port-translation, configure TCP or UDP service objects (the **object service** command). To create a service object, see the "Configuring a Service Object" section on page 13-4.

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

- 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.
- You can use the same objects in multiple 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.

Default Settings

- By default, the rule is added to the end of section 1 of the NAT table.
- (Routed mode) The default real and mapped interface is Any, which applies the rule to all interfaces.
- (8.3(1), 8.3(2), and 8.4(1)) The default behavior for identity NAT has proxy ARP disabled. You cannot configure this setting. (8.4(2) and later) The default behavior for identity NAT has proxy ARP enabled, matching other static NAT rules. You can disable proxy ARP if desired.
- If you specify an optional interface, then the ASA uses the NAT configuration to determine the egress interface. (8.3(1) through 8.4(1)) The only exception is for identity NAT, which always uses a route lookup, regardless of the NAT configuration. (8.4(2) and later) For identity NAT, the default behavior is to use the NAT configuration, but you have the option to always use a route lookup instead.

Configuring Twice NAT

This section describes how to configure twice NAT. This section includes the following topics:

Configuring Dynamic NAT

This section describes how to configure twice NAT for dynamic 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*
	{**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 MyInsNet`
	`hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0`
		Configure the **real source** addresses.
		You can configure either a network object or a network object group. For more information, see the "Configuring Objects" section on page 13-3.
		If you want to translate all traffic, you can skip this step and specify the **any** keyword instead of creating an object or group.

Step 2	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 NAT_POOL`
	`hostname(config-network-object)# range 209.165.201.10 209.165.201.20`
		Configure the **mapped source** addresses.
		You can configure either a network object or a network object group.
		For dynamic NAT, you typically configure a larger group of addresses to be mapped to a smaller group. 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.
		Note The mapped object or group cannot contain a subnet.
		See the "Guidelines and Limitations" section on page 31-2 for information about disallowed mapped IP addresses.

	Command	Purpose

Step 3	(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 Server1`
	`hostname(config-network-object)# host 209.165.201.8`
		Configure the **real destination** addresses.
		You can configure either a network object or a network object group.
		Although the main feature of twice NAT is the inclusion of the destination IP address, the destination address is optional. If you do specify the destination address, you can configure static translation for that address or just use identity NAT for it. You might want to configure twice NAT without a destination address to take advantage of some of the other qualities of twice NAT, including the use of network object groups for real addresses, or manually ordering of rules. For more information, see the "Main Differences Between Network Object NAT and Twice NAT" section on page 27-15.

Step 4	(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 Server1_mapped`
	`hostname(config-network-object)# host 10.1.1.67`
		Configure the **mapped destination** addresses.
		The destination translation is always static. For identity NAT, you can skip this step and simply use the same object or group for both the real and mapped addresses.
		If you want to translate the destination address, you can configure either a network object or a network object group. The static mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired. For more information, see the "Static NAT" section on page 27-3.
		For static interface NAT with port translation (routed mode only), you can skip this step and specify the **interface** keyword instead of a network object/group for the mapped address. For more information, see the "Static Interface NAT with Port Translation" section on page 27-5.

Command	Purpose

Step 5 (Optional)
object service *obj_name*
service {**tcp** | **udp**} **destination** *operator port*
 	
Example:
`hostname(config)# object service REAL_SVC`
`hostname(config-service-object)# service tcp destination eq 80`
 	
`hostname(config)# object service MAPPED_SVC`
`hostname(config-service-object)# service tcp destination eq 8080`
	Configure service objects for:
	• **Destination real port**
	• **Destination mapped port**
	Dynamic NAT does not support port translation. However, because the *destination* translation is always static, you can perform port translation for the destination port. A service object can contain both a source and destination port, but only the destination port is used in this case. If you specify the source port, it will be ignored. NAT only supports TCP or UDP. When translating a port, be sure the protocols in the real and mapped service objects are identical (both TCP or both UDP). For identity NAT, you can use the same service object for both the real and mapped ports. The "not equal" (**neq**) operator is not supported.

