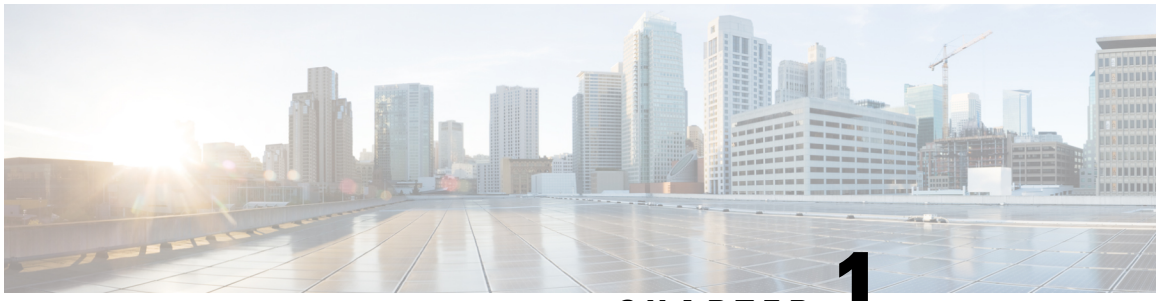
IP Addressing: NAT Configuration Guide, Cisco IOS Release 15SY

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



CHAPTER 1

Configuring NAT for IP Address Conservation

This module describes how to configure Network Address Translation (NAT) for IP address conservation and how to 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 device, usually connecting two networks. Before packets are forwarded onto another network, NAT translates the private (not globally unique) addresses in the internal network into legal addresses. NAT can be configured to advertise to the outside world only one address for the entire network. This ability provides more 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. It allows 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 caveats and feature information, see Bug Search Tool and 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.

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

All access lists that are required for use with the configuration tasks that are described in this module must be configured before initiating a configuration task. For information about how to configure an access list, see the *IP Access List EntrySequence Numbering* document.

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

NAT Requirements

Before configuring NAT in your network, you should know the interfaces on which NAT will be configured and for what purposes. The following requirements will help you decide how to configure and use NAT:

- Define the NAT inside and outside interfaces if:
  - Users exist off multiple interfaces.
  - Multiple interfaces connect to the Internet.
- 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.
  - Allow overlapping networks to communicate.
  - Allow networks with different address schemes to communicate.
  - Allow the use of an application level gateway.
  - Redirect TCP traffic to another TCP port or address.
  - Use NAT during a network transition.

Restrictions for Configuring NAT for IP Address Conservation

- It is not practical to use Network Address Translation (NAT) if a large number of hosts in the stub domain communicate outside of the domain.
- Some applications use embedded IP addresses in such a way that translation by a NAT device is impractical. 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 device 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 **permit ip any any** command that is commonly used in the access list.
- NAT configuration is not supported on the access side of the Intelligent Services Gateway (ISG).
- On Cisco Catalyst 6500 Series Switches, if you have a NAT overload configuration, we recommend that you limit the number of NAT translations to less than 64512, by using the **ip nat translation max-entries** command. If the number of NAT translations is 64512 or more, a limited number of ports are available for use by local applications, which, in turn can cause security issues such as denial-of-service (DoS) attacks. The port numbers used by local applications can easily be identified by DoS attacks, leading to security threats. This restriction is specific to all NAT overload configurations (for example, interface overload or pool overload configurations) that use a logical, loopback, or physical address for NAT configurations.
- Configuring zone-based policy firewall high availability with NAT and NAT high availability with zone-based policy firewalls is not recommended.
- If the NAT outside local address matches with any logical interface address, interface IP address, or a tunnel-configured address; then packets are software-switched.
- NAT outside interface is not supported on a VRF. However, NAT outside interface is supported in iWAN and is part of the Cisco Validated Design.
- The **acl-log** keyword will not work with an ACL used in NAT. The permit or deny functions for NAT ACL are used to filter the traffic according to the NAT rule. A rule like **permit tcp any any log** in the ACL used for NAT configuration is similar to **permit tcp any any**. Native ACL logging does not work in this ACL.
- BFD sessions may fail if you configure them to operate using an address that is also used for dynamic NAT. One common scenario is when you configure BFD on the same interface that you use to carry out interface-based dynamic NAT overload. To avoid this, you can instead employ a pool-based dynamic NAT overload configuration. However, even in this scenario, ensure that you do not use the chosen NAT pool address for BFD.

  When you configure BFD, we recommend you to use an address that does not overlap with NAT in order to avoid a conflict in case dynamic NAT is also configured on the device.

Information About Configuring NAT for IP Address Conservation

Benefits of Configuring NAT for IP Address Conservation

Network Address Translation (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 IP addresses, and if more than 254 clients are present or are planned, the scarcity of Class B addresses becomes a serious issue. 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 clients. With client addresses hidden, a degree of security is established. 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).

The expansion of Class A addresses occurs within the organization without a concern for addressing changes at the LAN or the Internet interface.

Cisco software can selectively or dynamically perform NAT. This flexibility allows 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, 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 to those few routers on which NAT will be configured.

Purpose of NAT

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.

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 NAT configuration without any static configuration. Full support for NetMeeting Directory (Internet Locator Service) is also provided through NAT.

How NAT Works

A device that is configured with NAT has at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit device between a stub domain and the backbone. When a packet exits the domain, NAT translates the locally significant source address into a globally unique address. When a packet enters 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 NAT cannot allocate an address because it has run out of addresses, it drops the packet. Then, NAT sends an Internet Control Message Protocol (ICMP) host unreachable packet to the destination.

Uses of NAT

NAT can be used for the following scenarios:

- Connect to the internet when all your hosts do not have globally unique IP addresses. Network Address Translation (NAT) enables private IP networks that use nonregistered IP addresses to connect to the Internet. NAT is configured on a device at the border of a stub domain (mentioned as the *inside network*) and a public network such as the Internet (mentioned as the *outside network*). NAT translates 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 simultaneously outside the domain. When outside communication is necessary, only a small subset of the IP addresses in the domain must be translated into globally unique IP addresses. Also, these addresses can be reused when they are no longer in use.
- Change your internal addresses. Instead of changing the internal addresses, which can be a considerable amount of work, you can translate them by using NAT.

- For basic load-sharing of TCP traffic. You can map a single global IP address with many local IP addresses by using the TCP Load Distribution feature.

NAT Inside and Outside Addresses

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

Similarly, the term *outside* refers to those networks to which the stub network connects, and which are not under the control of an organization. Also, hosts in outside networks can be subject to translation, and can thus have local and global addresses. NAT uses the following definitions:

- Inside local address—An IP address that is assigned to a host on the inside network. The address that the Network Information Center (NIC) or service provider assigns is probably not a legitimate IP address.
- 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 that is routable on the inside.
- Outside global address—The IP address that is 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.

