IP Addressing: NAT Configuration Guide, Cisco IOS Release 12.2SX

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Configuring NAT for IP Address Conservation

This module describes how to configure Network Address Translation (NAT) for IP address conservation and configure inside and outside source addresses. This module also provides information about the benefits of configuring NAT for IP address conservation.

NAT enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT operates on a router, usually connecting two networks, and translates the private (not globally unique) addresses in the internal network into legal addresses before packets are forwarded onto another network. NAT can be configured to advertise only one address for the entire network to the outside world. This ability provides additional security by effectively hiding the entire internal network behind that one address.

NAT is also used at the enterprise edge to allow internal users access to the Internet and to allow Internet access to internal devices such as mail servers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring NAT for IP Address Conservation

- Access Lists, page 2

- NAT Requirements Objectives and Interfaces, page 2

Access Lists

All access lists required for use with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, refer to the *IP Access List Sequence Numbering* document.

Note If you specify an access list with a NAT command, NAT will not support the commonly used **permit ip any any** command in the access list.

NAT Requirements Objectives and Interfaces

Before configuring NAT in your network, it is important to understand on which interfaces NAT will be configured and for what purposes. The requirements listed below would help you to decide on how to configure and use NAT:

1. Define the NAT inside and outside interfaces if:
    - Users exist off multiple interfaces.
    - There are multiple interfaces connecting to the Internet.
2. Define what you need NAT to accomplish:
    - Allow internal users to access the Internet.
    - Allow the Internet to access internal devices such as a mail server.
    - Redirect TCP traffic to another TCP port or address.
    - To use NAT during a network transition.
    - Allow overlapping networks to communicate.
    - Allow networks with different address schemes to communicate.
    - Allow the use of an application level gateway.

Restrictions for Configuring NAT for IP Address Conservation

- NAT is not practical if large numbers of hosts in the stub domain communicate outside of the domain.
- Some applications use embedded IP addresses in such a way that it is impractical for a NAT device to translate them. These applications may not work transparently or not work at all through a NAT device.
- NAT hides the identity of hosts, which may be an advantage or a disadvantage depending on the desired result.
- A router configured with NAT must not advertise the local networks to the outside. However, routing information that NAT receives from the outside can be advertised in the stub domain as usual.
- If you specify an access list with a NAT command, NAT will not support the commonly used **permit ip any any** command in the access list.

Information About Configuring NAT for IP Address Conservation

Benefits of Configuring NAT for IP Address Conservation

NAT allows organizations to resolve the problem of IP address depletion when they have existing networks and need to access the Internet. Sites that do not yet possess network information center (NIC)-registered IP addresses must acquire them, and if more than 254 clients are present or planned, the scarcity of Class B addresses becomes a serious issue. Cisco IOS NAT addresses these issues by mapping thousands of hidden internal addresses to a range of easy-to-get Class C addresses.

Sites that already have registered IP addresses for clients on an internal network may want to hide those addresses from the Internet so that hackers cannot directly attack the clients. With client addresses are hidden, a degree of security is established. Cisco IOS NAT gives LAN administrators complete freedom to expand Class A addressing, which is drawn from the reserve pool of the Internet Assigned Numbers Authority (RFC 1597). This expansion occurs within the organization without the concern for addressing changes at the LAN or Internet interface.

Cisco IOS software can selectively or dynamically perform NAT. This flexibility allows the network administrator to use a mix of RFC 1597 and RFC 1918 addresses or registered addresses. NAT is designed for use on a variety of routers for IP address simplification and conservation. In addition, Cisco IOS NAT allows the selection of internal hosts that are available for NAT.

A significant advantage of NAT is that it can be configured without requiring any changes to hosts or routers other than those few routers on which NAT will be configured.

Purpose of NAT

Two key problems facing the Internet are the depletion of IP address space and the scaling in routing. NAT is a feature that allows the IP network of an organization to appear, from the outside, to use a different IP address space than what it is actually using. Thus, NAT allows an organization with nonglobally routable addresses to connect to the Internet by translating those addresses into a globally routable address space. NAT also allows a graceful renumbering strategy for organizations that are changing service providers or voluntarily renumbering into classless interdomain routing (CIDR) blocks. NAT is described in RFC 1631.

Beginning with Cisco IOS Release 12.1(5)T, NAT supports all H.225 and H.245 message types, including FastConnect and Alerting as part of the H.323 version 2 specification. Any product that makes use of these message types will be able to pass through a Cisco IOS NAT configuration without any static configuration. Full support for NetMeeting Directory (Internet Locator Service) is also provided through Cisco IOS NAT.

How NAT Works

A router configured with NAT will have at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit router between a stub domain and the backbone. When a packet is leaving the domain, NAT translates the locally significant source address into a globally unique address. When a packet is entering the domain, NAT translates the globally unique destination address into a local address. If more than one exit point exists, each NAT must have the same translation table. If the software cannot allocate an address because it has run out of addresses, it drops the packet and sends an Internet Control Message Protocol (ICMP) host unreachable packet.

Uses of NAT

NAT can be used in the following scenarios:

- When you want to connect to the Internet, but not all of your hosts have globally unique IP addresses. NAT enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT is configured on the router at the border of a stub domain (referred to as the *inside network*) and a public network such as the Internet (referred to as the *outside network*). NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network. As a solution to the connectivity problem, NAT is practical only when relatively few hosts in a stub domain communicate outside of the domain at the same time. Only a small subset of the IP addresses in the domain must be translated into globally unique IP addresses when outside communication is necessary, and these addresses can be reused when no longer in use.
- When you must change your internal addresses. Instead of changing them, which can be a considerable amount of work, you can translate them by using NAT.
- When you want to do basic load sharing of TCP traffic. You can map a single global IP address to many local IP addresses by using the TCP load distribution feature.

NAT Inside and Outside Addresses

The term inside in a NAT context refers to networks owned by an organization that must be translated. When NAT is configured, hosts within this network will have addresses in one space (knows as the local address space) that will appear to those outside the network as being in another space (known as the global address space).

Similarly, *outside* refers to those networks to which the stub network connects, and which are generally not under the control of the organization. Hosts in outside networks can also be subject to translation, and thus have local and global addresses.

NAT uses the following definitions:

- Inside local address--The IP address that is assigned to a host on the inside network. The address is probably not a legitimate IP address assigned by the NIC or service provider.
- Inside global address--A legitimate IP address (assigned by the NIC or service provider) that represents one or more inside local IP addresses to the outside world.
- Outside local address--The IP address of an outside host as it appears to the inside network. Not necessarily a legitimate address, it is allocated from the address space routable on the inside.

- Outside global address--The IP address assigned to a host on the outside network by the owner of the host. The address is allocated from a globally routable address or network space.

- Inside Source Address Translation, page 5
- Inside Global Addresses Overloading, page 6

Inside Source Address Translation

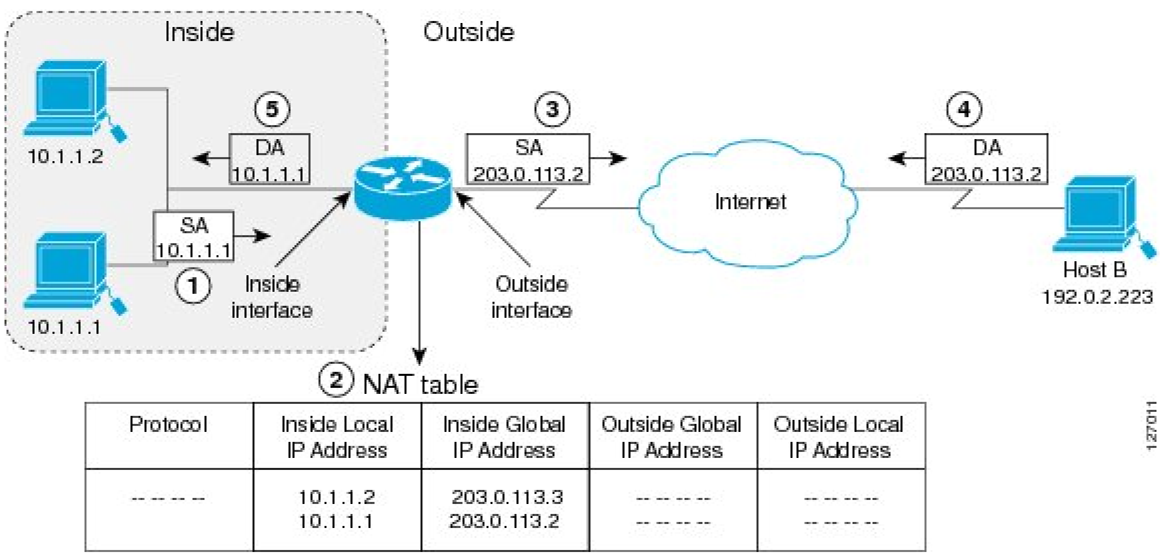
You can translate your own IP addresses into globally unique IP addresses when communicating outside of your network. You can configure static or dynamic inside source translation as follows:

- *Static translation* establishes a one-to-one mapping between your inside local address and an inside global address. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.
- *Dynamic translation* establishes a mapping between an inside local address and a pool of global addresses.

In Cisco IOS Release 15.1(3)T and later releases, when you configure the **traceroute** command, NAT returns the same inside global IP address for all inside local IP addresses.

The figure below illustrates a router that is translating a source address inside a network to a source address outside the network.

Figure 1



The following process describes inside source address translation, as shown in the figure above:

1. The user at host 10.1.1.1 opens a connection to host B.
2. The first packet that the router receives from host 10.1.1.1 causes the router to check its NAT table:
   - If a static translation entry was configured, the router goes to Step 3.
   - If no translation entry exists, the router determines that the source address (SA) 10.1.1.1 must be translated dynamically, selects a legal, global address from the dynamic address pool, and creates a translation entry. This type of entry is called a *simple entry*.
3. The router replaces the inside local source address of host 10.1.1.1 with the global address of the translation entry and forwards the packet.

4 Host B receives the packet and responds to host 10.1.1.1 by using the inside global IP destination address (DA) 203.0.113.2.

5 When the router receives the packet with the inside global IP address, it performs a NAT table lookup by using the inside global address as a key. It then translates the address to the inside local address of host 10.1.1.1 and forwards the packet to host 10.1.1.1.