	Command	Purpose
Step 6	**nat** [(*real_ifc*,*mapped_ifc*)] [*line* | {**after-auto** [*line*]}] **source dynamic** {*real_obj* | **any**} {*mapped_obj* [**interface**]} [**destination static** {*mapped_obj* | **interface**} *real_obj*] [**service** *mapped_dest_svc_obj real_dest_svc_obj*] [**dns**] [**inactive**] [**description** *desc*] **Example:** `hostname(config)# nat (inside,outside) source dynamic MyInsNet NAT_POOL destination static Server1_mapped Server1 service MAPPED_SVC REAL_SVC`	Configure **dynamic NAT**. 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. • Section and Line—(Optional) By default, the NAT rule is added to the end of section 1 of the NAT table (see the "NAT Rule Order" section on page 27-19). If you want to add the rule into section 3 instead (after the network object NAT rules), then use the **after-auto** keyword. You can insert a rule anywhere in the applicable section using the *line* argument. • Source addresses: – Real—Specify a network object, group, or the **any** keyword (see Step 1). Use the **any** keyword if you want to translate all traffic from the real interface to the mapped interface. – Mapped—Specify a different network object or group (see Step 2). You can optionally configure the following fallback method: Interface PAT fallback—(Routed mode only) 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. For this option, you must configure a specific interface for the *mapped_ifc*.

Command	Purpose
	(Continued) • Destination addresses (Optional): – Mapped—Specify a network object or group, or for static interface NAT with port translation only, specify the **interface** keyword (see Step 4). If you specify **interface**, be sure to also configure the **service** keyword. For this option, you must configure a specific interface for the *real_ifc*. See the "Static Interface NAT with Port Translation" section on page 27-5 for more information. – Real—Specify a network object or group (see Step 3). For identity NAT, simply use the same object or group for both the real and mapped addresses. • Destination port—(Optional) Specify the **service** keyword along with the mapped and real service objects (see Step 5). For identity port translation, simply use the same service object for both the real and mapped ports. • DNS—(Optional; for a source-only rule) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). You cannot configure the **dns** keyword if you configure a **destination** address. See the "DNS and NAT" section on page 27-29 for more information. • Inactive—(Optional) To make this rule inactive without having to remove the command, use the **inactive** keyword. To reactivate it, reenter the whole command without the **inactive** keyword. • Description—Optional) Provide a description up to 200 characters using the **description** keyword.

Configuring Dynamic PAT (Hide)

This section describes how to configure twice NAT for dynamic PAT (hide). For more information, see the "Dynamic PAT" section on page 27-10.

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.
- (8.4(3) and later, not including 8.5(1) or 8.6(1)) 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 (8.4(3) and later, not including 8.5(1) or 8.6(1)):

- Many application inspections do not support extended PAT. See the "Default Settings" section on page 39-4 in Chapter 39, "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:

- (8.4(3) and later, not including 8.5(1) or 8.6(1)) 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.
- (8.4(2), 8.5(1), and 8.6(1)) If a host has an existing connection, then subsequent connections from that host will likely use *different* PAT addresses for each connection because of the round robin allocation. In this case, you may have problems when accessing two websites that exchange information about the host, for example an e-commerce site and a payment site. When these sites see two different IP addresses for what is supposed to be a single host, the transaction may fail.
- 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	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 MyInsNet`
	`hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0`
		Configure the **real source** addresses.
		
		You can configure either a network object or a network object group. For more information, see the "Configuring Objects" section on page 13-3.
		
		If you want to translate all traffic, you can skip this step and specify the **any** keyword instead of creating an object or group.

	Command	Purpose
Step 2	Network object: **object network** *obj_name* {**host** *ip_address* | **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 PAT_POOL1 hostname(config-network-object)# range 10.5.1.80 10.7.1.80 hostname(config)# object network PAT_POOL2 hostname(config-network-object)# range 10.9.1.1 10.10.1.1 hostname(config)# object network PAT_IP hostname(config-network-object)# host 10.5.1.79 hostname(config-network-object)# object-group network PAT_POOLS hostname(config-network)# network-object object PAT_POOL1 hostname(config-network)# network-object object PAT_POOL2 hostname(config-network)# network-object object PAT_IP	Specify the **mapped** address(es) (that you want to translate to). You can configure a single address or, for a PAT pool, multiple addresses. Configure a network object or network object group. A network object group can contain objects and/or inline addresses. Alternatively, you can skip this step if you want to enter a single IP address as an inline value for the **nat** command or if you want to use the interface address by specifying the **interface** keyword. For mapped addresses used as a PAT pool, all addresses in the object or group, including ranges, are used as PAT addresses. **Note** The object or group cannot contain a subnet. See the "Guidelines and Limitations" section on page 31-2 for information about disallowed mapped IP addresses. For more information about configuring a network object or group, see the "Configuring Objects" section on page 13-3.
Step 3	(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 Server1 hostname(config-network-object)# host 209.165.201.8	Configure the **real destination** addresses. You can configure either a network object or a network object group. Although the main feature of twice NAT is the inclusion of the destination IP address, the destination address is optional. If you do specify the destination address, you can configure static translation for that address or just use identity NAT for it. You might want to configure twice NAT without a destination address to take advantage of some of the other qualities of twice NAT, including the use of network object groups for real addresses, or manually ordering of rules. For more information, see the "Main Differences Between Network Object NAT and Twice NAT" section on page 27-15.

