CHAPTER 29

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:

- Information About Twice NAT, page 29-1
- Licensing Requirements for Twice NAT, page 29-2
- Prerequisites for Twice NAT, page 29-2
- Guidelines and Limitations, page 29-2
- Configuring Twice NAT, page 29-3
- Monitoring Twice NAT, page 29-20
- Configuration Examples for Twice NAT, page 29-20
- Feature History for Twice NAT, page 29-23

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 11-1.
- 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 11-4.

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

Configuring Twice NAT

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

- Configuring Dynamic NAT, page 29-3
- Configuring Dynamic PAT (Hide), page 29-8
- Configuring Static NAT or Static NAT with Port Translation, page 29-12
- Configuring Identity NAT, page 29-17

Configuring Dynamic NAT

This section describes how to configure a dynamic NAT rule using twice NAT. For more information about dynamic NAT, 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 11-3.
		If you want to translate all traffic, you can specify the **any** keyword instead of creating an object or group; skip this step.

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.
		Note The mapped object or group cannot contain a subnet.
		You can share this mapped IP address across different dynamic NAT rules, if desired.
		See the "Guidelines and Limitations" section on page 29-2 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.
		You can configure either a network object or a network object group.
		The destination translation is always static. For identity NAT, simply use the same object or group for both the real and mapped addresses, and skip this step.
		If you want to translate the destination address, then 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, you can specify the **interface** keyword instead of a network object/group for the mapped address; you can skip this step. For more information, see the "Static Interface NAT with Port Translation" section on page 27-5.
		See the "Guidelines and Limitations" section on page 29-2 for information about disallowed mapped IP addresses.

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 TCP or UDP port
	• Destination mapped TCP or UDP 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`	Configures dynamic NAT. See the following guidelines: • Interfaces—If you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword **any** for one or both of the interfaces. • Line—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 share this mapped object across different dynamic NAT rules, if desired. If you specify a mapped object or group followed by the **interface** keyword (routed mode only), then the IP address of the mapped interface is only used if all other mapped addresses are already allocated. For this option, you must configure a specific interface for the *mapped_ifc*. • Destination addresses: – 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—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. • (For a source-only rule) DNS—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 27-21 for more information. • Inactive—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—Provide a description up to 200 characters using the **description** keyword.

Configuring Dynamic PAT (Hide)

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

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 11-3.
		If you want to translate all traffic, you can specify the **any** keyword instead of creating an object or group; skip this step.

Step 2	**object network** *obj_name*
	host *ip_address*
	Example:
	`hostname(config)# object network PAT_IP`
	`hostname(config-network-object)# host 209.165.201.10`
		Configure the mapped source address.
		You can configure a network object.
		For dynamic PAT, configure a group of addresses to be mapped to a single address. You can either translate the real addresses to a single mapped address of your choosing, or you can translate them to the mapped interface address. If you want to use the interface address, do not configure a network object for the mapped address; instead use the **interface** keyword.
		Note You can share this mapped object across different dynamic PAT rules, if desired.
		See the "Guidelines and Limitations" section on page 29-2 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.
		You can configure either a network object or a network object group.
		The destination translation is always static. For identity NAT, simply use the same object or group for both the real and mapped addresses, and skip this step.
		If you want to translate the destination address, then 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, you can specify the **interface** keyword instead of a network object/group for the mapped address; you can skip this step. For more information, see the "Static Interface NAT with Port Translation" section on page 27-5.
		See the "Guidelines and Limitations" section on page 29-2 for information about disallowed mapped IP addresses.

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 TCP or UDP port
	• Destination mapped TCP or UDP 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**} [**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—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. • Line—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 between the from the real interface to the mapped interface. – Mapped—Specify a network object that contains a host address (see Step 2). You can share this mapped object across different dynamic PAT rules, if desired. To use the mapped interface IP address, specify the **interface** keyword (routed mode only). For this option, you must configure a specific interface for the *mapped_ifc*. • Destination addresses: – 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—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. • (For a source-only rule) DNS—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 27-21 for more information. • Inactive—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—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 11-3.

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, you can specify the **interface** keyword instead of a network object/group for the mapped address; you can skip this step. For more information, see the "Static Interface NAT with Port Translation" section on page 27-5.
		
		See the "Guidelines and Limitations" section on page 29-2 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. For more information, see the "Configuring Objects" section on page 11-3.
		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.
		You can configure either a network object or a network object group.
		The destination translation is always static. For identity NAT, simply use the same object or group for both the real and mapped addresses, and skip this step.
		If you want to translate the destination address, then 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.S
		For static interface NAT with port translation, you can specify the **interface** keyword instead of a network object/group for the mapped address; you can skip this step. For more information, see the "Static Interface NAT with Port Translation" section on page 27-5.
		See the "Guidelines and Limitations" section on page 29-2 for information about disallowed mapped IP addresses.

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 TCP or UDP port
	• Source *or* destination mapped TCP or UDP port
	
	A service object can contain both a source and destination port. 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_obj* [*mapped_obj* | **interface**] [**destination static** {*mapped_obj* | **interface**} *real_obj*] [**service** *real_src_mapped_dest_svc_obj mapped_src_real_dest_svc_obj*] [**dns**] [**unidirectional**] [**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—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. • Line—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: – 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) • Port—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 is **service** *real mapped*. For destination port translation, the objects must specify the destination service. The order of the service objects is **service** *mapped real*. 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). • (For a source-only rule) DNS—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 27-21 for more information. • Inactive—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—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-obvject)# service tcp source range 65000 65004

hostname(config)# object network HOST_FTP_SERVER
hostname(config-network-obvject)# 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. You configure identity NAT using a static NAT rule where you map an address to itself. 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 11-3.
		These are the addresses on which you want to perform identity NAT. If you want to perform identity NAT for all addresses, you can instead use the keywords **any any**; skip this step.