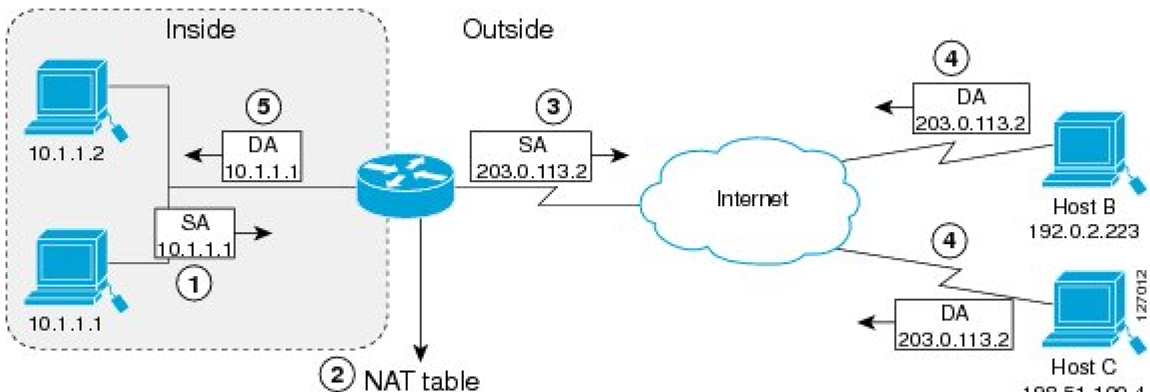
Host 10.1.1.1 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

Inside Global Addresses Overloading

You can conserve addresses in the inside global address pool by allowing the router to use one global address for many local addresses. When this overloading is configured, the router maintains enough information from higher-level protocols (for example, TCP or UDP port numbers) to translate the global address back to the correct local address. When multiple local addresses map to one global address, the TCP or UDP port numbers of each inside host distinguish between the local addresses.

The figure below illustrates NAT operation when one inside global address represents multiple inside local addresses. The TCP port numbers act as differentiators.

Figure 2



Protocol	Inside Local IP address:port	Inside Global IP address:port	Outside Global IP address:port	Outside Local IP address
TCP TCP	10.1.1.2:1723 10.1.1.1:1024	203.0.113.2:1723 203.0.113.2:1024	198.51.100.4:23 192.0.2.223:23	198.51.100.4:23 192.0.2.223:23

The router performs the following process in overloading inside global addresses, as shown in the figure above. Both host B and host C believe that they are communicating with a single host at address 2.2.2.2. They are actually communicating with different hosts; the port number is the differentiator. In fact, many inside hosts could share the inside global IP address by using many port numbers.

1 The user at host 10.1.1.1 opens a connection to host B.

2 The first packet that the router receives from host 10.1.1.1 causes the router to check its NAT table:

- If no translation entry exists, the router determines that the address 10.1.1.1 must be translated, and sets up a translation of the inside local address 10.1.1.1 to a legal global address.
- If overloading is enabled, and another translation is active, the router reuses the global address from that translation and saves enough information to be able to translate back. This type of entry is called an *extended entry*.

3. The router replaces the inside local source address 10.1.1.1 with the selected global address and forwards the packet.
4. Host B receives the packet and responds to host 10.1.1.1 by using the inside global IP address 203.0.113.2.
5. When the router receives the packet with the inside global IP address, it performs a NAT table lookup, using the protocol, the inside global address and port, and the outside address and port as keys; translates the address to the inside local address 10.1.1.1; and forwards the packet to host 10.1.1.1.

Host 10.1.1.1 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

Types of NAT

NAT operates on a router--generally connecting only two networks--and translates the private (inside local) addresses within the internal network into public (inside global) addresses before any packets are forwarded to another network. This functionality gives you the option to configure NAT so that it will advertise only a single address for your entire network to the outside world. Doing this effectively hides the internal network from the world, giving you some additional security.

NAT types include:

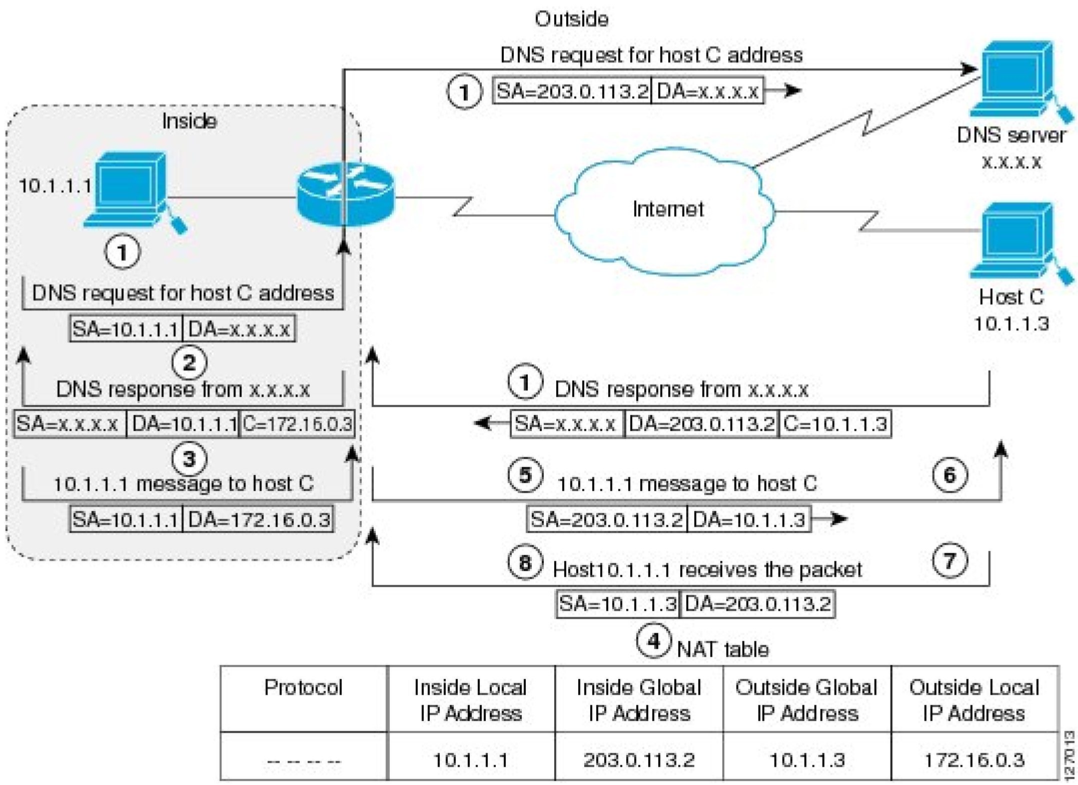
- Static address translation (static NAT)--allows one-to-one mapping between local and global addresses.
- Dynamic address translation (dynamic NAT)--maps unregistered IP addresses to registered IP addresses from a pool of registered IP addresses.
- Overloading--a form of dynamic NAT--maps multiple unregistered IP addresses to a single registered IP address (many to one) using different ports. This method is also known as Port Address Translation (PAT). By using PAT (NAT overload), thousands of users can be connected to the Internet using only one real global IP address.

Address Translation of Overlapping Networks

NAT is used to translate your IP addresses, if your IP addresses are not legal or officially assigned IP addresses. Perhaps you chose IP addresses that officially belong to another network. When an IP address is used both illegally and legally, it is called *index overlapping*. You can use NAT to translate inside addresses that overlap with outside addresses.

The figure below shows how NAT translates overlapping networks.

Figure 3



The router performs the following tasks when translating overlapping addresses:

1 The user at host 10.1.1.1 opens a connection to host C by name, requesting a name-to-address lookup from a DNS server.
2 The router intercepts the DNS reply and translates the returned address if there is an overlap (that is, the resulting legal address resides illegally in the inside network). To translate the return address, the router creates a simple translation entry mapping the overlapping address 10.1.1.3 to an address from a separately configured, outside local address pool.

The router examines every DNS reply from everywhere, ensuring that the IP address is not in the stub network. If it is, the router translates the address.

1 Host 10.1.1.1 opens a connection to 172.16.0.3.
2 The router sets up a translations mapping of the inside local and global addresses to each other and the outside global and local addresses to each other.
3 The router replaces the SA with the inside global address and replaces the DA with the outside global address.
4 Host C receives the packet and continues the conversation.
5 The router does a lookup, replaces the DA with the inside local address, and replaces the SA with the outside local address.
6 Host 10.1.1.1 receives the packet and the conversation continues, using this translation process.

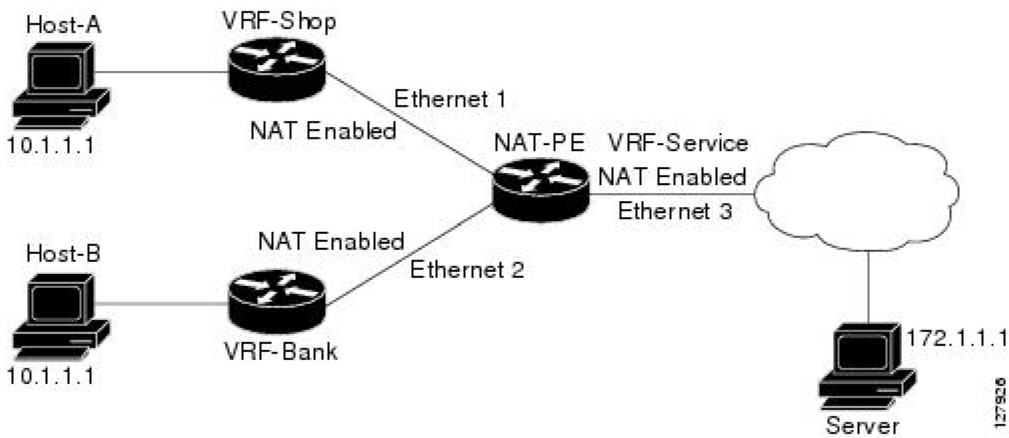
NAT Virtual Interface Design

The NVI feature allows NAT traffic flows on the virtual interface, eliminating the need to specify inside and outside domains. When a domain is specified, the translation rules are applied either before or after the route decisions depending on the traffic flow from inside to outside or outside to inside. The translation rules are applied only after the route decision for an NVI.

When a NAT pool is shared for translating packets from multiple networks connected to a NAT router, an NVI is created and a static route is configured that forwards all packets addressed to the NAT pool to the NVI. The standard interfaces connected to the various networks will be configured to identify that the traffic originating from and received on the interfaces needs to be translated.

The figure below shows a typical NVI configuration.