	Command	Purpose

Step 4	(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 Server1_mapped`
	`hostname(config-network-object)# host 10.1.1.67`
		Configure the **mapped destination** addresses.
		The destination translation is always static. For identity NAT, you can skip this step and simply use the same object or group for both the real and mapped addresses.
		If you want to translate the destination address, you can configure either a network object or a network object group. The static mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired. For more information, see the "Static NAT" section on page 27-3.
		For static interface NAT with port translation (routed mode only), you can skip this step and specify the **interface** keyword instead of a network object/group for the mapped address. For more information, see the "Static Interface NAT with Port Translation" section on page 27-5.

Step 5	(Optional)
	object service *obj_name*
	service {**tcp** | **udp**} **destination** *operator port*
	Example:
	`hostname(config)# object service REAL_SVC`
	`hostname(config-service-object)# service tcp destination eq 80`
	
	`hostname(config)# object service MAPPED_SVC`
	`hostname(config-service-object)# service tcp destination eq 8080`
		Configure service objects for:
		• **Destination real port**
		• **Destination mapped port**
		Dynamic PAT does not support additional port translation. However, because the *destination* translation is always static, you can perform port translation for the destination port. A service object can contain both a source and destination port, but only the destination port is used in this case. If you specify the source port, it will be ignored. NAT only supports TCP or UDP. When translating a port, be sure the protocols in the real and mapped service objects are identical (both TCP or both UDP). For identity NAT, you can use the same service object for both the real and mapped ports. The "not equal" (**neq**) operator is not supported.

	Command	Purpose
Step 6	**nat** [(*real_ifc*,*mapped_ifc*)] [*line* | {**after-auto** [*line*]}] **source dynamic** {*real-obj* | **any**} {*mapped_obj* [**interface**] | [**pat-pool** *mapped_obj* [**round-robin**] [**extended**] [**flat** [**include-reserve**]] [**interface**] | **interface**} [**destination static** {*mapped_obj* | **interface**} *real_obj*] [**service** *mapped_dest_svc_obj real_dest_svc_obj*] [**dns**] [**inactive**] [**description** *desc*] **Example:** `hostname(config)# nat (inside,outside) source dynamic MyInsNet interface destination static Server1 Server1 description Interface PAT for inside addresses when going to server 1`	Configures **dynamic PAT (hide)**. 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. • Section and Line—(Optional) By default, the NAT rule is added to the end of section 1 of the NAT table (see the "NAT Rule Order" section on page 27-19). If you want to add the rule into section 3 instead (after the network object NAT rules), then use the **after-auto** keyword. You can insert a rule anywhere in the applicable section using the *line* argument. • Source addresses: – Real—Specify a network object, group, or the **any** keyword (see Step 1). Use the **any** keyword if you want to translate all traffic from the real interface to the mapped interface. – Mapped—Configure one of the following: - Network object—Specify a network object that contains a host address (see Step 2). - **pat-pool**—Specify the **pat-pool** keyword and a network object or group that contains multiple addresses (see Step 2). - **interface**—(Routed mode only) Specify the **interface** keyword alone to only use interface PAT. When specified with a PAT pool or network object, the **interface** keyword enables interface PAT fallback. After the PAT IP addresses are used up, then the IP address of the mapped interface is used. For this option, you must configure a specific interface for the *mapped_ifc*. (continued)

Command	Purpose
	(continued) 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. -- Extended PAT—(8.4(3) and later, not including 8.5(1) or 8.6(1)) 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—(8.4(3) and later, not including 8.5(1) or 8.6(1)) 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. (continued)

Command	Purpose
	(continued) • Destination addresses (Optional): – Mapped—Specify a network object or group, or for static interface NAT with port translation only (routed mode), specify the **interface** keyword (see Step 4). If you specify **interface**, be sure to also configure the **service** keyword. For this option, you must configure a specific interface for the *real_ifc*. See the "Static Interface NAT with Port Translation" section on page 27-5 for more information. – Real—Specify a network object or group (see Step 3). For identity NAT, simply use the same object or group for both the real and mapped addresses. • Destination port—(Optional) Specify the **service** keyword along with the real and mapped service objects (see Step 5). For identity port translation, simply use the same service object for both the real and mapped ports. • DNS—(Optional; for a source-only rule) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). You cannot configure the **dns** keyword if you configure a **destination** address. See the "DNS and NAT" section on page 27-29 for more information. • Inactive—(Optional) To make this rule inactive without having to remove the command, use the **inactive** keyword. To reactivate it, reenter the whole command without the **inactive** keyword. • Description—(Optional) Provide a description up to 200 characters using the **description** keyword.