NAT supports the following VRFs:

Table 1: VRF NAT Support

NAT Inside Interface	NAT Outside Interface	Condition
Global VRF (also referred to as a non-VRF interface)	Global VRF (also referred to as a non-VRF interface)	Normal
VRF X	Global VRF (also referred to as a non-VRF interface)	When NAT is not configured for Match-in-VRF support. For more details, see the *Match-in-VRF Support for NAT* chapter.
VRF X	VRF X	When both inside and outside interfaces are in the same VRF, and NAT is configured with Match-in-VRF support.

This section describes the following topics:

- Inside Source Address Translation, on page 6
- Overloading of Inside Global Addresses, on page 7

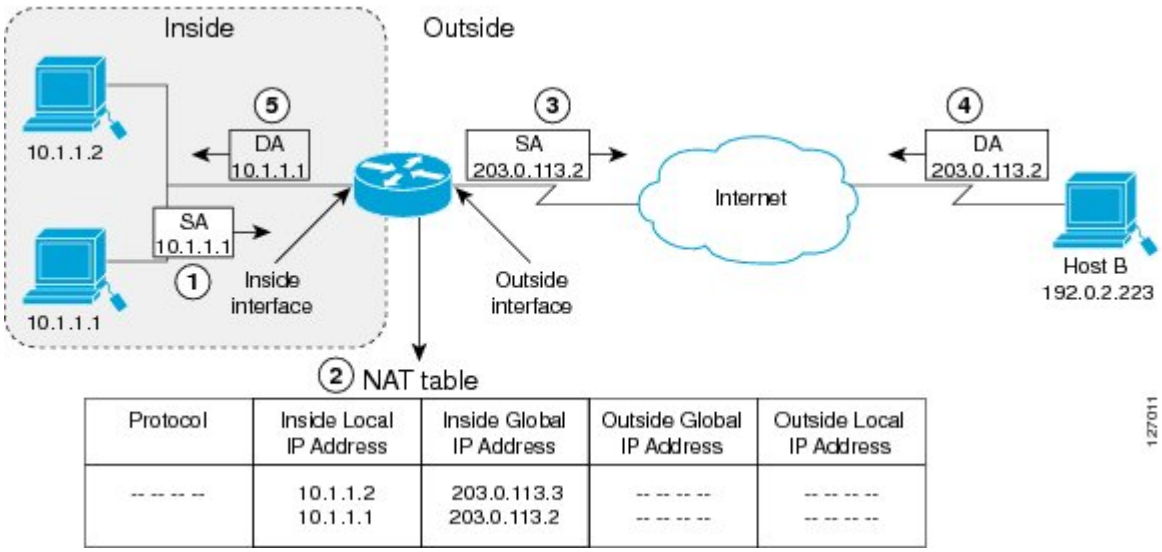
Inside Source Address Translation

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

- *Static translation* establishes a one-to-one mapping between the 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.

The following figure illustrates a device that is translating a source address inside a network to a source address outside the network.

Figure 1: NAT Inside Source Translation



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

1. The user at host 10.1.1.1 opens a connection to Host B in the outside network.
2. The first packet that the device receives from host 10.1.1.1 causes the device to check its Network Address Translation (NAT) table. Based on the NAT configuration, the following scenarios are possible:
   - If a static translation entry is configured, the device goes to Step 3.
   - If no translation entry exists, the device determines that the source address (SA) 10.1.1.1 must be translated dynamically. The device selects a legal, global address from the dynamic address pool, and creates a translation entry in the NAT table. This kind of translation entry is called a *simple entry*.
3. The device 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 device 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.

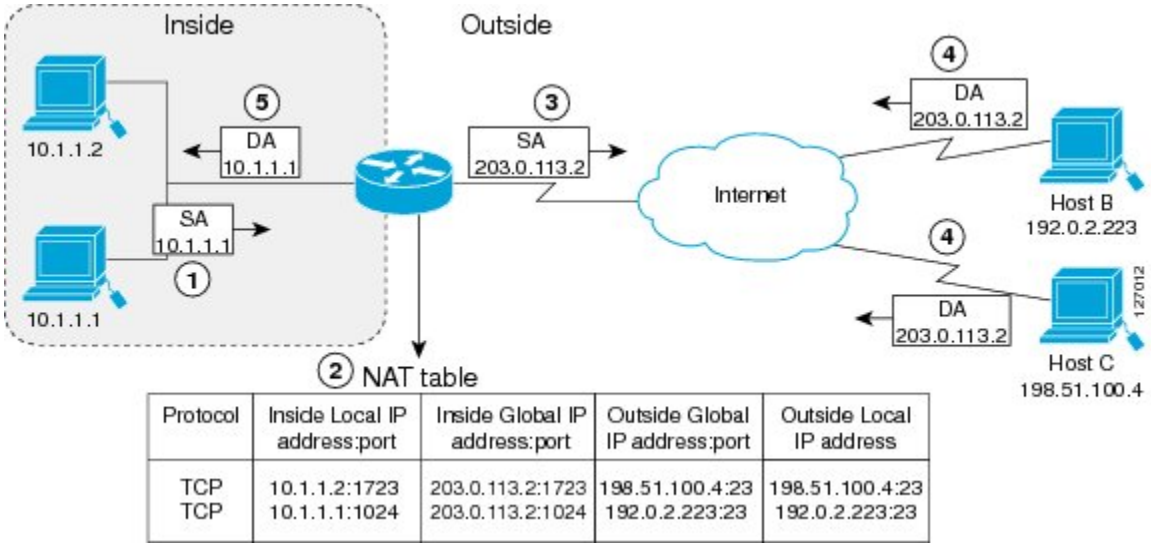
Host 10.1.1.1 receives the packet and continues the conversation. The device performs Steps 2 to 5 for each packet that it receives.

Overloading of Inside Global Addresses

You can conserve addresses in the inside global address pool by allowing a device to use one global address for many local addresses. This type of Network Address Translation (NAT) configuration is called overloading. When overloading is configured, the device maintains enough information from higher-level protocols (for example, TCP or UDP port numbers). This action translates 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 local addresses.

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

Figure 2: NAT Overloading Inside Global Addresses



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

The device performs the following process in the overloading of inside global addresses, as shown in the preceding figure. Both Host B and Host C believe that they are communicating with a single host at address 203.0.113.2. Whereas, they are actually communicating with different hosts; the port number is the differentiator. In fact, many inside hosts can 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 device receives from host 10.1.1.1 causes the device to check its NAT table. Based on your NAT configuration the following scenarios are possible:
   - If no translation entry exists, the device determines that IP address 10.1.1.1 must be translated, and translates inside local address 10.1.1.1 to a legal global address.
   - If overloading is enabled and another translation is active, the device reuses the global address from that translation and saves enough information. This saved information can be used to translate the global address back, as an entry in the NAT table. This type of translation entry is called an *extended entry*.
3. The device replaces 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 device receives the packet with the inside global IP address, it performs a NAT table lookup by using a protocol, the inside global address and port, and the outside address and port as keys. It 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 device performs Steps 2 to 5 for each packet it receives.