Figure 4



NAT Virtual Interface has the following benefits:

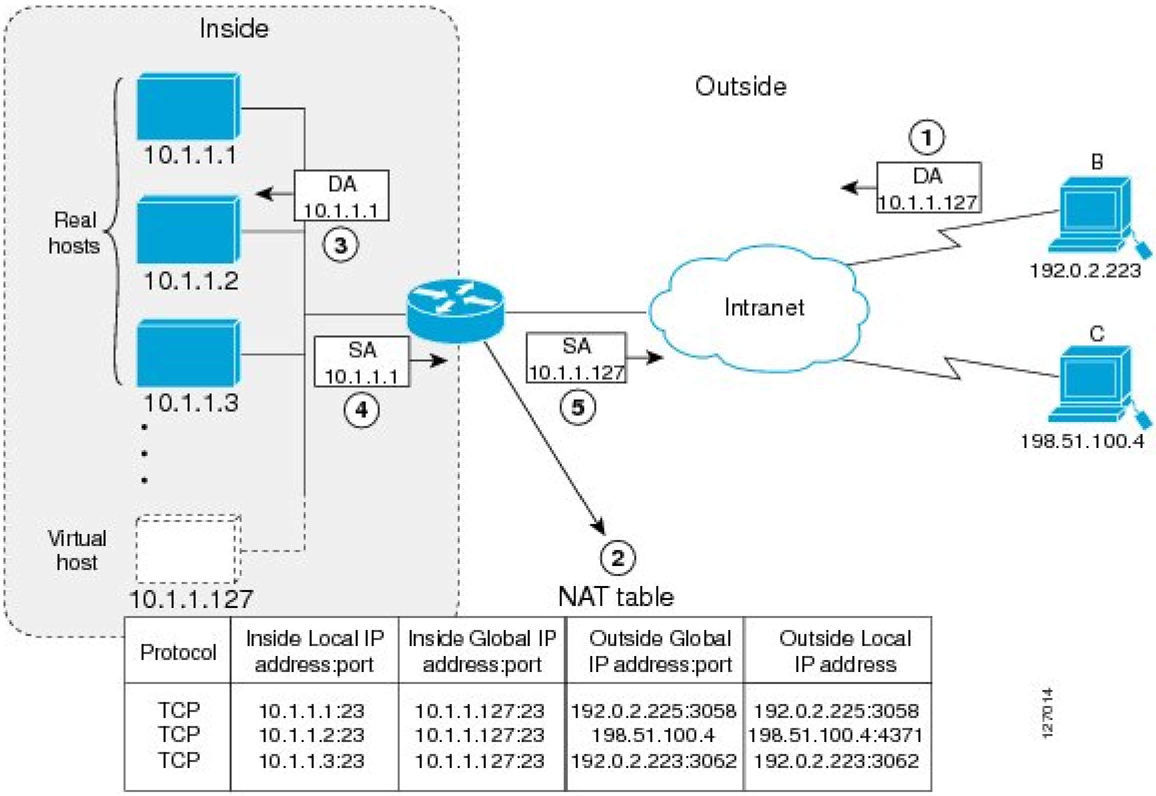
- A NAT table is maintained per interface for better performance and scalability.
- Domain-specific NAT configurations can be eliminated.

- TCP Load Distribution for NAT, page 9

TCP Load Distribution for NAT

Your organization may have multiple hosts that must communicate with a heavily used host. Using NAT, you can establish a virtual host on the inside network that coordinates load sharing among real hosts. DAs that match an access list are replaced with addresses from a rotary pool. Allocation is done on a roundrobin basis, and only when a new connection is opened from the outside to the inside. Non-TCP traffic is passed untranslated (unless other translations are in effect). The figure below illustrates this feature.

Figure 5



The router performs the following process when translating rotary addresses:

1. The user on host B (9.6.7.3) opens a connection to the virtual host at 10.1.1.127.
2. The router receives the connection request and creates a new translation, allocating the next real host (10.1.1.1) for the inside local IP address.
3. The router replaces the destination address with the selected real host address and forwards the packet.
4. Host 10.1.1.1 receives the packet and responds.
5. The router receives the packet and performs a NAT table lookup using the inside local address and port number, and the outside address and port number as the key. The router then translates the source address to the address of the virtual host and forwards the packet.

The next connection request will cause the router to allocate 10.1.1.2 for the inside local address.

Route Map Overview

For NAT, a route map must be processed instead of an access list. A route map allows you to match any combination of access list, next hop IP address, and output interface to determine which pool to use. The ability to use route maps with static translations enables the NAT multihoming capability with static address translations. Multihomed internal networks can host common services such as the Internet and DNS, which are accessed from different outside networks. NAT processes route map-based mappings in lexicographical order. When static NAT and dynamic NAT are configured with route maps that share the same name, static NAT is given precedence over dynamic NAT. In order to ensure the precedence of static NAT over dynamic NAT, you can either configure the route map associated with static NAT and dynamic NAT to share the same name, or configure the static NAT route map name so that it is lexicographically lower than that of the dynamic NAT route map name.

Benefits of Using Route Maps for Address Translation are the following:

- The ability to configure route map statements provides the option of using IPsec with NAT.
- Translation decisions can be made based on the destination IP address when static translation entries are used.

Public Wireless LAN

A public wireless LAN provides users of mobile computing devices with wireless connections to a public network, such as the Internet.

RADIUS

RADIUS is a distributed client/server system that secures networks against unauthorized access. Communication between a network access server (NAS) and a RADIUS server is based on the UDP. Generally, the RADIUS protocol is considered a connectionless service. Issues related to server availability, retransmission, and timeouts are handled by the RADIUS-enabled devices rather than the transmission protocol.

RADIUS is a client/server protocol. The RADIUS client is typically a NAS, and the RADIUS server is usually a daemon process running on a UNIX or Windows NT machine. The client passes user information to designated RADIUS servers and acts on the response that is returned. RADIUS servers receive user connection requests, authenticate the user, and then return the configuration information necessary for the client to deliver service to the user. A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers.

Denial-of-Service Attacks

A denial-of-service (DoS) attack typically involves the misuse of standard protocols or connection processes with the intent to overload and disable a target, such as a router or web server. DoS attacks can come from a malicious user or from a computer infected with a virus or worm. When the attack comes from many different sources at once, such as when a virus or worm has infected many computers, it is known as a distributed denial-of-service (DDoS) attack. Such DDoS attacks can spread rapidly and involve thousands of systems.

Viruses and Worms That Target NAT

Viruses and worms are programs designed to attack computer and networking equipment. Although viruses are typically embedded in discrete applications and run only when executed, worms self-propagate and can quickly spread on their own. Although a specific virus or worm may not expressly target NAT, it might use NAT resources to propagate itself. The Rate Limiting NAT Translation feature can be used to limit the impact of viruses and worms that originate from specific hosts, access control lists, and VRF instances.

How to Configure NAT for IP Address Conservation

The tasks described in this section configure NAT for IP address conservation. At least one of the tasks must be performed. More than one of the tasks may be needed.

Configuring Inside Source Addresses

Inside source address can be configured for static or dynamic translations. Perform one of the following tasks depending on your requirements:

Configuring Static Translation of Inside Source Addresses

Configure static translation of inside source addresses when you want to allow one-to-one mapping between your inside local address and an inside global address. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.

Prior to Cisco IOS Release 15.1(1)T, if the static inside source address matched the inside global address, the output of the **show ip aliases** command displayed only the static inside source address. In Cisco IOS Release 15.1(1)T and later releases, if the static inside source address matches the inside global address, the output of the **show ip aliases** command displays both the addresses. The static inside source address is displayed as an interface address and the inside global address is displayed as a dynamic address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip*
4. **interface** *type number*
5. **ip address** *ip-address mask* [**secondary**]
6. **ip nat inside**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **ip nat outside**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Router(config)# ip nat inside source static 10.10.10.1 172.16.131.1`	Establishes static translation between an inside local address and an inside global address.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for an interface.
Step 6	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies a different interface and enters interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 10	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 11	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Dynamic Translation of Inside Source Addresses

Dynamic translation establishes a mapping between an inside local address and a pool of global addresses. Dynamic translation is useful when multiple users on a private network need to access the Internet. The dynamically configured pool IP address may be used as needed and is released for use by other users when access to the Internet is no longer required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside source list** *access-list -number* **pool** *name*
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Router(config)# ip nat pool net-208 172.16.233.208 172.16.233.223 prefix-length 28`	Defines a pool of global addresses to be allocated as needed.

	Command or Action	Purpose
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 192.168.34.0 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated.

Step 5	**ip nat inside source list** *access-list -number* **pool** *name*
	
	Example:
	
	`Router(config)#`
	`ip nat inside source list 1 pool net-208`
		Establishes dynamic source translation, specifying the access list defined in the prior step.

Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies a different interface and enters interface configuration mode.

	Command or Action	Purpose
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 13	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Allowing Internal Users Access to the Internet

Perform this task to allow your internal users access to the Internet and conserve addresses in the inside global address pool using overloading of global addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside source list** *access-list -number* **pool** *name* **overload**
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Router(config)# ip nat pool net-208 192.168.202.129 192.168.202.158 netmask 255.255.255.224`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 192.168.201.30 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated. • The access list must permit only those addresses that are to be translated. (Remember that there is an implicit "deny all" at the end of each access list.) Use of an access list that is too permissive can lead to unpredictable results.
Step 5	**ip nat inside source list** *access-list -number* **pool** *name* **overload** **Example:** `Router(config)# ip nat inside source list 1 pool net-208 overload`	Establishes dynamic source translation with overloading, specifying the access list defined in the prior step.
Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.201.1 255.255.255.240`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies a different interface and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.201.29 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 13	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Address Translation Timeouts

This section describes how to change the default translation timeout when overloading is configured and not configured. You can use the configuration that is applicable to your specific NAT configuration.

- Changing the Translation Timeout, page 20
- Changing the Timeouts When Overloading Is Configured, page 20

Changing the Translation Timeout

By default, dynamic address translations time out after some period of nonuse. You can change the default values on timeouts, if necessary. When overloading is not configured, simple translation entries time out after 24 hours. Configure the **ip nat translation timeout** *seconds* commands to change the timeout value for dynamic address translations that do not use overloading.

Changing the Timeouts When Overloading Is Configured

If you have configured overloading, you have more control over translation entry timeouts, because each entry contains more context about the traffic using it. To change timeouts on extended entries, use the following commands as needed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat translation udp-timeout** *seconds*
4. **ip nat translation dns-timeout** *seconds*
5. **ip nat translation tcp-timeout** *seconds*
6. **ip nat translation finrst-timeout** *seconds*
7. **ip nat translation icmp-timeout** *seconds*
8. **ip nat translation syn-timeout** *seconds*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat translation udp-timeout** *seconds* **Example:** `Router(config)# ip nat translation udp-timeout 300`	(Optional) Changes the UDP timeout value.
Step 4	**ip nat translation dns-timeout** *seconds* **Example:** `Router(config)#` **ip nat translation dns-timeout** *45*	(Optional) Changes the Domain Name System (DNS) timeout value.
Step 5	**ip nat translation tcp-timeout** *seconds* **Example:** `Router(config)# i`**p nat translation tcp-timeout** *2500*	(Optional) Changes the TCP timeout value.
Step 6	**ip nat translation finrst-timeout** *seconds* **Example:** `Router(config)#` **ip nat translation finrst-timeout** *45*	(Optional) Changes the finish and reset timeout value.
Step 7	**ip nat translation icmp-timeout** *seconds* **Example:** `Router(config)# ip` **nat translation icmp-timeout** *45*	(Optional) Changes the ICMP timeout value.
Step 8	**ip nat translation syn-timeout** *seconds* **Example:** `Router(config)# ip nat translation syn-timeout 45`	(Optional) Changes the synchronous (SYN) timeout value.
Step 9	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Allowing Overlapping Networks to Communicate Using NAT

The tasks in this section are grouped because they perform the same action but are executed differently depending on the type of translation that is implemented--static or dynamic:

Perform the task that applies to the translation type that is implemented.