Configuring Static NAT or Static NAT-with-Port-Translation

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

Detailed Steps

	Command	Purpose

Step 1	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 MyInsNet`
	`hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0`
		Configure the **real source** addresses.
		You can configure either a network object or a network object group. For more information, see the "Configuring Objects" section on page 13-3.

Step 2	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 MyInsNet_mapped`
	`hostname(config-network-object)# subnet 192.168.1.0 255.255.255.0`
		Configure the **mapped source** addresses.
		You can configure either a network object or a network object group. For static NAT, the mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired. For more information, see the "Static NAT" section on page 27-3.
		For static interface NAT with port translation (routed mode only), you can skip this step and specify the **interface** keyword instead of a network object/group for the mapped address. For more information, see the "Static Interface NAT with Port Translation" section on page 27-5.
		See the "Guidelines and Limitations" section on page 31-2 for information about disallowed mapped IP addresses.

	Command	Purpose

Step 3	(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 Server1`
	`hostname(config-network-object)# host 209.165.201.8`
		Configure the **real destination** addresses.
		You can configure either a network object or a network object group.
		Although the main feature of twice NAT is the inclusion of the destination IP address, the destination address is optional. If you do specify the destination address, you can configure static translation for that address or just use identity NAT for it. You might want to configure twice NAT without a destination address to take advantage of some of the other qualities of twice NAT, including the use of network object groups for real addresses, or manually ordering of rules. For more information, see the "Main Differences Between Network Object NAT and Twice NAT" section on page 27-15.

Step 4	(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 Server1_mapped`
	`hostname(config-network-object)# host 10.1.1.67`
		Configure the **mapped destination** addresses.
		The destination translation is always static. For identity NAT, you can skip this step and simply use the same object or group for both the real and mapped addresses.
		If you want to translate the destination address, you can configure either a network object or a network object group. The static mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired. For more information, see the "Static NAT" section on page 27-3.
		For static interface NAT with port translation (routed mode only), you can skip this step and specify the **interface** keyword instead of a network object/group for the mapped address. For more information, see the "Static Interface NAT with Port Translation" section on page 27-5.

	Command	Purpose

Step 5	(Optional)
	object service *obj_name*
	service {**tcp** | **udp**} [**source** *operator port*] [**destination** *operator port*]
	
	Example:
	`hostname(config)# object service REAL_SRC_SVC`
	`hostname(config-service-object)# service tcp source eq 80`
	
	`hostname(config)# object service MAPPED_SRC_SVC`
	`hostname(config-service-object)# service tcp source eq 8080`
		Configure service objects for:
		• **Source *or* destination real port**
		• **Source *or* destination mapped port**
		A service object can contain both a source and destination port; however, you should specify *either* the source *or* the destination port for both service objects. You should only specify *both* the source and destination ports if your application uses a fixed source port (such as some DNS servers); but fixed source ports are rare. NAT only supports TCP or UDP. When translating a port, be sure the protocols in the real and mapped service objects are identical (both TCP or both UDP). For identity NAT, you can use the same service object for both the real and mapped ports. The "not equal" (**neq**) operator is not supported.
		For example, if you want to translate the port for the source host, then configure the source service.