Types of NAT

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

The types of NAT include:

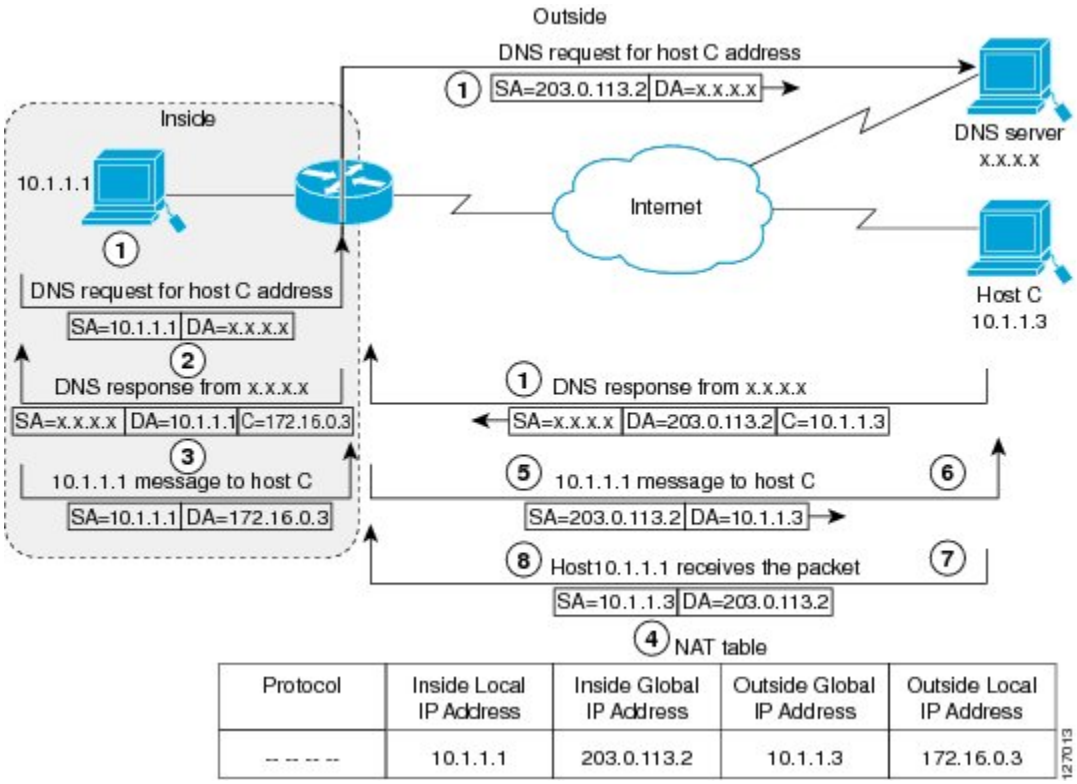
- Static address translation (static NAT)—Allows one-to-one mapping between local and global addresses.
- Dynamic address translation (dynamic NAT)—Maps unregistered IP addresses to registered IP addresses from a pool of registered IP addresses.
- Overloading—Maps multiple unregistered IP addresses to a single registered IP address (many to one) by using different ports. This method is also known as Port Address Translation (PAT). Thousands of users can be connected to the Internet by using only one real global IP address through overloading.

Address Translation of Overlapping Networks

Use Network Address Translation (NAT) to translate IP addresses if the IP addresses that you use are not legal or officially assigned. Overlapping networks result when you assign an IP address to a device on your network. This device is already legally owned and assigned to a different device on the Internet or outside the network.

The following figure shows how NAT translates overlapping networks.

Figure 3: NAT Translating Overlapping Addresses



The following steps describe how a device translates overlapping addresses:

1. Host 10.1.1.1 opens a connection to Host C using a name, requesting a name-to-address lookup from a Domain Name System (DNS) server.
2. The device 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 device creates a simple translation entry. This entry maps the overlapping address, 10.1.1.3 to an address from a separately configured, outside the local address pool.

The device examines every DNS reply to ensure that the IP address is not in a stub network. If it is, the device translates the address as described in the following steps:

1. Host 10.1.1.1 opens a connection to 172.16.0.3.
2. The device sets up the translation mapping of the inside local and global addresses to each other. It also sets up the translation mapping of the outside global and local addresses to each other.
3. The device 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 device 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.

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 lists, next-hop IP addresses, and output interfaces 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. 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 the dynamic NAT route map name.

Benefits of using route maps for address translation are as follows:

- 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.

NAT Route Maps Outside-to-Inside Support Feature

The NAT Route Maps Outside-to-Inside Support feature enables the deployment of a NAT route map configuration that allows IP sessions to be initiated from the outside to the inside.

An initial session from inside to outside is required to trigger a NAT. New translation sessions can then be initiated from the 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. The outside-to-inside functionality remains unchanged (by not creating additional entries to allow the return traffic for a route-map-based dynamic entries) unless you configure the **ip nat inside source reversible** command.

The following restrictions apply to the NAT Route Maps Outside-to-Inside Support feature:

- 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.
- 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.
- The **match interface** and **match ip next-hop** commands are not supported for reversible route maps.

Static IP Address Support

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

To support users who are configured with a static IP address, the NAT Static IP Address Support feature extends the capabilities of public wireless LAN providers. By configuring a device to support users with a static IP address, public wireless LAN providers extend their services to a greater number of users.

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.

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 UDP. Generally, the RADIUS protocol is considered a connectionless service. RADIUS-enabled devices handle issues that are related to a server availability, retransmission, and timeouts rather than the transmission 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. To deliver service to the user, RADIUS servers receive a user connection request, authenticate the user, and then return the configuration information necessary for the client. 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 misuse of standard protocols or connection processes. The intent of DoS attack is to overload and disable a target, such as a device or web server. DoS attacks can come from a malicious user or from a computer that is infected with a virus or worm. Distributed DoS attack is an attack that comes from many different sources at once. This attack can be when a virus or worm has infected many computers. Such distributed DoS attacks can spread rapidly and involve thousands of systems.

Viruses and Worms That Target NAT

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

How to Configure NAT for IP Address Conservation

The tasks that are described in this section configure NAT for IP address conservation. Ensure that you configure at least one of the tasks that are described in this section. Based on your configuration, you may need to configure more than one task.

Configuring Inside Source Addresses

Inside source addresses, can be configured for static or dynamic translations. Based on your requirements, you can configure either static or dynamic translations.

Configuring Static Translation of Inside Source Addresses

Configure static translation of the inside source addresses to allow one-to-one mapping between an 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.

Note Configure different IP addresses for an interface on which NAT is configured and for inside addresses that are configured by using the **ip nat inside source static** command.

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* [**secondary**]
10. **ip nat outside**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(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:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:**	Sets a primary IP address for an interface.