- Configuring Static Translation of Overlapping Networks, page 22
- What to Do Next, page 24
- Configuring Dynamic Translation of Overlapping Networks, page 24

Configuring Static Translation of Overlapping Networks

Configure static translation of overlapping networks if your IP addresses in the stub network are legitimate IP addresses belonging to another network and you want to communicate with those hosts or routers using static translation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip*
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **ip nat inside**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **ip nat outside**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Router(config)# ip nat inside source static 192.168.121.33 2.2.2.1`	Establishes static translation between an inside local address and an inside global address.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 6	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies a different interface and enters interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.

	Command or Action	Purpose
Step 10	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 11	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

What to Do Next

When you have completed the required configuration, go to the “Monitoring and Maintaining NAT” module.

Configuring Dynamic Translation of Overlapping Networks

Configure dynamic translation of overlapping networks if your IP addresses in the stub network are legitimate IP addresses belonging to another network and you want to communicate with those hosts or routers using dynamic translation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat outside source list** *access-list -number* **pool** *name*
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Router(config)# ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 10.114.11.0 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated. • The access list must permit only those addresses that are to be translated. (Remember that there is an implicit "deny all" at the end of each access list.) Use of an access list that is too permissive can lead to unpredictable results.
Step 5	**ip nat outside source list** *access-list -number* **pool** *name* **Example:** `Router(config)# ip nat outside source list 1 pool net-10`	Establishes dynamic outside source translation, specifying the access list defined in Step 4.
Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies a different interface and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 13	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the NAT Virtual Interface

The NAT Virtual Interface (NVI) feature removes the requirement to configure an interface as either NAT inside or NAT outside. An interface can be configured to use or not use NAT.

This section contains the following procedure:

Restrictions for NAT Virtual Interface

- Route maps are not supported.
- NVI is not supported in a NAT on-a-stick scenario. The term NAT on-a-stick implies the use of a single physical interface of a router for translation. NVI is designed for traffic from one VPN routing and forwarding (VRF) instance to another and not for routing between subnets in a global routing table. For more information on NAT on-a-stick, see http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080094430.shtml.

Enabling a Dynamic NAT Virtual Interface

Perform this task to enable a dynamic NVI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat enable**
5. **exit**
6. **ip nat pool** *name start-ip end-ip* **netmask** *netmask* **add-route**
7. **ip nat source list** *access-list-number* **pool** *number* **vrf** *name*
8. **ip nat source list** *access-list-number* **pool** *number* **vrf** *name* **overload**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet l`	Configures an interface type and enters interface configuration mode.
Step 4	**ip nat enable** **Example:** `Router(config-if)# ip nat enable`	Configures an interface that connects VPNs and the Internet for NAT.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 6	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **add-route** **Example:** `Router(config)# ip nat pool pool1 192.168.200.225 192.168.200.254 netmask 255.255.255.0 add-route`	Configures a NAT pool and the associated mappings.
Step 7	**ip nat source list** *access-list-number* **pool** *number* **vrf** *name* **Example:** `Router(config)# ip nat source list 1 pool pool1 vrf vrf1`	Configures an NVI without an inside or outside specification for the specified customer.
Step 8	**ip nat source list** *access-list-number* **pool** *number* **vrf** *name* **overload** **Example:** `Router(config)# ip nat source list 1 pool 1 vrf vrf2 overload`	Configures an NVI without an inside or outside specification for the specified customer.

	Command or Action	Purpose
Step 9	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Enabling a Static NAT Virtual Interface

Perform this task to enable a static NVI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat enable**
5. **exit**
6. **ip nat source static** *local -ip global-ip* **vrf** *name*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet l`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip nat enable** **Example:** `Router(config-if)# ip nat enable`	Configures an interface that connects VPNs and the Internet for NAT.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 6	**ip nat source static** *local -ip global-ip* **vrf** *name* **Example:** `Router(config)# ip nat source static 192.168.123.1 192.168.125.10 vrf vrf1`	Configures a static NVI.
Step 7	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Translating Rotary Addresses

Perform this task to configure server TCP load balancing by way of destination address rotary translation. The commands specified in the task allow you to map one virtual host to many real hosts. Each new TCP session opened with the virtual host will be translated into a session with a different real host.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **type rotary**
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside destination-list** *access-list-number* **pool** *name*
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **type rotary** **Example:** `Router(config)# ip nat pool real-hosts 192.168.201.2 192.168.201.5 prefix-length 28 type rotary`	Defines a pool of addresses containing the addresses of the real hosts.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 192.168.201.30 0.0.0.255`	Defines an access list permitting the address of the virtual host.

	Command or Action	Purpose
Step 5	**ip nat inside destination-list** *access-list-number* **pool** *name* **Example:** `Router(config)# ip nat inside destination-list 2 pool real-hosts`	Establishes dynamic inside destination translation, specifying the access list defined in the prior step.
Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.201.1 255.255.255.240`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface serial 0`	Specifies a different interface and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.15.129 255.255.255.240`	Sets a primary IP address for the interface.

	Command or Action	Purpose
Step 12	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 13	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Enabling Route Maps on Inside Interfaces

Perform this task to use route maps for address translation decisions.

All route maps required for use with this task should be configured before you begin the configuration task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** *local-ip global-ip* [**route-map** *map-name*]}
4. **exit**
5. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** *local-ip global-ip* [**route-map** *map-name*]} **Example:** `Router(config)# ip nat inside source static 192.168.201.6 192.168.201.21 route-map isp2`	Enables route mapping with static NAT configured on the NAT inside interface.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations`	(Optional) Displays active NAT.

Enabling NAT Route Maps Outside-to-Inside Support

The NAT Route Maps Outside-to-Inside Support feature enables the deployment of a NAT route map configuration that will allow IP sessions to be initiated from the outside to the inside. Perform this task to enable the NAT Route Maps Outside-to-Inside Support feature.

An initial session from inside-to-outside is required to trigger a NAT. New translation sessions can then be initiated from outside to the inside host that triggered the initial translation.

When route maps are used to allocate global addresses, the global address can allow return traffic, and the return traffic is allowed only if it matches the defined route map in the reverse direction. No additional entries are created to allow the return traffic for a route-map-based dynamic entry unless the **reversible** keyword is used with the **ip nat inside source** command.

Note

- Access lists with reversible route maps must be configured to match the inside-to-outside traffic.
- In Cisco IOS Release 12.2(33)SXI5, the NAT Route Maps Outside-to-Inside Support feature is supported only on Cisco ME 6500 series Ethernet switches.
- Match-interface or Match Next-hop is not supported for reversible route maps.
- Only IP hosts that are part of the route-map configuration will allow outside sessions.
- Outside-to-inside support is not available with PAT.
- Outside sessions must use an access list.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start* **-** *ip end* **-** *ip* **netmask** *netmask*
4. **ip nat pool** *name start* **-** *ip end* **-** *ip* **netmask** *netmask*
5. **ip nat inside source route-map** *name* **pool** *name* [**reversible**]
6. **ip nat inside source route-map** *name* **pool** *name* [**reversible**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router(config)# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start* **-** *ip end* **-** *ip* **netmask** *netmask* **Example:** `Router(config)# ip nat pool POOL-A 192.168.201.4 192.168.201.6 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 4	**ip nat pool** *name start* **-** *ip end* **-** *ip* **netmask** *netmask* **Example:** `Router(config)# ip nat pool POOL-B 192.168.201.7 192.168.201.9 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 5	**ip nat inside source route-map** *name* **pool** *name* [**reversible**] **Example:** `Router(config)# ip nat inside source route-map MAP-A pool POOL-A reversible`	Enables outside-to-inside initiated sessions to use route maps for destination-based NAT.

	Command or Action	Purpose
Step 6	**ip nat inside source route-map** *name* **pool** *name* [**reversible**] **Example:** `Router(config)# ip nat inside source route-map MAP-B pool POOL-B reversible`	Enables outside-to-inside initiated sessions to use route maps for destination-based NAT.
Step 7	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring NAT of External IP Addresses Only

When you configure NAT of external IP addresses, NAT can be configured to ignore all embedded IP addresses for any application and traffic type. Traffic between a host and the traffic outside an enterprise's network flows through the internal network. A router configured for NAT translates the packet to an address that can be routed inside the internal network. If the intended destination is outside an enterprise's network, the packet gets translated back to an external address and is sent out.

Benefits of Configuring NAT of External IP Addresses Only are:

- Supports public and private network architecture with no specific route updates.
- Gives the end client a usable IP address at the starting point. This address will be the address used for IPsec connections and traffic.
- Allows the use of network architecture that requires only the header translation.
- Allows an enterprise to use the Internet as its enterprise backbone network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**list** {*access-list-number*| *access-list-name*} **pool** *pool-name* [**overload**] | **static network** *local-ip global-ip* [**no-payload**]}
4. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *local-ip local-port global-ip global-port* [**no-payload**]}
5. **ip nat inside source** {**list** {*access-list-number* | *access-list-name}* **pool** *pool-name* [**overload**] |**static** [**network**] *local-network-mask global-network-mask* [**no-payload**]}
6. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** *local-ip global-ip* [**no-payload**]}
7. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *local-ip local-port global-ip global-port* [**no-payload**]}
8. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]}
9. **exit**
10. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**list** {*access-list-number*| *access-list-name*} **pool** *pool-name* [**overload**] | **static network** *local-ip global-ip* [**no-payload**]} **Example:** `Router(config)# ip nat inside source static network 10.1.1.1 192.168.251.0/24 no-payload`	Disables the network packet translation on the inside host router.

	Command or Action	Purpose
Step 4	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *local-ip local-port global-ip global-port* [**no-payload**]} **Example:** `Router(config)# ip nat inside source static tcp 10.1.1.1 2000 192.168.1.1 2000 no-payload`	Disables port packet translation on the inside host router.
Step 5	**ip nat inside source** {**list** {*access-list-number* | *access-list-name}* **pool** *pool-name* [**overload**] |**static** [**network**] *local-network-mask global-network-mask* [**no-payload**]} **Example:** `Router(config)# p nat inside source static 10.1.1.1 192.168.1.1 no-payload`	Disables the packet translation on the inside host router.
Step 6	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** *local-ip global-ip* [**no-payload**]} **Example:** `Router(config)# ip nat outside source static 10.1.1.1 192.168.1.1 no-payload`	Disables packet translation on the outside host router.
Step 7	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *local-ip local-port global-ip global-port* [**no-payload**]} **Example:** `Router(config)# ip nat outside source static tcp 10.1.1.1 20000 192.168.1.1 20000 no-payload`	Disables port packet translation on the outside host router.
Step 8	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]} **Example:** `Router(config)# ip nat outside source static network 10.1.1.1 192.168.251.0/24 no-payload`	Disables network packet translation on the outside host router.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 10	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations`	Displays active NAT.