	Command	Purpose
Step 6	**nat** [(*real_ifc*,*mapped_ifc*)] [*line* | {**after-object** [*line*]}] **source static** *real_ob* [*mapped_obj* | **interface**] [**destination static** {*mapped_obj* | **interface**} *real_obj*] [**service** *real_src_mapped_dest_svc_obj mapped_src_real_dest_svc_obj*] [**dns**] [**no-proxy-arp**] [**inactive**] [**description** *desc*] **Example:** `hostname(config)# nat (inside,dmz) source static MyInsNet MyInsNet_mapped destination static Server1 Server1 service REAL_SRC_SVC MAPPED_SRC_SVC`	Configures **static NAT**. 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. • Section and Line—(Optional) By default, the NAT rule is added to the end of section 1 of the NAT table. See the "NAT Rule Order" section on page 27-19 for more information about sections. If you want to add the rule into section 3 instead (after the network object NAT rules), then use the **after-auto** keyword. You can insert a rule anywhere in the applicable section using the *line* argument. • Source addresses: – Real—Specify a network object or group (see Step 1). – Mapped—Specify a different network object or group (see Step 2). For static interface NAT with port translation only, you can specify the **interface** keyword (routed mode only). If you specify **interface**, be sure to also configure the **service** keyword (in this case, the service objects should include only the source port). For this option, you must configure a specific interface for the *mapped_ifc*. See the "Static Interface NAT with Port Translation" section on page 27-5 for more information. • Destination addresses (Optional): – Mapped—Specify a network object or group, or for static interface NAT with port translation only, specify the **interface** keyword (see Step 4). If you specify **interface**, be sure to also configure the **service** keyword (in this case, the service objects should include only the destination port). For this option, you must configure a specific interface for the *real_ifc*. – Real—Specify a network object or group (see Step 3). For identity NAT, simply use the same object or group for both the real and mapped addresses.

Command	Purpose
	(Continued) • Ports—(Optional) Specify the **service** keyword along with the real and mapped service objects (see Step 5). For source port translation, the objects must specify the source service. The order of the service objects in the command for source port translation is **service** *real_obj mapped_obj*. For destination port translation, the objects must specify the destination service. The order of the service objects for destination port translation is **service** *mapped_obj real_obj*. In the rare case where you specify both the source and destination ports in the object, the first service object contains the real source port/mapped destination port; the second service object contains the mapped source port/real destination port. For identity port translation, simply use the same service object for both the real and mapped ports (source and/or destination ports, depending on your configuration). • DNS—(Optional; for a source-only rule) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). You cannot configure the **dns** keyword if you configure a **destination** address. See the "DNS and NAT" section on page 27-29 for more information. • No Proxy ARP—(Optional) Specify **no-proxy-arp** to disable proxy ARP for incoming packets to the mapped IP addresses. See the "Mapped Addresses and Routing" section on page 27-21 for more information. • Inactive—(Optional) To make this rule inactive without having to remove the command, use the **inactive** keyword. To reactivate it, reenter the whole command without the **inactive** keyword. • Description—(Optional) Provide a description up to 200 characters using the **description** keyword.

Examples

The following example shows the use of static interface NAT with port translation. Hosts on the outside access an FTP server on the inside by connecting to the outside interface IP address with destination port 65000 through 65004. The traffic is untranslated to the internal FTP server at 192.168.10.100:6500 through :65004. Note that you specify the source port range in the service object (and not the destination port) because you want to translate the source address and port as identified in the command; the destination port is "any." Because static NAT is bidirectional, "source" and "destination" refers primarily to the command keywords; the actual source and destination address and port in a packet depends on

which host sent the packet. In this example, connections are originated from outside to inside, so the "source" address and port of the FTP server is actually the destination address and port in the originating packet.

```
hostname(config)# object service FTP_PASV_PORT_RANGE
hostname(config-service-object)# service tcp source range 65000 65004

hostname(config)# object network HOST_FTP_SERVER
hostname(config-network-object)# host 192.168.10.100

hostname(config)# nat (inside,outside) source static HOST_FTP_SERVER interface service
FTP_PASV_PORT_RANGE FTP_PASV_PORT_RANGE
```

Configuring Identity NAT

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

Detailed Steps

	Command	Purpose

Step 1	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 MyInsNet`
	`hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0`
		Configure the **real source** addresses.
		
		You can configure either a network object or a network object group. For more information, see the "Configuring Objects" section on page 13-3.
		
		These are the addresses on which you want to perform identity NAT. If you want to perform identity NAT for all addresses, you can skip this step and instead use the keywords **any any**.

	Command	Purpose

Step 2	(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 Server1`
	`hostname(config-network-object)# host 209.165.201.8`
		Configure the **real destination** addresses.
		
		You can configure either a network object or a network object group.
		
		Although the main feature of twice NAT is the inclusion of the destination IP address, the destination address is optional. If you do specify the destination address, you can configure static translation for that address or just use identity NAT for it. You might want to configure twice NAT without a destination address to take advantage of some of the other qualities of twice NAT, including the use of network object groups for real addresses, or manually ordering of rules. For more information, see the "Main Differences Between Network Object NAT and Twice NAT" section on page 27-15.

Step 3	(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 Server1_mapped`
	`hostname(config-network-object)# host 10.1.1.67`
		Configure the **mapped destination** addresses.
		
		The destination translation is always static. For identity NAT, you can skip this step and simply use the same object or group for both the real and mapped addresses.
		