	Command or Action	Purpose
	`Device(config-if)# ip address 10.114.11.39 255.255.255.0`	
Step 6	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies a different interface and enters the interface configuration mode.
Step 9	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(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:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 11	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode. **Note** Conditional translation is not supported with **ip nat outside source route-map** configuration.

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 must access the Internet. The dynamically configured pool IP address may be used as needed. It is released for use by other users when access to the Internet is no longer required.

Note When inside global or outside local addresses belong to a directly connected subnet on a NAT device, the device adds IP aliases for them. This action enables it to answer Address Resolution Protocol (ARP) requests. However, a situation can arise where the device answers packets that are not destined for it, possibly causing a security issue. This security issue can happen when an incoming Internet Control Message Protocol (ICMP) packet or a UDP packet that is destined for one of the aliased addresses does not have a corresponding NAT translation in the NAT table. Also, the device itself runs a corresponding service, for example, Network Time Protocol (NTP). Such a situation can cause minor security risks.

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(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.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(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:** `Device(config)# ip nat inside source list 1 pool net-208`	Establishes dynamic source translation, specifying the access list defined in Step 4.
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters an interface configuration mode.

	Command or Action	Purpose
Step 7	**ip address** *ip-address mask* **Example:** `Device(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:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters an interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(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:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Using NAT to Allow 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(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:** `Device(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:** `Device(config)# ip nat inside source list 1 pool net-208 overload`	Establishes dynamic source translation with overloading, specifying the access list defined in Step 4.
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.201.1 255.255.255.240`	Sets a primary IP address for the interface.

	Command or Action	Purpose
Step 8	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(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:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Address Translation Timeouts

You can configure address translation timeouts that is based on your NAT configuration.

By default, dynamic address translations time out after a period of remaining idle. You can change the default values on timeouts, if necessary. When overloading is not configured, simple translation entries time out after 24 hours. Use the **ip nat translation timeout** command to change the timeout value for dynamic address translations.

You can use the **ip nat translation max-entries** command to change the default global NAT translation limit.

Note On Catalyst 6500 Series Switches, when the NAT translation is done in the hardware, timers are reset every 100 seconds or once the set timeout value is reached.

Changing the Translation Timeout

By default, dynamic address translations time out after some period of remaining idle. 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* command 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 can control the translation entry timeout, because each translation entry contains more context about the traffic using it.

Based on your configuration, you can change the timeouts that are described in this section. If you must quickly free your global IP address for a dynamic configuration, configure a shorter timeout than the default timeout. You can do it by using the **ip nat translation timeout** command. However, the configured timeout is longer than the other timeouts configured using commands specified in the following task. If a finish (FIN) packet does not close a TCP session properly from both sides or during a reset, change the default TCP timeout. You can do it by using the **ip nat translation tcp-timeout** command.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat translation** *seconds* **Example:** `Device(config)# ip nat translation 300`	(Optional) Changes the amount of time after which NAT translations time out. • The default timeout is 24 hours, and it applies to the aging time for half-entries.
Step 4	**ip nat translation udp-timeout** *seconds* **Example:** `Device(config)# ip nat translation udp-timeout 300`	(Optional) Changes the UDP timeout value.
Step 5	**ip nat translation dns-timeout** *seconds* **Example:**	(Optional) Changes the Domain Name System (DNS) timeout value.

	Command or Action	Purpose
	`Device(config)# ip nat translation dns-timeout 45`	
Step 6	**ip nat translation tcp-timeout** *seconds* **Example:** `Device(config)# ip nat translation tcp-timeout 2500`	(Optional) Changes the TCP timeout value. • The default is 24 hours.
Step 7	**ip nat translation finrst-timeout** *seconds* **Example:** `Device(config)# ip nat translation finrst-timeout 45`	(Optional) Changes the finish and reset timeout value. • **finrst-timeout**—The aging time after a TCP session receives both finish-in (FIN-IN) and finish-out (FIN-OUT) requests or after the reset of a TCP session.
Step 8	**ip nat translation icmp-timeout** *seconds* **Example:** `Device(config)# ip nat translation icmp-timeout 45`	(Optional) Changes the ICMP timeout value.
Step 9	**ip nat translation syn-timeout** *seconds* **Example:** `Device(config)# ip nat translation syn-timeout 45`	(Optional) Changes the synchronous (SYN) timeout value. • The synchronous timeout or the aging time is used only when a SYN request is received on a TCP session. When a synchronous acknowledgment (SYNACK) request is received, the timeout changes to TCP timeout.
Step 10	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Allowing Overlapping Networks to Communicate Using NAT

Tasks in this section are grouped because they perform the same action. However, the tasks 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 you have implemented.

This section contains the following tasks:

- Configuring Static Translation of Overlapping Networks
- Configuring Dynamic Translation of Overlapping Networks
- What to Do Next

Configuring Static Translation of Overlapping Networks

Configure static translation of overlapping networks that are based on the following requirements:

- If your IP addresses in the stub network are legitimate IP addresses belonging to another network.
- If you want to communicate with those hosts or routers by 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(config)# ip nat inside source static 192.168.121.33 10.2.2.1`	Establishes static translation between an inside local address and an inside global address.
Step 4	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Device(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:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 10	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 11	**end** **Example:** `Device(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.
- You want to communicate with those hosts or routers by 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(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:** `Device(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:** `Device(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:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(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:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(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:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 13	**end** **Example:** `Device(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Server TCP Load Balancing

Perform this task to configure a server TCP load balancing by way of destination address rotary translation. The commands that are specified in the task allow you to map one virtual host with many real hosts. Each new TCP session opened with the virtual host is 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# 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:** `Device(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:** `Device(config)# access-list 1 permit 192.168.201.30 0.0.0.255`	Defines an access list permitting the address of the virtual host.
Step 5	**ip nat inside destination-list** *access-list-number* **pool** *name* **Example:** `Device(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:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(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:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 10	**interface** *type number* **Example:** `Device(config)# interface serial 0`	Specifies a different interface and enters the interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.15.129 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 13	**end** **Example:** `Device(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Enabling Route Maps on Inside Interfaces

Before you begin

All route maps required for use with this task must 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# 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:** `Device(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:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip nat translations** [**verbose**] **Example:** `Device# 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 you to configure a Network Address Translation (NAT) route map configuration. It allows 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.

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:**	Defines a pool of network addresses for NAT.

	Command or Action	Purpose
	`Device(config)# ip nat pool POOL-A 192.168.201.4 192.168.201.6 netmask 255.255.255.128`	
Step 4	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Device(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:** `Device(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.
Step 6	**ip nat inside source route-map** *name* **pool** *name* [**reversible**] **Example:** `Device(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:** `Device(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 device that is 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.