Forwarding Packets from Outside to Inside Local Address

The NAT Default Inside Server feature helps forward packets from the outside to a specified inside local address. Traffic that does not match any existing dynamic translations or static port translations is redirected, and the packets are not dropped. For online games, outside traffic comes on a different UDP.

Dynamic mapping and interface overload can be configured for the PC traffic and also for the gaming device. If a packet is destined for the 806 interface from outside an enterprise’s network and there no match in the NAT table for the fully extended entry or the static port entry, the packet is forwarded to the gaming device using a simple static entry.

Note

- You can use the feature to configure gaming devices with an IP address that is different from that of the PC. To avoid unwanted traffic or attacks, use access lists.
- For traffic going from the PC to the outside world, it is better to use a route map so that extended entries are created.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip* **interface** *type number*
4. **ip nat inside source static tcp** *local-ip local-port* **interface** *global-port*
5. **exit**
6. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip* **interface** *type number* **Example:** `Router(config)# ip nat inside source static 10.1.1.1 interface Ethernet 1/1`	Enables static NAT on the interface.
Step 4	**ip nat inside source static tcp** *local-ip local-port* **interface** *global-port* **Example:** `Router(config)# ip nat inside source static tcp 10.1.1.1 23 interface 23`	(Optional) Enables the use of telnet to the router from the outside.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations`	(Optional) Displays active NAT.

Reenabling RTSP on a NAT Router

The Real Time Streaming Protocol (RTSP) is a client/server multimedia presentation control protocol that supports multimedia application delivery. Some of the applications that use RTSP include Windows Media Services (WMS) by Microsoft, QuickTime by Apple Computer, and RealSystem G2 by RealNetworks.

When the RTSP protocol passes through a NAT router, the embedded address and port must be translated in order for the connection to be successful. NAT uses Network Based Application Recognition (NBAR) architecture to parse the payload and translate the embedded information in the RTSP payload.

RTSP is enabled by default. Use the **ip nat service rtsp port** *port-number* command to re-enable RTSP on a NAT router if this configuration has been disabled.

Configuring Static IP Support

Configuring support for users with static IP addresses enables those users to establish an IP session in a public wireless LAN environment.

The NAT Static IP Support feature extends the capabilities of public wireless LAN providers to support users configured with a static IP address. By configuring a router to support users with a static IP address, public wireless LAN providers extend their services to a greater number of potential users, which can lead to greater user satisfaction and additional revenue.

Users with static IP addresses can use services of the public wireless LAN provider without changing their IP address. NAT entries are created for static IP clients and a routable address is provided.

Perform this task to configure the NAT Static IP Support feature.

Before configuring support for users with static IP addresses for NAT, you must first enable NAT on your router and configure a RADIUS server host. For additional information on NAT and RADIUS configuration, see the "Related Documents" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat inside**
5. **exit**
6. **ip nat allow-static-host**
7. **ip nat pool** *name start-ip end-ip* **netmask** *netmask* **accounting** *list-name*
8. **ip nat inside source list** *access-list-number* **pool** *name*
9. **access-list** *access-list-number* **deny ip** *source*
10. **end**
11. **show ip nat translations verbose**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1`	Specifies the interface to be configured, and enters interface configuration mode.
Step 4	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**ip nat allow-static-host** **Example:** `Router(config)# ip nat allow-static-host`	Enables static IP address support. • Dynamic Address Resolution Protocol (ARP) learning will be disabled on this interface, and NAT will control the creation and deletion of ARP entries for the static IP host.
Step 7	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **accounting** *list-name* **Example:** `Router(config)# ip nat pool pool1 172.16.0.0 172.16.0.0 netmask 255.255.255.0 accounting WLAN-ACCT`	Specifies an existing RADIUS profile name to be used for authentication of the static IP host.
Step 8	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Router(config)# ip nat inside source list 1 pool net-208`	Specifies the access list and pool to be used for static IP support. • The specified access list must permit all traffic.

	Command or Action	Purpose
Step 9	**access-list** *access-list-number* **deny ip** *source* **Example:** `Router(config)# access-list 1 deny ip 192.168.196.51`	Removes the router's own traffic from NAT. • The *source* argument is the IP address of the router that supports the NAT Static IP Support feature.
Step 10	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 11	**show ip nat translations verbose** **Example:** `Router# show ip nat translations verbose`	(Optional) Displays active NAT translations and additional information for each translation table entry, including how long ago the entry was created and used.

Examples

The following is sample output from the **show ip nat translations verbose** command:

```
Router# show
 ip
 nat
 translations
 verbose
--- 172.16.0.0 10.1.1.1           ---                ---
create 00:05:59, use 00:03:39, left 23:56:20, Map-Id(In): 1, flags: none wlan-flags:
Secure ARP added, Accounting Start sent Mac-Address:0010.7bc2.9ff6 Input-IDB:Ethernet1/2,
use_count: 0, entry-id:7, lc_entries: 0
```

Configuring Support for ARP Ping

When the static IP client's NAT entry times out, the NAT entry and the secure ARP entry associations are deleted for the client. Reauthentication with the Service Selection Gateway (SSG) is needed for the client to reestablish WLAN services. The ARP Ping feature enables the NAT entry and the secure ARP entry to not be deleted when the static IP client exists in the network where the IP address is unchanged after authentication.

An ARP ping is necessary to determine static IP client existence and to restart the NAT entry timer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start* **-** *ip end* **-** *ip* **prefix-length** *prefix-length* [**accounting** *method* **-** *list* **-** *name*] [**arp-ping**]
4. **ip nat translation arp -ping-timeout** [*seconds*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start* **-** *ip end* **-** *ip* **prefix-length** *prefix-length* [**accounting** *method* **-** *list* **-** *name*] [**arp-ping**] **Example:** `Router(config)# ip nat pool net-208 172.16.233.208 172.16.233.223 prefix-length 28 accounting radius1 arp-ping`	Defines a pool of IP addresses for NAT.
Step 4	**ip nat translation arp -ping-timeout** [*seconds*] **Example:** `Router(config)# ip nat translation arp-ping-timeout 600`	Changes the amount of time after each network address translation.
Step 5	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Limiting the Number of Concurrent NAT Operations

Limiting the number of concurrent NAT operations using the Rate Limiting NAT Translation feature provides users more control over how NAT addresses are used. The Rate Limiting NAT Translation feature can be used to limit the effects of viruses, worms, and denial-of-service attacks.

Because NAT is a CPU-intensive process, router performance can be adversely affected by denial-of-service attacks, viruses, and worms that target NAT. The Rate Limiting NAT Translation feature allows you to limit the maximum number of concurrent NAT requests on a router.

- Classify current NAT usage and determine the sources of requests for NAT. A specific host, access control list, or VRF instance generating an unexpectedly high number of NAT requests may be the source of a malicious virus or worm attack.
- Once you have identified the source of excess NAT requests, you can set a NAT rate limit that contains a specific host, access control list, or VRF instance, or you can set a general limit for the maximum number of NAT requests allowed regardless of their source.

SUMMARY STEPS

1. **enable**
2. **show ip nat translations**
3. **configure terminal**
4. **ip nat translation max-entries** {*number* | **all-vrf** *number* | **host** *ip-address number* | **list** *listname number* | **vrf** *name number*}
5. **end**
6. **show ip nat statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip nat translations** **Example:** `Router# show ip nat translations`	(Optional) Displays active NAT. • A specific host, access control list, or VRF instance generating an unexpectedly high number of NAT requests may be the source of a malicious virus or worm attack.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 4	**ip nat translation max-entries** {*number* | **all-vrf** *number* | **host** *ip-address number* | **list** *listname number* | **vrf** *name number*} **Example:** `Router(config)# ip nat translation max-entries 300`	Configures the maximum number of NAT entries allowed from the specified source. • The maximum number of allowed NAT entries is 2147483647, although a typical range for a NAT rate limit is 100 to 300 entries. • When you configure a NAT rate limit for all VRF instances, each VRF instance is limited to the maximum number of NAT entries that you specify. • When you configure a NAT rate limit for a specific VRF instance, you can specify a maximum number of NAT entries for the named VRF instance that is greater than or less than that allowed for all VRF instances.
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip nat statistics** **Example:** `Router# show ip nat statistics`	(Optional) Displays current NAT usage information, including NAT rate limit settings. • After setting a NAT rate limit, use the **show ip nat statistics** command to verify current NAT rate limit settings.

Configuration Examples for Configuring NAT for IP Address Conservation

Example Configuring Static Translation of Inside Source Addresses

The following example translates between inside hosts addressed from the 10.114.11.0 network to the globally unique 172.31.233.208/28 network. Further packets from outside hosts addressed from the

10.114.11.0 network (the true 10.114.11.0 network) are translated to appear to be from the 10.0.1.0/24 network.

```
ip nat pool net-208 172.31.233.208 172.31.233.223 prefix-length 28
ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24
ip nat inside source list 1 pool net-208
ip nat outside source list 1 pool net-10
!
interface ethernet 0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
interface ethernet 1
 ip address 10.114.11.39 255.255.255.0
 ip nat inside
!
access-list 1 permit 10.114.11.0 0.0.0.255
```

The following example shows NAT configured on the provider edge (PE) router with a static route to the shared service for the vrf1 and vrf2 VPNs. NAT is configured as inside source static one-to-one translation.

```
ip nat pool outside 10.4.4.1 10.4.4.254 netmask 255.255.255.0
ip nat outside source list 1 pool mypool
access-list 1 permit 172.16.18.0 0.0.0.255
ip nat inside source static 192.168.121.33 10.2.2.1 vrf vrf1
ip nat inside source static 192.169.121.33.10.2.2.2 vrf vrf2
```

Example Configuring Dynamic Translation of Inside Source Addresses

The following example translates between inside hosts addressed from either the 192.168.1.0 or 192.168.2.0 network to the globally unique 172.31.233.208/28 network:

```
ip nat pool net-208 172.31.233.208 172.31.233.223 prefix-length 9
ip nat inside source list 1 pool net-208
!
interface ethernet 0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
interface ethernet 1
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 1 permit 192.168.2.0 0.0.0.255
```