		If you want to translate the destination address, you can configure either a network object or a network object group. The static mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired. For more information, see the "Static NAT" section on page 27-3.
		
		For static interface NAT with port translation (routed mode only), you can skip this step and specify the **interface** keyword instead of a network object/group for the mapped address. For more information, see the "Static Interface NAT with Port Translation" section on page 27-5.

Command	Purpose

Step 4 (Optional)
object service *obj_name*
service {**tcp** | **udp**} [**source** *operator port*] [**destination** *operator port*]
 	
Example:
`hostname(config)# object service REAL_SRC_SVC`
`hostname(config-service-object)# service tcp source eq 80`
 	
`hostname(config)# object service MAPPED_SRC_SVC`
`hostname(config-service-object)# service tcp source eq 8080`
	Configure service objects for:
	• **Source *or* destination real port**
	• **Source *or* destination mapped port**
	A service object can contain both a source and destination port; however, you should specify *either* the source *or* the destination port for both service objects. You should only specify *both* the source and destination ports if your application uses a fixed source port (such as some DNS servers); but fixed source ports are rare. NAT only supports TCP or UDP. When translating a port, be sure the protocols in the real and mapped service objects are identical (both TCP or both UDP). For identity NAT, you can use the same service object for both the real and mapped ports. The "not equal" (**neq**) operator is not supported.
	For example, if you want to translate the port for the source host, then configure the source service.

	Command	Purpose
Step 5	**nat** [(*real_ifc*,*mapped_ifc*)] [*line* | {**after-object** [*line*]}] **source static** {*nw_obj nw_obj* | **any any**} [**destination static** {*mapped_obj* | **interface**} *real_obj*] [**service** *real_src_mapped_dest_svc_obj mapped_src_real_dest_svc_obj*] [**no-proxy-arp**] [**route-lookup**] [**inactive**] [**description** *desc*] **Example:** `hostname(config)# nat (inside,outside) source static MyInsNet MyInsNet destination static Server1 Server1`	Configures **identity NAT**. 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. • Section and Line—(Optional) By default, the NAT rule is added to the end of section 1 of the NAT table. See the "NAT Rule Order" section on page 27-19 for more information about sections. If you want to add the rule into section 3 instead (after the network object NAT rules), then use the **after-auto** keyword. You can insert a rule anywhere in the applicable section using the *line* argument. • Source addresses—Specify a network object, group, or the **any** keyword for both the real and mapped addresses (see Step 1). • Destination addresses (Optional): – Mapped—Specify a network object or group, or for static interface NAT with port translation only, specify the **interface** keyword (routed mode only) (see Step 3). If you specify **interface**, be sure to also configure the **service** keyword (in this case, the service objects should include only the destination port). For this option, you must configure a specific interface for the *real_ifc*. See the "Static Interface NAT with Port Translation" section on page 27-5 for more information. – Real—Specify a network object or group (see Step 2). For identity NAT, simply use the same object or group for both the real and mapped addresses. • Port—(Optional) Specify the **service** keyword along with the real and mapped service objects (see Step 4). For source port translation, the objects must specify the source service. The order of the service objects in the command for source port translation is **service** *real_obj mapped_obj*. For destination port translation, the objects must specify the destination service. The order of the service objects for destination port translation is **service** *mapped_obj real_obj*. In the rare case where you specify both the source and destination ports in the object, the first service object contains the real source port/mapped destination port; the second service object contains the mapped source port/real destination port. For identity port translation, simply use the same service object for both the real and mapped ports (source and/or destination ports, depending on your configuration).

Command	Purpose
	(Continued) • No Proxy ARP—(Optional) Specify **no-proxy-arp** to disable proxy ARP for incoming packets to the mapped IP addresses. See the "Mapped Addresses and Routing" section on page 27-21 for more information. • Route lookup—(Optional; 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 the "Determining the Egress Interface" section on page 27-23 for more information. • Inactive—(Optional) To make this rule inactive without having to remove the command, use the **inactive** keyword. To reactivate it, reenter the whole command without the **inactive** keyword. • Description—(Optional) Provide a description up to 200 characters using the **description** keyword.

Monitoring Twice NAT

To monitor twice 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.
`show xlate`	Shows current NAT session information.