Note When you configure the **ip nat outside source static** command to add static routes for outside local addresses, there is a delay in the translation of packets and packets are dropped. Packets are dropped because a shortcut is not created for the initial synchronization (SYN) packet when NAT is configured for static translation. To avoid dropped packets, configure either the **ip nat outside source static add-route** command or the **ip route** command.

Benefits of configuring NAT of external IP addresses only are:

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

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** | **udp**} *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* | **static** *local-ip global-ip* [**no-payload**]}
7. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** {**tcp** | **udp**} *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* | **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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# 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:** `Device(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 device.
Step 4	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]} **Example:** `Device(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 device.
Step 5	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static**	Disables packet translation on the inside host device.

	Command or Action	Purpose
	[**network**] *local-network-mask global-network-mask* [**no-payload**]} **Example:** `Device(config)# ip nat inside source static 10.1.1.1 192.168.1.1 no-payload`	
Step 6	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** *local-ip global-ip* [**no-payload**]} **Example:** `Device(config)# ip nat outside source static 10.1.1.1 192.168.1.1 no-payload`	Disables packet translation on the outside host device.
Step 7	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]} **Example:** `Device(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 device.
Step 8	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]} **Example:** `Device(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 device.
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 10	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	Displays active NAT.

Configuring the NAT Default Inside Server Feature

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 are redirected, and packets are not dropped.

Dynamic mapping and interface overload can be configured for gaming devices. For online games, outside traffic comes on a different UDP port. If a packet is destined for an interface from outside an enterprise's network, and there is no match in the NAT table for fully extended entry or static port entry, the packet is forwarded to the gaming device using a simple static entry.

Note

- You can use this feature to configure gaming devices with an IP address different from the IP address of the PC. To avoid unwanted traffic or DoS attacks, use access lists.
- For traffic going from the PC to the outside, 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip* **interface** *type number* **Example:** `Device(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:** `Device(config)# ip nat inside source static tcp 10.1.1.1 23 interface 23`	(Optional) Enables the use of telnet to the device from the outside.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip nat translations** [**verbose**] **Example:** `Device# 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 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 reenable RTSP on a NAT router if this configuration has been disabled.

Configuring Support for Users with Static IP Addresses

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

Before you begin

Before configuring support for users with static IP addresses, you must first enable NAT on your router and configure a RADIUS server host.

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Configures an interface and enters an interface configuration mode.
Step 4	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**ip nat allow-static-host** **Example:** `Device(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:** `Device(config)# ip nat pool pool1 172.16.0.0 172.16.0.254 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:** `Device(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.
Step 9	**access-list** *access-list-number* **deny ip** *source* **Example:** `Device(config)# access-list 1 deny ip 192.168.196.51`	Removes the traffic of the device from NAT. • The *source* argument is the IP address of the device that supports the NAT Static IP Support feature.
Step 10	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 11	**show ip nat translations verbose** **Example:** `Device# 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:

```
Device# 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 NAT entry of the static IP client times out, the NAT entry and the secure ARP entry associations are deleted for the client. 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# 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:** `Device(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:** `Device(config)# ip nat translation arp-ping-timeout 600`	Changes the amount of time after each network address translation.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring the Rate Limiting NAT Translation Feature

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:** `Device enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip nat translations** **Example:** `Device# 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:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip nat translation max-entries** {*number* | **all-vrf** *number* | **host** *ip-address number* | **list** *listname number* | **vrf** *name number*} **Example:** `Device(config)# ip nat translation max-entries 300`	Configures the maximum number of NAT entries that are 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

	Command or Action	Purpose
		NAT entries for the named VRF instance that is greater than or less than that allowed for all VRF instances.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip nat statistics** **Example:** `Device# 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 the current NAT rate limit settings. **Note** The CEF counters associated with the output of the **show ip nat statistics** command signify the number of packets that are translated and forwarded in the SW plane. Packets that require translation are punted to the SW plane in the absence of the corresponding NF shortcuts in the HW plane. This enables SW plane to carry out the translation and program the corresponding NF shortcuts in the HW in order to facilitate the HW translation for subsequent packets that match the given flow. A route-map based NAT rule does not maintain Half Entry mappings and this implies that every new packet flow that matches the given rule is directed to the SW plane for translation and forwarding. Such packets undergo translation in the SW plane. This in turn results in the increment of the afore mentioned CEF counters. This is an expected behavior when you employ a route-map-based NAT configuration. However, note that these packets that undergo translation in the SW result in the corresponding full flow NF shortcuts to be programmed in the HW. This is to facilitate the HW translation of subsequent packets that match the given flow.

Configuration Examples for Configuring NAT for IP Address Conservation

Example: Configuring Static Translation of Inside Source Addresses

The following example shows how inside hosts addressed from the 10.114.11.0 network are translated to the globally unique 172.31.233.208/28 network. Further, packets from outside hosts that are addressed from the 10.114.11.0 network (the true 10.114.11.0 network) are translated to appear 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 gigabitethernet 0/0/0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
interface gigabitethernet 1/1/1
 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) device 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 192.168.121.33.10.2.2.2 vrf vrf2
```

Example: Configuring Dynamic Translation of Inside Source Addresses

The following example shows how inside hosts addressed from either the 192.168.1.0 or the 192.168.2.0 network are translated 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 gigabitethernet 0/0/0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
interface gigabitethernet 1/1/1
 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 shows how only traffic local to the provider edge (PE) device running NAT is translated:

```
ip nat inside source list 1 interface gigabitethernet 0/0/0 vrf vrf1 overload
ip nat inside source list 1 interface gigabitethernet 0/0/0 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 gigabitethernet 1/1/1 vrf vrf1 overload
ip nat inside source list 1 interface gigabitethernet 1/1/1 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: Using NAT to Allow Internal Users Access to the Internet

The following example shows how to create a pool of addresses that is named net-208. The pool contains addresses from 172.31.233.208 to 172.31.233.233. Access list 1 allows packets with SA from 192.168.1.0 to 192.168.1.255. If no translation exists, packets matching access list 1 is 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
access-list 1 permit 192.168.1.0 0.0.0.255
ip nat inside source list 1 pool net-208 overload
interface gigabitethernet 1/1/1
 ip address 192.168.201.1 255.255.255.240
 ip nat inside
!
interface gigabitethernet 0/0/0
 ip address 192.168.201.29 255.255.255.240
 ip nat outside
!
```

Example: Configuring Server TCP 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 addresses of real hosts. The access list defines the virtual address. If a translation does not 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
access-list 2 permit 192.168.15.1
ip nat inside destination list 2 pool real-hosts
interface gigabitethernet 0/0/0
 ip address 192.168.15.129 255.255.255.240
 ip nat inside
!
interface serial 0
 ip address 192.168.15.17 255.255.255.240
```

```
 ip nat outside
!
```

Example: Enabling Route Maps on Inside Interfaces

```
ip nat inside source static 192.168.201.6 192.168.201.21
!
```

Example: Enabling NAT Route Maps Outside-to-Inside Support

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