The following example translates only traffic local to the provider edge device running NAT (NAT-PE):

```
ip nat inside source list 1 interface e 0 vrf vrf1 overload
ip nat inside source list 1 interface e 0 vrf vrf2 overload
!
ip route vrf vrf1 0.0.0.0 0.0.0.0 192.168.1.1
ip route vrf vrf2 0.0.0.0 0.0.0.0 192.168.1.1
!
access-list 1 permit 10.1.1.1.0 0.0.0.255
!
ip nat inside source list 1 interface e 1 vrf vrf1 overload
ip nat inside source list 1 interface e 1 vrf vrf2 overload
!
ip route vrf vrf1 0.0.0.0 0.0.0.0 172.16.1.1 global
ip route vrf vrf2 0.0.0.0 0.0.0.0 172.16.1.1 global
access-list 1 permit 10.1.1.0 0.0.0.255
```

Example Overloading Inside Global Addresses

The following example creates a pool of addresses named net-208. The pool contains addresses from 172.31.233.208 to 172.31.233.233. Access list 1 allows packets having the SA from 192.168.1.0 to

192.168.1.255. If no translation exists, packets matching access list 1 are translated to an address from the pool. The router allows multiple local addresses (192.168.1.0 to 192.168.1.255) to use the same global address. The router retains port numbers to differentiate the connections.

```
ip nat pool net-208 172.31.233.208 172.31.233.233 netmask 255.255.255.240
ip nat inside source list 1 pool net-208 overload
!
interface serial 0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
interface ethernet 0
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
```

Example Translating Overlapping Address

In the following example, the addresses in the local network are being used legitimately by someone else on the Internet. An extra translation is required to access that external network. Pool net-10 is a pool of outside local IP addresses. The **ip nat outside source list 1 pool net-10**statementtranslates the addresses of hosts from the outside overlapping network to addresses in that pool.

```
ip nat pool net-208 172.31.233.208 172.31.233.223 prefix-length 28
ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24
ip nat inside source list 1 pool net-208
ip nat outside source list 1 pool net-10
!
interface serial 0
 ip address 172.31.232.192 255.255.255.240
 ip nat outside
!
interface ethernet0
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
```

Example Enabling NAT Virtual Interface

The following example shows how to configure NAT virtual interfaces without the use of inside or outside source addresses:

```
interface Ethernet 0/0
 ip vrf forwarding bank
 ip address 192.168.122.1 255.255.255.0
 ip nat enable
!
interface Ethernet 1/0
 ip vrf forwarding park
 ip address 192.168.122.1 255.255.255.0
 ip nat enable
!
interface Serial 2/0
 ip vrf forwarding services
 ip address 192.168.123.2 255.255.255.0
ip nat enable
!
ip nat pool NAT 192.168.25.20 192.168.25.30 netmask 255.255.255.0 add-route
ip nat source list 1 pool NAT vrf vrf1 overload
ip nat source list 1 pool NAT vrf vrf2 overload
ip nat source static 192.168.123.1 192.168.125.10 vrf services
!
access-list 1 permit 192.168.122.20
```

```
access-list 1 permit 192.168.122.0 0.0.0.255
!
```

Example Avoiding Server Overload Using Load Balancing

In the following example, the goal is to define a virtual address, connections to which are distributed among a set of real hosts. The pool defines the addresses of the real hosts. The access list defines the virtual address. If a translation does not already exist, TCP packets from serial interface 0 (the outside interface) whose destination matches the access list are translated to an address from the pool.

```
ip nat pool real-hosts 192.168.15.2 192.168.15.15 prefix-length 28 type rotary
ip nat inside destination list 2 pool real-hosts
!
interface serial 0
 ip address 192.168.15.129 255.255.255.240
 ip nat outside
!
interface ethernet 0
 ip address 192.168.15.17 255.255.255.240
 ip nat inside
!
access-list 2 permit 192.168.15.1
```

Example Enabling NAT Route Mapping

The following example shows the use of route mapping with static NATs:

```
interface Ethernet3
 ip address 172.18.1.100 255.255.255.0
 ip nat outside
 media-type 10BaseT
!
interface Ethernet4
 ip address 192.168.1.100 255.255.255.0
 ip nat outside
 media-type 10BaseT
!
interface Ethernet5
 ip address 110.1.1.100 255.255.255.0
 ip nat inside
 ip policy route-map isp1
 media-type 10BaseT
!
router rip
 network 172.18.0.0
 network 192.168.1.0
!
 ip nat inside source static 10.1.1.2 192.168.1.21 route-map isp2
 ip nat inside source static 10.1.1.2 172.18.1.21 route-map isp1
 ip nat inside source static 10.1.1.1 192.168.1.11 route-map isp2
 ip nat inside source static 10.1.1.1 172.18.1.11 route-map isp1
 access-list 101 permit ip 10.1.1.0 0.0.0.255 172.16.0.0 0.255.255.255
 access-list 102 permit ip 10.1.1.0 0.0.0.255 192.168.0.0 0.255.255.255
!
route-map isp2 permit 10
 match ip address 102
 set ip next-hop 192.168.1.1
!
route-map isp1 permit 10
 match ip address 101
 set ip next-hop 172.18.1.1
```

Example Enabling NAT Route Maps Outside-to-Inside Support

The following example shows how to configure route map A and route map B to allow outside-to-inside translation for a destination-based NAT:

```
ip nat pool POOL-A 10.1.10.1 10.1.10.126 netmask 255.255.255.128
ip nat pool POOL-B 10.1.20.1 10.1.20.126 netmask 255.255.255.128
ip nat inside source route-map MAP-A pool POOL-A reversible
ip nat inside source route-map MAP-B pool POOL-B reversible
!
ip access-list extended ACL-A
 permit ip any 10.1.10.128 0.0.0.127
ip access-list extended ACL-B
 permit ip any 10.1.20.128 0.0.0.127
!
route-map MAP-A permit 10
 match ip address ACL-A
!
route-map MAP-B permit 10
 match ip address ACL-B
```

The following example shows how to configure route map R1 to allow outside-to-inside translation for static NAT:

```
ip nat inside source static 10.1.1.1 10.2.2.2 route-map R1 reversible
!
ip access-list extended ACL-A
 permit ip any 10.1.10.128 0.0.0.127
route-map R1 permit 10
 match ip address ACL-A
```

Example Configuring NAT Translation of External IP Addresses Only

The following example shows how to translate the packet to an address that is able to be routed inside the internal network:

```
interface ethernet 3
ip address 10.1.1.1 255.255.255.0
ip nat outside
no ip mroute-cache
media-type 10BaseT
!
interface Ethernet4
ip address 192.168.15.1 255.255.255.0
ip nat inside
no ip mroute-cache
media-type 10BaseT
!
router rip
network 10.0.0.0
Network 192.168.15.0
!
ip nat outside source static network 10.1.1.0 192.168.251.0/24 no-payload
!
ip route 10.1.1.0 255.255.255.0 Ethernet4
ip route 10.1.1.0 255.255.255.0 Ethernet3
```

Configuration Examples for NAT Static IP Support

- Example Configuring NAT Static IP Support, page 51
- Example Creating a RADIUS Profile for NAT Static IP Support, page 51

Example Configuring NAT Static IP Support

The following example shows how to enable static IP address support for the router at 192.168.196.51:

```
interface ethernet 1
 ip nat inside
ip nat allow-static-host
ip nat pool net-208 172.16.1.1 172.16.1.10 netmask 255.255.255.0 accounting WLAN-ACCT
ip nat inside source list 1 pool net-208
access-list 1 deny ip 192.168.196.51
```

Example Creating a RADIUS Profile for NAT Static IP Support

The following example shows how to create a RADIUS profile for use with the NAT Static IP Support feature:

```
aaa new-model

!

aaa group server radius WLAN-RADIUS

 server 172.16.88.1 auth-port 1645 acct-port 1645

 server 172.16.88.1 auth-port 1645 acct-port 1646

!

aaa accounting network WLAN-ACCT start-stop group WLAN-RADIUS

aaa session-id common

ip radius source-interface Ethernet3/0

radius-server host 172.31.88.1 auth-port 1645 acct-port 1646

radius-server key cisco
```

Configuration Examples for Limiting the Number of Concurrent NAT Operations

This section provides the following configuration examples:

Example Setting a Global NAT Rate Limit

The following example shows how to limit the maximum number of allowed NAT entries to 300:

```
ip nat translation max-entries 300
```

Example Setting NAT Rate Limits for a Specific VRF Instance

The following example shows how to limit the VRF instance named “vrf1” to 150 NAT entries:

```
ip nat translation max-entries vrf vrf1 150
```

Example Setting NAT Rate Limits for All VRF Instances

The following example shows how to limit each VRF instance to 200 NAT entries:

```
ip nat translation max-entries all-vrf 200
```

The following example shows how to limit the VRF instance named “vrf2” to 225 NAT entries, but limit all other VRF instances to 100 NAT entries each:

```
ip nat translation max-entries all-vrf 100
ip nat translation max-entries vrf vrf2 225
```

Example Setting NAT Rate Limits for Access Control Lists

The following example shows how to limit the access control list named “vrf3” to 100 NAT entries:

```
ip nat translation max-entries list vrf3 100
```

Example Setting NAT Rate Limits for an IP Address

The following example shows how to limit the host at IP address 10.0.0.1 to 300 NAT entries:

```
ip nat translation max-entries host 10.0.0.1 300
```

Where to Go Next

- To configure NAT for use with application level gateways, see the “Using Application Level Gateways with NAT” module.
- To verify, monitor, and maintain NAT, see the “Monitoring and Maintaining NAT” module.
- To integrate NAT with Multiprotocol Label Switching (MPLS) VPNs, see the “Integrating NAT with MPLS VPNs” module.
- To configure NAT for high availability, see the “Configuring NAT for High Availability” module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
NAT commands: complete command syntax, command mode command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
Application Level Gateways	“Using Application Level Gateways with NAT” module
IP Access List Sequence Numbering	*IP Access List Sequence Numbering* document
NAT on a Stick technology note	Network Address Translation on a Stick technology note
NAT maintenance	“Monitoring and Maintaining NAT” module
RADIUS attributes overview	“RADIUS Attributes Overview and RADIUS IETF Attributes” module
Using Hot Standby Router Protocol (HSRP) and Stateful NAT (SNAT) for high availability	“Configuring NAT for High Availability” module
Using NAT with MPLS VPNs	“Integrating NAT with MPLS VPNs” module

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1597	*Internet Assigned Numbers Authority*
RFC 1631	*The IP Network Address Translation (NAT)*
RFC 1918	*Address Allocation for Private Internets*

RFCs	Title
RFC 2663	*IP Network Address Translation (NAT) Terminology and Considerations*
RFC 3022	*Traditional IP Network Address Translation (Traditional NAT)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring NAT for IP Address Conservation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for Configuring NAT for IP Address Conservation

Feature Name	Releases	Feature Configuration Information
Configuring Support for ARP Ping in a Public Wireless LAN	12.4(6)T	The ARP Ping feature enables the NAT entry and the secure ARP entry to not be deleted when the static IP client exists in the network where the IP address is unchanged after authentication.