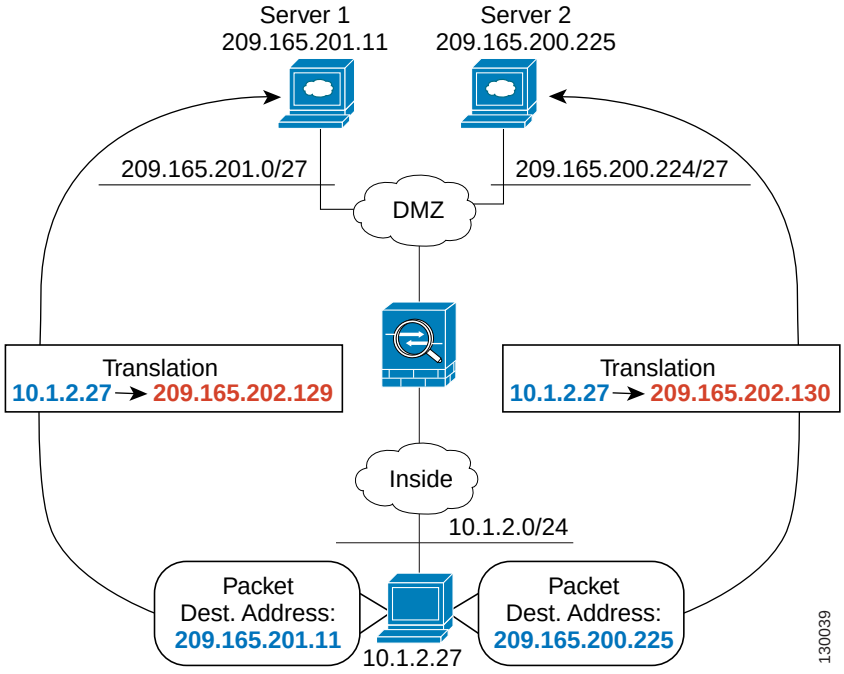
Configuration Examples for Twice NAT

This section includes the following configuration examples:

Different Translation Depending on the Destination (Dynamic PAT)

Figure 31-1 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:*port*. When the host accesses the server at 209.165.200.225, the real address is translated to 209.165.202.130:*port*.

Figure 31-1 Twice NAT with Different Destination Addresses



Step 1 Add a network object for the inside network:

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

Step 2 Add a network object for the DMZ network 1:

```
hostname(config)# object network DMZnetwork1
hostname(config-network-object)# subnet 209.165.201.0 255.255.255.224
```

Step 3 Add a network object for the PAT address:

```
hostname(config)# object network PATaddress1
hostname(config-network-object)# host 209.165.202.129
```

Step 4 Configure the first twice NAT rule:

```
hostname(config)# nat (inside,dmz) source dynamic myInsideNetwork PATaddress1 destination
static DMZnetwork1 DMZnetwork1
```

Because you do not want to translate the destination address, you need to configure identity NAT for it by specifying the same address for the real and mapped destination addresses.

By default, the NAT rule is added to the end of section 1 of the NAT table, See the "Configuring Dynamic PAT (Hide)" section on page 31-8 for more information about specifying the section and line number for the NAT rule.

Step 5 Add a network object for the DMZ network 2:

```
hostname(config)# object network DMZnetwork2
hostname(config-network-object)# subnet 209.165.200.224 255.255.255.224
```

Step 6 Add a network object for the PAT address:

```
hostname(config)# object network PATaddress2
```

```
hostname(config-network-object)# host 209.165.202.130
```

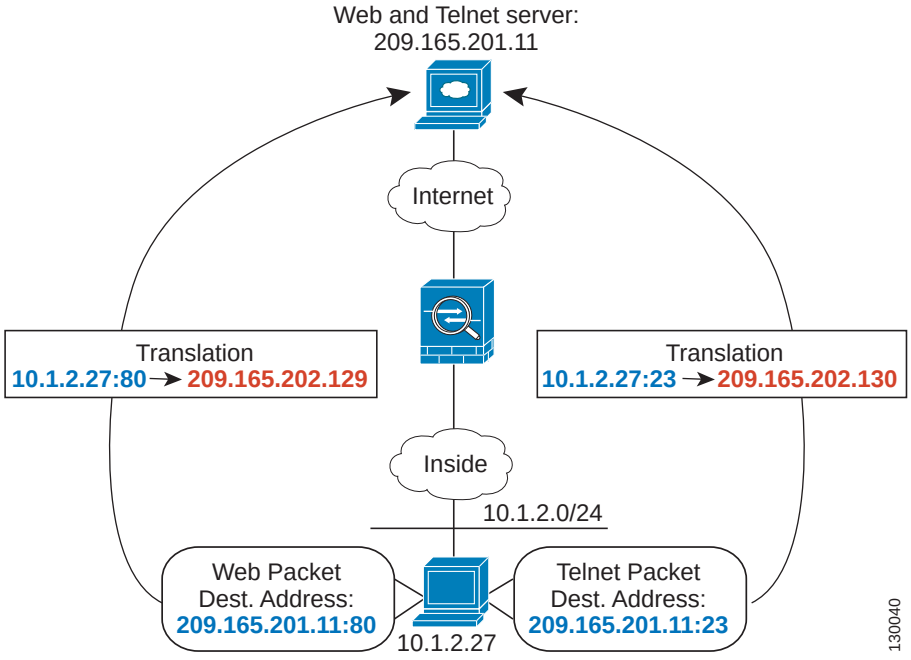
Step 7 Configure the second twice NAT rule:

```
hostname(config)# nat (inside,dmz) source dynamic myInsideNetwork PATaddress2 destination
static DMZnetwork2 DMZnetwork2
```