```
ip nat pool POOL-A 192.168.201.4 192.168.201.6 netmask 255.255.255.128
ip nat pool POOL-B 192.168.201.7 192.168.201.9 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
```

Example: Configuring NAT of External IP Addresses Only

```
ip nat inside source static network 10.1.1.1 192.168.2510/24 no-payload
ip nat inside source static tcp 10.1.1.1 2000 192.168.1.1 2000 no-payload
ip nat inside source static 10.1.1.1 192.168.1.1 no-payload
ip nat outside source static 10.1.1. 192.168.1.1 no-payload
ip nat outside source static tcp 10.1.1.1 20000 192.168.1.1 20000 no-payload
ip nat outside source static network 10.1.1.1 192.168.251.0/24 no-payload
```

Example: Configuring Support for Users with Static IP Addresses

```
interface gigabitethernet 1/1/1
 ip nat inside
!
ip nat allow-static-host
ip nat pool pool1 172.16.0.0 172.16.0.254 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: Configuring NAT Static IP Support

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

```
interface gigabitethernet 1/1/1
 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 gigabitethernet3/0
radius-server host 172.31.88.1 auth-port 1645 acct-port 1646
radius-server key cisco
```

Example: Configuring the Rate Limiting NAT Translation Feature

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

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

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

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

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, “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
```

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
```

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
```

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, “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

Related Topic	Document Title
Using HSRP and stateful NAT for high availability	*Configuring NAT for High Availability* module
Using NAT with MPLS VPNs	*Integrating NAT with MPLS VPNs* module

Standards and RFCs

Standard/RFC	Title
RFC 1597	Internet Assigned Numbers Authority
RFC 1631	The IP Network Address Translation (NAT)
RFC 1918	Address Allocation for Private Internets
RFC 2663	IP Network Address Translation (NAT) Terminology and Considerations
RFC 3022	Traditional IP Network Address Translation (Traditional NAT)

MIBs

MIB	MIBs Link
CISCO-NAT-STAT-MIB	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 Configuring NAT for IP Address Conservation

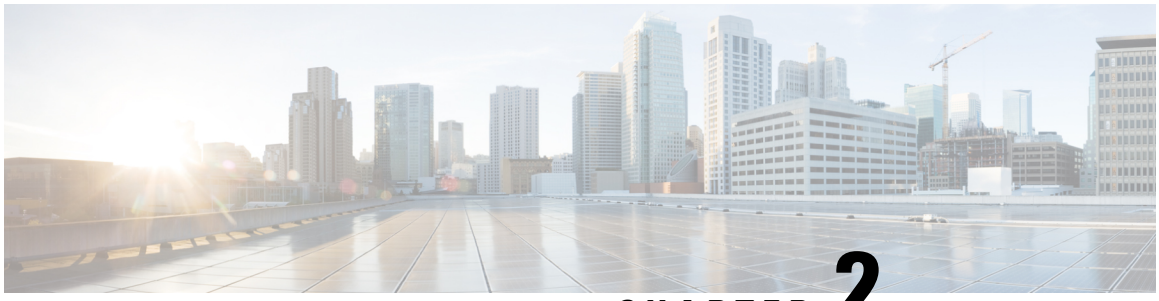
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 2: Feature Information for Configuring NAT for IP Address Conservation

Feature Name	Releases	Feature 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 the 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 enables forwarding of packets from 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 allows IP sessions to be initiated from the outside to the inside.
NAT RTSP Support Using NBAR	12.3(7)T	The NAT RTSP Support Using NBAR feature is a client/server multimedia presentation control protocol that supports multimedia application delivery. Applications that use RTSP include 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.
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	Use the NAT Translation of External IP Addresses Only feature to configure NAT to ignore all embedded IP addresses for any application and traffic type.

Feature Name	Releases	Feature Information
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 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.
Support for ARP Ping in a Public Wireless LAN	12.4(6)T	The Support for ARP Ping in a Public Wireless LAN feature ensures that the NAT entry and the secure ARP entry from removal when the static IP client exists in the network, where the IP address is unchanged after authentication.



CHAPTER 2

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 ALGs for IP header translation.

NAT performs translation services on any TCP/UDP traffic that does not carry source and destination IP addresses in the application data stream. Protocols that do not carry the source and destination IP addresses 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 embed the IP address information within the payload require the support of an ALG. NAT requires a variety of ALGs to handle application data stream (Layer 7) protocol-specific services such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection/session information from control channels.

NAT supports virtual routing and forwarding (VRF) for protocols that have a supported ALG.

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 NAT device configured in Overload or Port Address Translation (PAT) mode. You can use the **ip nat service dns-v6** command to control processing of IPv6 DNS packets by ALG

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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.

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

- Network Address Translation (NAT) translates only embedded IPv4 addresses.
- Protocols that require application-level gateway (ALG) processing are not compatible with load balancing. All process-switched packets use the per-packet load-balancing algorithm. Process switching uses the per-packet load-balancing algorithm across equal-cost paths. As a result, every odd or even process-switched packet may be dropped by ISPs in a dual-ISP scenario due to a failed Unicast Reverse Path Forwarding (uRPF) check because these packets have the same source IP address (which is allocated by NAT or Port Address Translation [PAT]), but are routed to different outside interfaces. The packet drop causes excessive delay and retransmission of packets.
- In Cisco IOS Release 12.4 Mainline, the NAT ALG for Session Initiation Protocol (SIP) does not support the following T.38 session attributes in the Session Description Protocol (SDP): sqn, cdsc, and cpar. These session attributes are removed from the SDP header by the NAT ALG, which causes the SIP-based T.38 calls to fail. This restriction is applicable only to the Cisco IOS Release 12.4 mainline. As a workaround, upgrade to Cisco IOS Release 12.4(1)T and later releases.

Information About Using Application-Level Gateways with NAT

Benefits of Configuring NAT IPsec

- NAT enables customers to deploy private IP addresses within their network and translate the private IP addresses to public IP addresses when connecting to the Internet or when interconnecting with another corporate network.
- NAT support for the Session Initiation Protocol (SIP) adds the ability to deploy NAT on VoIP solutions based on SIP.
- With NAT ALGs, customers can control their IP address scheme and include complete support for H.323 v2 gatekeeper designs.

- ESP entries in the translation table are normally 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 SPI entries are matched. Some third-party concentrators require both source ports and incoming ports to use port 500. Use the **ip nat service preserve-port** command to preserve the ports rather than changing them, which is required with regular NAT.

IPsec

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 Encapsulating Security Payload (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. You can enable IPsec packet processing using ESP with the **ip nat service ipsec-esp enable** command.

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.

We recommend that TCP and UDP be used when conducting IPsec sessions that traverse a NAPT device. However, not all VPN servers or clients support TCP or UDP.

SPI Matching

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..

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 packet networks. NAT supports four versions of the H.323 protocols: Version 1, Version 2, Version 3, and Version 4. The NAT Support for H.323 v3 and v4 in v2 Compatibility Mode feature enables NAT routers to support messages coded in H.323 Version 3 and Version 4 when these messages contain fields that are compatible with H.323 Version 2. This feature does not support H.323 capabilities introduced in H.323 Version 3 and Version 4, such as new message types or new fields that require address translation.

NAT H.245 Tunneling Support

The NAT H.245 Tunneling Support feature supports H.245 tunneling in H.323 ALGs. The H.245 tunneling supports H.245 tunnel messages that are needed to create a media channel setup.

For an H.323 call to take place, an H.225 connection on TCP port 1720 must be opened. When the H.225 connection is opened, the H.245 session is initiated and established. The H.323 connection can take place on a separate channel other than the H.225 or it can be done by using H.245 tunneling on the same H.225 channel whereby the H.245 messages are embedded in H.225 messages and sent on the previously established H.225 channel.

If the H.245 tunneled message is not understood by NAT, the media address or the port number is left untranslated by NAT, resulting in media traffic failure. The H.245 FastConnect procedures will not help if the H.245 tunneled message is not understood by NAT 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 Client Control Protocol (SCCP) messages, also called Skinny control messages, are exchanged over TCP. If either the IP phone or the Cisco Unified CallManager is configured to have a TCP maximum segment size (MSS) lower than the Skinny control message payload, the Skinny control message is segmented across multiple TCP segments. Prior to the introduction of this feature, Skinny control message exchanges used to fail during TCP segmentation because the NAT Skinny ALG was not able to reassemble Skinny control messages. The NAT SCCP Fragmentation Support feature adds support for TCP segments for the NAT Skinny ALG and fragmented payloads that requires an IP translation or a port translation is no longer dropped.

Skinny control messages can also be IP fragmented by 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, Skinny Client Control Protocol (SCCP), and the TCP Domain Name System (DNS) protocol. 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 setting sequence numbers in order, acknowledging the numbers in a packet, resegmenting the translated packet if it is larger than the maximum segment size (MSS), and handling retransmissions in case of packet loss. Layer 4 forwarding also handles out-of-order packets and these packets are buffered and not dropped. Layer 4 forwarding buffers received packets and notifies the NAT ALG when an in-order packet is available, sends acknowledgments to end hosts for received packets, and sends translated packets that it receives from the NAT ALG back into the output packet path.

Restrictions

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

- Firewalls are configured using the **ip inspect name** command. (Context-Based Access Control (CBAC) firewalls are not supported. 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 Unified CallManager are configured on the same device. In this case, a colocated solution in Call Manager Express is used.
- NAT Virtual Interface (NVI) is configured.
- Stateful Network Address Translation (SNAT) is enabled.

Note Effective January 31, 2014, Stateful NAT is not available in Cisco IOS software. For more information, see End-of-Sale and End-of-Life Announcement for the Cisco IOS Stateful Failover of Network Address Translation (SNAT).

- 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

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:**	Enables privileged EXEC mode. • Enter your password if prompted.

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

	Command or Action	Purpose
	`Router> enable`	
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.

Before you begin

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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• 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 Endpoints

Before you begin

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

Note Security parameter index (SPI) matching must be configured on the Network Address Translation (NAT) device and on both endpoint devices.

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
	`Device(config)# crypto ipsec nat-transparency spi-matching`	
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Enabling MultiPart SDP Support for NAT

The MultiPart SDP Support for NAT feature provides support for the multipart Session Description Protocol (SDP) in a SIP ALG. MultiPart SDP support for NAT is disabled by default.

Note NAT translates only embedded IPv4 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat service allow-multipart** **Example:** `Device(config)# ip nat service allow-multipart`	Enables multipart SDP.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 5	**show ip nat translations** **Example:** `Device# 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: Specifying a Port for NAT Translation

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

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: Enabling SPI Matching on Endpoints

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

Example: Enabling MultiPart SDP Support for NAT

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

Example: Specifying a Port for NAT Translation