Feature Name	Releases	Feature Configuration Information
NAT Ability to Use Route Maps with Static Translation	12.2.(4)T	The NAT Ability to Use Route Maps with Static Translation feature provides a dynamic translation command that can specify a route map to be processed instead of an access list. A route map allows you to match any combination of access list, next-hop IP address, and output interface to determine which pool to use. The ability to use route maps with static translations enables NAT multihoming capability with static address translations.
NAT Default Inside Server	12.3(13)T	The NAT Default Inside Server feature provides for the need to forward packets from the outside to a specified inside local address.
NAT Route Maps Outside-to-Inside Support	12.2(33)SXI5 12.3(14)T	The NAT Route Maps Outside-to-Inside Support feature enables the deployment of a NAT route map configuration that will allow IP sessions to be initiated from the outside to the inside.
NAT RTSP Support Using NBAR	12.3(7)T	The Real Time Streaming Protocol (RTSP) is a client/server multimedia presentation control protocol that supports multimedia application delivery. Applications that use RTSP include Windows Media Services (WMS) by Microsoft, QuickTime by Apple Computer, and RealSystem G2 by RealNetworks.
NAT Static and Dynamic Route Map Name-Sharing	15.0(1)M	The NAT Static and Dynamic Route Map Name-Sharing feature provides the ability to configure static and dynamic NAT to share the same route map name, while enforcing precedence of static NAT over dynamic NAT.

Feature Name	Releases	Feature Configuration Information
NAT Static IP Support	12.3(7)T	The NAT Static IP Support feature provides support for users with static IP addresses, enabling those users to establish an IP session in a public wireless LAN environment.
NAT Translation of External IP Addresses Only	12.2(4)T 12.2(4)T2 15.0(1)S	You can use the NAT Translation of External IP Addresses Only feature, NAT can be configured to ignore all embedded IP addresses for any application and traffic type.
NAT Virtual Interface	12.3(14)T	The NAT Virtual Interface feature removes the requirement to configure an interface as either Network Address Translation (NAT) inside or NAT outside. An interface can be configured to use NAT or not use NAT.
Rate Limiting NAT Translation	12.3(4)T 15.0(1)S	The Rate Limiting NAT Translation feature provides the ability to limit the maximum number of concurrent Network Address Translation (NAT) operations on a router. In addition to giving users more control over how NAT addresses are used, the Rate Limiting NAT Translation feature can be used to limit the effects of viruses, worms, and denial-of-service attacks.

Using Application Level Gateways with NAT

This module describes the basic tasks to configure an Application Level Gateway (ALG) with Network Address Translation (NAT). This module also provides information about the protocols that use ALG for IP header translation.

NAT performs translation service on any TCP/UDP traffic that does not carry the source and destination IP addresses in the application data stream. These protocols include HTTP, TFTP, telnet, archie, finger, Network Time Protocol (NTP), Network File System (NFS), remote login (rlogin), remote shell (rsh) protocol, and remote copy (rcp). Specific protocols that do embed IP the address information within the payload require support of an ALG.

NAT with an ALG will translate packets from applications that do not use H.323, as long as the applications use port 1720.

The Support for IPsec ESP Through NAT feature provides the ability to support multiple concurrent IPsec Encapsulating Security Payload (ESP) tunnels or connections through a Cisco IOS NAT device configured in Overload or Port Address Translation (PAT) mode.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Using Application Level Gateways with NAT

- Before performing the tasks in this module, you should be familiar with the concepts described in the “Configuring NAT for IP Address Conservation” module.
- All access lists required for use with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the “IP Access List Sequence Numbering” document.
- Before performing the tasks in this module, you should verify that the Session Initiation Protocol (SIP) and H.323 are not disabled. SIP and H.323 are enabled by default.

Restrictions for Using Application Level Gateways with NAT

NAT will translate only embedded IP version 4 addresses.

Information About Using Application Level Gateways with NAT

Application Level Gateway

An application level gateway is an application that translates IP address information inside the payload of an applications packet.

Benefits of Configuring NAT ALG

- NAT support for SIP adds the ability to deploy Cisco IOS NAT between VoIP solutions based on SIP.
- Customers can control their IP address scheme and include complete support for H.323 v2 gatekeeper designs.
- NAT enables customers to deploy private IP addresses within their network and perform translation to public IP addresses when connecting to the Internet or interconnecting with another corporate network.
- Normally ESP entries in the translation table are delayed from being transmitted until a reply is received from the destination. With predictable security parameter indexes (SPIs) and SPI matching, the delay can be eliminated because the SPI entries are matched. Some third-party concentrators require both the source and incoming ports to use port 500. Use of the **preserve-port** keyword with the **ip nat service** command preserves the ports rather than changing one, which is required with regular NAT.

IP Security

IPsec is a set of extensions to the IP protocol family in a framework of open standards for ensuring secure private communications over the Internet. Based on standards developed by the IETF, IPsec ensures confidentiality, integrity, and authenticity of data communications across the public network and provides cryptographic security services.

Secure tunnels between two peers, such as two routers, are provided and decisions are made as to which packets are considered sensitive and should be sent through these secure tunnels, and which parameters should be used to protect these sensitive packets by specifying characteristics of these tunnels. When the IPsec peer receives a sensitive packet, it sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

IPsec using ESP can pass through a router running NAT without any specific support from it as long as Network Address Port Translation (NAPT) or address overloading is not configured.

There are a number of factors to consider when attempting an IPsec VPN connection that traverses a NAPT device that represents multiple private internal IP addresses as a single public external IP address. Such factors include the capabilities of the VPN server and client, the capabilities of the NAPT device, and whether more than one simultaneous connection is attempted across the NAPT device.

There are two possible methods for configuring IPsec on a router with NAPT:

- Encapsulate IPsec in a Layer 4 protocol such as TCP or UDP. In this case, IPsec is *sneaking* through NAT. The NAT device is unaware of the encapsulation.
- Add IPsec specific support to NAPT. IPsec works with NAT in this case as opposed to *sneaking* through NAT. The NAT Support for IPsec ESP-- Phase II feature provides support for Internet Key Exchange (IKE) and ESP without encapsulation in tunnel mode through a Cisco IOS router configured with NAPT.

The recommended protocols to use when conducting IPsec sessions that traverse a NAPT device are TCP and UDP, but not all VPN servers or clients support TCP or UDP.

SPI Matching

Security Parameter Index (SPI) matching is used to establish VPN connections between multiple pairs of destinations. NAT entries will immediately be placed in the translation table for endpoints matching the configured access list. SPI matching is available only for endpoints that choose SPIs according to the predictive algorithm implemented in Cisco IOS Release 12.2(15)T.

Voice and Multimedia over IP Networks

SIP is a protocol developed by the IETF Multiparty Multimedia Session Control (MMUSIC) Working Group. The Cisco SIP functionality equips Cisco routers to signal the setup of voice and multimedia calls over IP networks. SIP provides an alternative to H.323 within the VoIP internetworking software.

Session Description Protocol (SDP) is a protocol that describes multimedia sessions. SDP may be used in SIP message bodies to describe multimedia sessions used for creating and controlling multimedia sessions with two or more participants.

The NAT Support for SIP feature allows SIP embedded messages passing through a router configured with NAT to be translated and encoded back to the packet. An ALG is used with NAT to translate the SIP or SDP messages.

Note By default support for SIP is enabled on port 5060. Therefore, NAT-enabled devices interpret all packets on this port as SIP call messages. If other applications in the system use port 5060 to send packets, the NAT service may corrupt the packet as it attempts to interpret the packet as a SIP call message.

NAT Support of H.323 v2 RAS

Cisco IOS NAT supports all H.225 and H.245 message types, including those sent in the Registration, Admission, and Status (RAS) protocol. RAS provides a number of messages that are used by software clients and VoIP devices to register their location, request assistance in call setup, and control bandwidth. The RAS messages are directed toward an H.323 gatekeeper.

Some RAS messages include IP addressing information in the payload, typically meant to register a user with the gatekeeper or learn about another user already registered. If these messages are not known to NAT, they cannot be translated to an IP address that will be visible to the public.

In Cisco IOS Release 12.2(2)T and later releases, embedded IP addresses can be inspected for potential address translation. Prior to Cisco IOS Release 12.2(2)T, NAT did not support H.323 v2 RAS messages.

NAT Support for H.323 v3 and v4 in v2 Compatibility Mode

H.323 is an ITU-T specification for transmitting audio, video, and data across a packet network. NAT supports four versions of the H.323 protocols: v1, v2, v3, and v4. The NAT Support for H.323 v3 and v4 in v2 Compatibility Mode feature enables Cisco NAT routers to support messages coded in H.323 v3 and v4 when those messages contain fields compatible with H.323 v2. This feature does not add support for H.323 capabilities introduced in v3 and v4, such as new message types or new fields that require address translation.

NAT H.245 Tunneling Support

NAT H.245 tunneling allows H.245 tunneling in H.323 ALGs. NAT H.245 tunneling provides a mechanism for supporting the H.245 tunnel message that is needed to create a media channel setup.

In order for an H.323 call to take place, an H.225 connection on TCP port 1720 needs to be opened. When the H.225 connection is opened, the H.245 session is initiated and established. This connection can take place on a separate channel from the H.225 or it can be done using H.245 tunneling on the same H.225 channel whereby the H.245 messages are embedded in the H.225 messages and sent on the previously established H.225 channel.

If the H.245 tunneled message is not understood, the media address or port will be left untranslated by the Cisco IOS NAT, resulting in failure in media traffic. H.245 FastConnect procedures will not help because FastConnect is terminated as soon as an H.245 tunneled message is sent.

NAT Support of Skinny Client Control Protocol

Cisco IP phones use the SCCP to connect with and register to Cisco CallManager.

To be able to configure Cisco IOS NAT between the IP phone and Cisco CallManager in a scalable environment, NAT needs to be able to detect the SCCP and understand the information passed within the messages. Messages flow back and forth that include IP address and port information used to identify other IP phone users with which a call can be placed.

The SCCP client to Cisco CallManager communication typically flows from inside to outside. Domain Name System (DNS) should be used to resolve the Cisco CallManager IP address connection when the

Cisco CallManager is on the inside (behind the NAT device), or static NAT should be configured to reach the Cisco CallManager in the inside.

When an IP phone attempts to connect to the Cisco CallManager and it matches the configured NAT rules, NAT will translate the original source IP address and replace it with one from the configured pool. This new address will be reflected in the Cisco CallManager and be visible to other IP phone users.