Different Translation Depending on the Destination Address and Port (Dynamic PAT)

Figure 31-2 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 Telnet services, the real address is translated to 209.165.202.129:*port*. When the host accesses the same server for web services, the real address is translated to 209.165.202.130:*port*.

Figure 31-2 Twice NAT with Different Destination Ports



Step 1 Add a network object for the inside network:

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

Step 2 Add a network object for the Telnet/Web server:

```
hostname(config)# object network TelnetWebServer
hostname(config-network-object)# host 209.165.201.11
```

Step 3 Add a network object for the PAT address when using Telnet:

```
hostname(config)# object network PATaddress1
```

```
hostname(config-network-object)# host 209.165.202.129
```

Step 4 Add a service object for Telnet:

```
hostname(config)# object service TelnetObj
hostname(config-network-object)# service tcp destination eq telnet
```

Step 5 Configure the first twice NAT rule:

```
hostname(config)# nat (inside,outside) source dynamic myInsideNetwork PATaddress1
destination static TelnetWebServer TelnetWebServer service TelnetObj TelnetObj
```

Because you do not want to translate the destination address or port, you need to configure identity NAT for them by specifying the same address for the real and mapped destination addresses, and the same port for the real and mapped service.

By default, the NAT rule is added to the end of section 1 of the NAT table, See the "Configuring Dynamic PAT (Hide)" section on page 31-8 for more information about specifying the section and line number for the NAT rule.

Step 6 Add a network object for the PAT address when using HTTP:

```
hostname(config)# object network PATaddress2
hostname(config-network-object)# host 209.165.202.130
```

Step 7 Add a service object for HTTP:

```
hostname(config)# object service HTTPObj
hostname(config-network-object)# service tcp destination eq http
```

Step 8 Configure the second twice NAT rule:

```
hostname(config)# nat (inside,outside) source dynamic myInsideNetwork PATaddress2
destination static TelnetWebServer TelnetWebServer service HTTPObj HTTPObj
```

Feature History for Twice NAT

Table 31-1 lists each feature change and the platform release in which it was implemented.

Table 31-1 Feature History for Twice NAT

Feature Name	Platform Releases	Feature Information
Twice NAT	8.3(1)	Twice NAT lets you identify both the source and destination address in a single rule. We modified or introduced the following commands: **nat**, **show nat**, **show xlate**, **show nat pool**.
Identity NAT configurable proxy ARP and route lookup	8.4(2)	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. For pre-8.3 configurations, the migration of NAT exempt rules (the **nat 0 access-list** command) to 8.4(2) and later now includes the following keywords to disable proxy ARP and to use a route lookup: **no-proxy-arp** and **route-lookup**. The **unidirectional** keyword that was used for migrating to 8.3(2) and 8.4(1) is no longer used for migration. 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. The **unidirectional** keyword is removed. We modified the following commands: **nat source static** [**no-proxy-arp**] [**route-lookup**].
PAT pool and round robin address assignment	8.4(2)	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 modified the following commands: **nat source dynamic** [**pat-pool** *mapped_object* [**round-robin**]].

Table 31-1 Feature History for Twice NAT (continued)

Feature Name	Platform Releases	Feature Information
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).*
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 modified the following commands: **nat source dynamic** [**pat-pool** *mapped_object* [**flat** [**include-reserve**]]]. *This feature is not available in 8.5(1) or 8.6(1).*

Table 31-1 Feature History for Twice NAT (continued)

Feature Name	Platform Releases	Feature Information
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 modified the following commands: **nat source dynamic** [**pat-pool** *mapped_object* [**extended**]]. *This feature is not available in 8.5(1) or 8.6(1).*
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).