```
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 Command 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*
NAT IP address conservation	*Configuring NAT for IP Address Conservation*

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

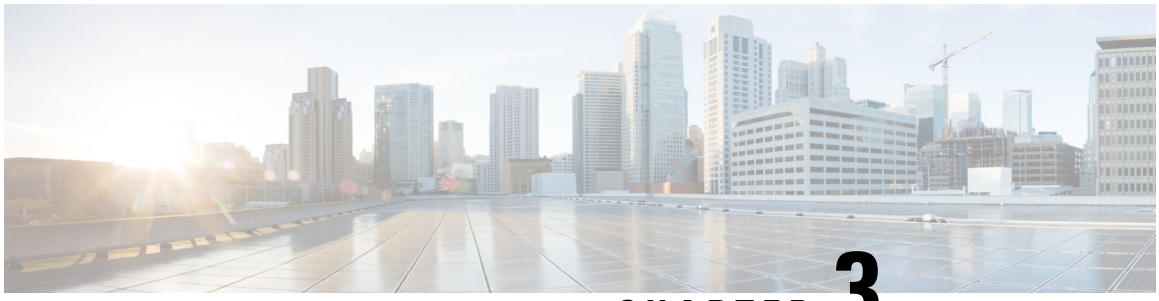
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 3: 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. This feature is disabled by default. The following commands were modified by this feature: **debug ip nat** and **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 Support for H.323 v2 RAS feature	12.2(2)T 15.0(1)S	NAT supports all H.225 and H.245 message types, including those sent in the RAS protocol.
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 NAT routers to support messages coded in H.323 Version 3 and Version 4 when these messages contain fields that are compatible with H.323 Version 2. This feature does not add support for H.323 capabilities introduced in H.323 Version 3 and Version 4, such as new message types or new fields that require address translation.

Feature Name	Releases	Feature Configuration Information
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 router configured with NAPT.
NAT Support of SCCP Fragmentation	12.4(6)T 15.1(3)T	The NAT Support of SCCP Fragmentation feature adds support for TCP segments for the NAT Skinny ALG. A fragmented payload that requires an IP translation or a port translation is no longer be dropped. The following command was modified by this feature: **debug ip nat**.
NAT Support for SIP	12.2(8)T	NAT Support for SIP adds the ability to configure NAT on 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 these applications use port 1720.
Support for IPsec ESP Through NAT	12.2(13)T	The IPsec ESP Through NAT feature provides the ability to support multiple concurrent IPsec Encapsulating Security Payload (ESP) tunnels or connections through a NAT device configured in Overload or Port Address Translation (PAT) mode.



CHAPTER 3

Integrating NAT with MPLS VPNs

Network Address Translation (NAT) Integration with MPLS VPNs feature allows multiple Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) to be configured on a single device to work together. NAT can differentiate which MPLS VPN it receives IP traffic from even if the MPLS VPNs are all using the same IP addressing scheme. This enhancement enables multiple MPLS VPN customers to share services while ensuring that each MPLS VPN is completely separate from the other.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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.