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.

	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_mapped`
	`hostname(config-network-object)# host 10.1.1.67`
		Configure the mapped destination addresses.
		You can configure either a network object or a network object group.
		The destination translation is always static. For identity NAT, simply use the same object or group for both the real and mapped addresses, and skip this step.
		If you want to translate the destination address, then 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, you can 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 29-2 for information about disallowed mapped IP addresses.

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 TCP or UDP port
		• Source *or* destination mapped TCP or UDP port
		A service object can contain both a source and destination port. 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*] [**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—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. • Line—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: – 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—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 is **service** *real mapped*. For destination port translation, the objects must specify the destination service. The order of the service objects is **service** *mapped real*. 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). • Inactive—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—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

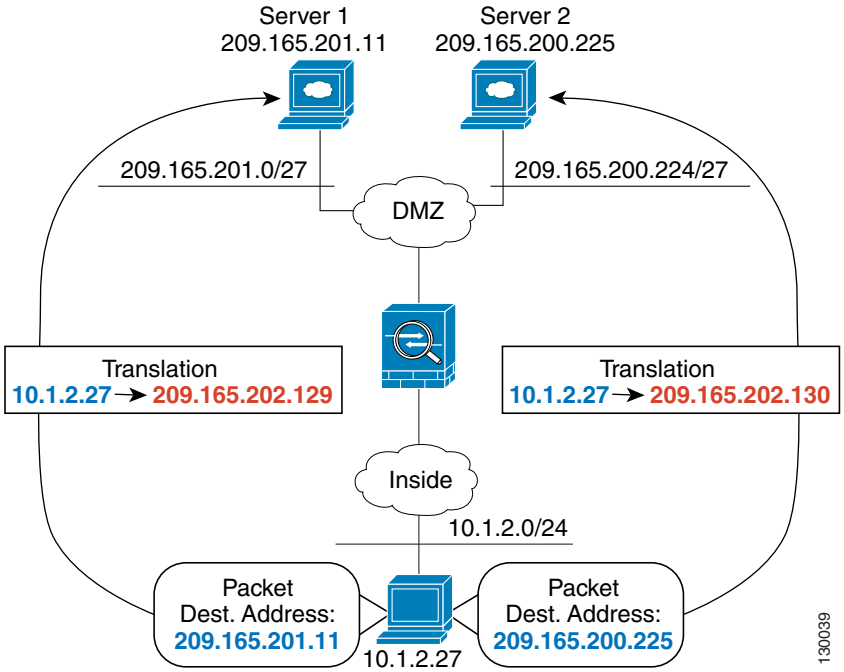
This section includes the following configuration examples:

- Different Translation Depending on the Destination (Dynamic PAT), page 29-20
- Different Translation Depending on the Destination Address and Port (Dynamic PAT), page 29-22

Different Translation Depending on the Destination (Dynamic PAT)

Figure 29-1 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 29-1 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 29-8 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
```

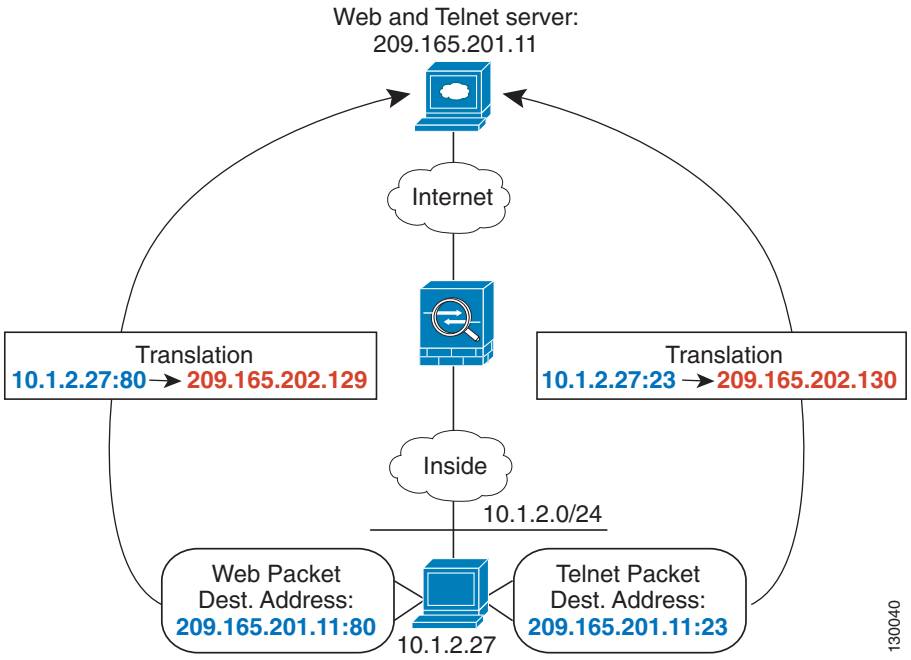
Step 7 Configure the second twice NAT rule:

```
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 29-2 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 29-2 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 29-8 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 29-1 lists each feature change and the platform release in which it was implemented.

Table 29-1 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. The following commands were modified or introduced: **nat**, **show nat**, **show xlate**, **show nat pool**.