NAT Support of SCCP Fragmentation

Skinny control messages are exchanged over TCP. If either the IP phone or Cisco CallManager has been configured to have a TCP maximum segment size (MSS) lower than the skinny control message payload, the skinny control message will be segmented across multiple TCP segments. Prior to this feature skinny control message exchanges would fail in a TCP segmentation scenario because NAT skinny ALG was not able to reassemble the skinny control messages. The NAT SCCP Fragmentation Support feature adds support for TCP segments for NAT skinny ALG. A fragmented payload that requires an IP or port translation will no longer be dropped.

Skinny control messages can also be IP fragmented but they are supported using Virtual Fragmentation Reassembly (VFR).

In Cisco IOS Release 15.1(3)T and later releases, NAT works with SCCP phones version 17 and higher.

NAT Segmentation with Layer 4 Forwarding

The NAT Segmentation with Layer 4 Forwarding feature is implemented for the H.323, SCCP, and TCP DNS protocols. NAT supports the processing of segmented H.323, SCCP, or TCP DNS messages that are split across multiple packets.

Layer 4 forwarding or TCP proxy is responsible for session handling that includes putting the sequence numbers in order, acknowledging the numbers in a packet, resegmenting the translated packet if it is larger than the MSS, and handling retransmissions in case of packet loss. Layer 4 forwarding also handles out-of-order packets. These packets are buffered and not dropped.

Layer 4 forwarding buffers the received packets and notifies NAT ALG when an in-order packet is available. It also sends acknowledgments to the end hosts for the received packets. Layer 4 forwarding also sends the translated packets that it receives from NAT ALG back into the output packet path.

Restrictions

The NAT Segmentation with Layer 4 Forwarding feature does not work when:

- Cisco IOS firewalls are configured using the **ip inspect name** command. (Zone-based firewalls are supported.)
- H.323, SCCP, or TCP DNS messages are larger than 18 KB.
- Multiprotocol Label Switching (MPLS) is configured.
- NAT and the Cisco CallManager are configured on the same device. In this case, the colocated solution in Call Manager Express (CME) is used.
- NAT Virtual Interface (NVI) is configured.
- Stateful Network Address Translation (SNAT) is enabled.
- The **match-in-vrf** keyword is configured along with the **ip nat inside source** command for packet translation.
- The packets are IPv6 packets.

How to Configure Application Level Gateways with NAT

Configuring IPsec Through NAT

To successfully configure application level gateways with NAT, you should understand the following concepts:

This section contains the following tasks related to configuring IPsec through NAT:

Configuring IPsec ESP Through NAT

IPsec ESP Through NAT provides the ability to support multiple concurrent IPsec ESP tunnels or connections through a Cisco IOS NAT device configured in Overload or PAT mode.

Perform this task to configure IPsec ESP through NAT.

Note IPsec can be configured for any NAT configuration, not just static NAT configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat** [**inside** | **outside**] **source static** *local-ip global-ip* [**vrf** *vrf-name*]
4. **exit**
5. **show ip nat translations**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat** [**inside** | **outside**] **source static** *local-ip global-ip* [**vrf** *vrf-name*] **Example:** `Router(config)# ip nat inside source static 10.10.10.10 192.168.30.30`	Enables static NAT.
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show ip nat translations** **Example:** `Router# show ip nat translations`	(Optional) Displays active NATs.

Enabling the Preserve Port

This task is used for IPsec traffic using port 500 for the source port. Perform this task to enable port 500 to be preserved for the source port.

Note This task is required by certain VPN concentrators. Cisco VPN devices generally do not use this feature.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service list** *access-list-number* **IKE preserve-port**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service list** *access-list-number* **IKE preserve-port** **Example:** `Router(config)# ip nat service list 10 IKE preserve-port`	Specifies IPsec traffic that matches the access list to preserve the port.

Enabling SPI Matching on the NAT Device

Note SPI matching is disabled by default.

Security parameter index (SPI) matching is used to establish VPN connections between multiple pairs of destinations. NAT entries are immediately placed in the translation table for endpoints matching the configured access list. SPI matching is available only for endpoints that choose SPIs according to the predictive algorithm implemented in Cisco IOS Release 12.2(15)T.

The generation of SPIs that are predictable and symmetric is enabled. SPI matching should be used in conjunction with NAT devices when multiple ESP connections across a NAT device are desired.

Cisco IOS software must be running on both the source router and the remote gateway enabling parallel processing.

Note SPI matching must be configured on the NAT device and both endpoint devices.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service list** *access-list-number* **ESP spi-match**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service list** *access-list-number* **ESP spi-match** **Example:** `Router(config)# ip nat service list 10 ESP spi-match`	Specifies an access list to enable SPI matching. • This example shows how to enter ESP traffic matching list 10 into the NAT table, making the assumption that both devices are Cisco devices and are configured to provide matchable SPIs.

Enabling SPI Matching on the Endpoints

Perform this task to enable SPI matching on both endpoints.

Cisco IOS software must be running on both the source router and the remote gateway enabling parallel processing.

Note SPI matching must be configured on the NAT device and both endpoint devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec nat-transparency spi-matching**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec nat-transparency spi-matching** **Example:** `Router(config)# crypto ipsec nat-transparency spi-matching`	Enables SPI matching on both endpoints.

Enabling MultiPart SDP Support for NAT

The MultiPart SDP Support for NAT feature provides support for multipart SDP in a SIP ALG for the Advanced NAT portfolio. MultiPart SDP support for NAT is disabled by default.

Perform this task to enable multipart SDP support for NAT.

Note NAT will translate only embedded IP version 4 addresses.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service allow-multipart**
4. **exit**
5. **show ip nat translations**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service allow-multipart** **Example:** `Router(config)# ip nat service allow-multipart`	Enables multipart SDP.
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show ip nat translations** **Example:** `Router# show ip nat translations`	(Optional) Displays active NATs.

Configuring NAT Between an IP Phone and Cisco CallManager

This section describes configuring Cisco's Skinny Client Control Protocol (SCCP) for Cisco IP phone to Cisco CallManager communication. The task in this section configures NAT between an IP phone and Cisco CallManager.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service skinny tcp port** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service skinny tcp port** *number* **Example:** `Router(config)# ip nat service skinny tcp port 20002`	Configures the skinny protocol on the specified TCP port.

Configuration Examples for Using Application Level Gateways with NAT

Example Configuring IPsec ESP Through NAT

The following example shows NAT configured on the provider edge (PE) router with a static route to the shared service for the vrf1 and vrf2 VPNs. NAT is configured as inside source static 1-to-1 translations.

```
ip nat pool outside 192.0.2.1 192.0.2.14 netmask 255.255.255.0
ip nat outside source list 1 pool mypool
access-list 1 permit 192.0.2.3 0.0.0.255
ip nat inside source static 192.0.2.23 192.0.2.22 vrf vrf1
ip nat inside source static 192.0.2.21 192.0.2.2 vrf vrf2
```

Example Enabling the Preserve Port

The following example shows how to configure TCP port 500 of the third-party concentrator. Access list 10 is configured:

```
ip nat service list 10 IKE preserve-port
access-list 10 permit 10.1.1.1
```

Example Enabling SPI Matching

The following example shows how to enable SPI matching. Access list 10 is configured:

```
ip nat service list 10 ESP spi-match
access-list 10 permit 10.1.1.1
```

Example Configuring SPI Matching on the Endpoint Routers

The following example show how to enable SPI matching on the endpoint routers:

```
crypto ipsec nat-transparency spi-matching
```

Example Enabling MultiPart SDP Support for NAT

The following example shows how to enable multipart SDP support for NAT:

```
ip nat service allow-multipart
```

Example Configuring NAT Between an IP Phone and Cisco CallManager

The following example shows how to configure the 20002 port of the Cisco CallManager:

```
ip nat service skinny tcp port 20002
```

Where to Go Next

- To learn about NAT and configure NAT for IP address conservation, see the “Configuring NAT for IP Address Conservation” module.
- To verify monitor, and maintain NAT, see the “Monitoring and Maintaining NAT” module.
- To integrate NAT with MPLS VPNs, see the “Integrating NAT with MPLS VPNs” module.
- To configure NAT for high availability, see the “Configuring NAT for High Availability” module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
NAT commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
IP access list sequence numbering	"IP Access List Sequence Numbering" document

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Using Application Level Gateways with NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2 Feature Information for Using Application Level Gateways with NAT

Feature Name	Releases	Feature Configuration Information
MultiPart SDP Support for NAT	15.0(1)M	The MultiPart SDP Support for NAT feature adds support for multipart SDP in a SIP ALG for the Advanced NAT Portfolio. This feature is disabled by default. The following commands were modified by this feature: **debug ip nat**, **ip nat service**.
NAT H.245 Tunneling Support	12.3(11)T	The NAT H.245 Tunneling Support feature allows H.245 tunneling in H.323 Application Level Gateways (ALGs).
NAT SCCP Fragmentation Support	12.4(6)T 15.1(3)T	The NAT SCCP Fragmentation Support feature adds support for TCP segments for NAT skinny ALG. A fragmented payload that requires an IP or port translation will no longer be dropped. In Cisco IOS Release 15.1(3)T, the NAT Segmentation with Layer 4 Forwarding feature was introduced. The following command was modified by this feature: **debug ip nat**.
NAT Support for H.323 v2 RAS feature	12.2(2)T 15.0(1)S	Cisco IOS NAT supports all H. 225 and H.245 message types, including those sent in the RAS protocol.

Feature Name	Releases	Feature Configuration Information
NAT Support for H.323 v3 and v4 in v2 Compatibility Mode	12.3(2)T	The NAT Support for H.323 v3 and v4 in v2 Compatibility Mode feature enables Cisco NAT routers to support messages coded in H.323 v3 and v4 when those messages contain fields compatible with H.323 v2. This feature does not add support for H.323 capabilities introduced in v3 and v4, such as new message types or new fields that require address translation.
NAT Support for IPsec ESP-- Phase II	12.2(15)T	The NAT Support for IPsec ESP-- Phase II feature provides support for Internet Key Exchange (IKE) and ESP without encapsulation in tunnel mode through a Cisco IOS router configured with NAPT.
NAT Support for SIP	12.2(8)T	NAT Support for SIP adds the ability to configure Cisco IOS NAT between VoIP solutions based on SIP.
Support for applications that do not use H.323	12.2(33)XNC	NAT with an ALG will translate packets from applications that do not use H.323, as long as the applications use port 1720.
Support for IPsec ESP Through NAT	12.2(13)T	IPsec ESP through NAT provides the ability to support multiple concurrent IPsec Encapsulating Security Payload (ESP) tunnels or connections through a Cisco IOS Network Address Translation (NAT) device configured in Overload or Port Address Translation (PAT) mode.

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.