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 Integrating NAT with MPLS VPNs

- 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 at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s14/fsaclseq.htm

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

Restrictions for Integrating NAT with MPLS VPNs

Inside VPN to VPN with NAT is not supported.

Information About Integrating NAT with MPLS VPNs

Benefits of NAT Integration with MPLS VPNs

MPLS service providers would like to provide value-added services such as Internet connectivity, domain name servers (DNS), and voice over IP (VoIP) service to their customers. The providers require that their customers; IP addresses be different when reaching the services. Because MPLS VPN allows customers to use overlapped IP addresses in their networks, NAT must be implemented to make the services possible.

Implementation Options for Integrating Nat with MPLS VPNs

There are two approaches to implementing NAT in the MPLS VPN network. NAT can be implemented on the customer edge (CE) router, which is already supported by NAT, or it can be implemented on a provider edge (PE) router. The NAT Integration with MPLS VPNs feature enables the implementation of NAT on a PE router in an MPLS cloud.

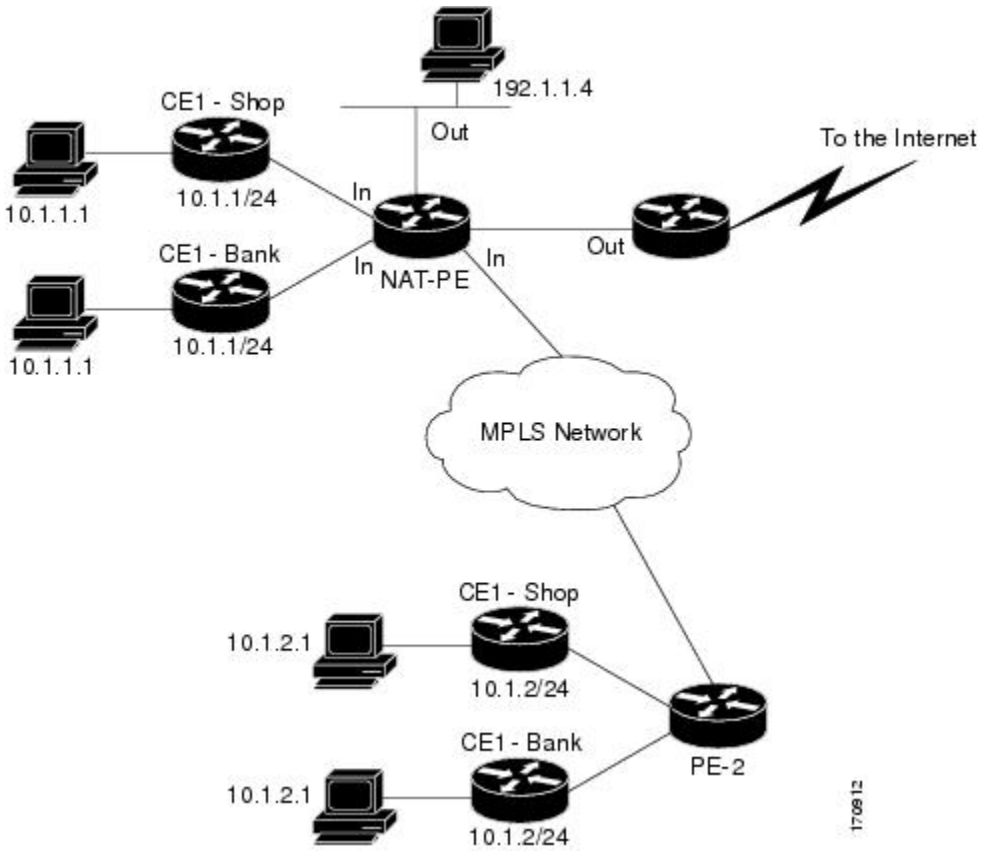
Scenarios for Implementing NAT on the PE Router

NAT could be implemented on the PE router in the following scenarios:

- Service point--Shared access can be from a generic interface or from a VPN interface.
- NAT point--NAT can be configured on the PE router that is directly connected to the shared access gateway, or on the PE router that is not directly connected to the shared access gateway.
- NAT interface--The shared access gateway interface most often is configured as the outside interface of NAT. The inside interface of NAT can be either the PE-CE interface of a VPN, the interface to the MPLS backbone, or both. The shared access gateway interface can also be configured as the inside interface.
- Routing type--Common service can be Internet connectivity or a common server. For Internet connectivity, a default route should be propagated to all the VPN customers that use the service. For common server access, a static or dynamically learned route should be propagated to the VPN customers.
- NAT configuration--NAT can have different configurations: static, dynamic, pool/interface overloading, and route-map.

The figure below shows a typical NAT integration with MPLS VPNs. The PE router connected to the internet and centralized mail service is employed to do the address translation.

Figure 4: Typical NAT Integration with MPLS VPNs



How to Integrate NAT with MPLS VPNs

Perform one or more of the following tasks depending on the type of translation you wish to configure for your network:

Configuring Inside Dynamic NAT with MPLS VPNs

Perform this task to configure your NAT PE router for dynamic translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
4. **ip nat** [**inside** | **outside**] **source** [**list** {*access-list-number* | *access-list-name*} | **route-map** *name*] [**interface** *type number* | **pool** *pool-name*] **vrf** *vrf-name*[**overload**]
5. Repeat Step 4 for each VPN being configured

6. **ip route vrf** *vrf-name prefix mask interface-type interface-number next-hop-address*
7. Repeat Step 6 for each VPN being configured.
8. **exit**
9. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as 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* **Example:** `Router(config)# ip nat pool inside 2.2.2.10 2.2.2.10 netmask 255.255.255.0`	Defines a pool of IP addresses for NAT.
Step 4	**ip nat** [**inside** | **outside**] **source** [**list** {*access-list-number* | *access-list-name*} | **route-map** *name*] [**interface** *type number* | **pool** *pool-name*] **vrf** *vrf-name*[**overload**] **Example:** `Router(config)# ip nat inside source list 1 pool mypool vrf shop overload`	Allows NAT to be configured on a particular VPN.
Step 5	Repeat Step 4 for each VPN being configured	--
Step 6	**ip route vrf** *vrf-name prefix mask interface-type interface-number next-hop-address* **Example:** `Router(config)# ip route vrf shop 0.0.0.0 0.0.0.0 ethernet 0 168.58.88.2`	Allows NAT to be configured on a particular VPN.
Step 7	Repeat Step 6 for each VPN being configured.	--
Step 8	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by virtual routing/forwarding (VRF) table translations.

Configuring Inside Static NAT with MPLS VPNs

Perform this task to configure your NAT PE router for static translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**static** {**esp** *local-ip* **interface** *type number* | *local-ip global-ip*}} [**extendable** | **mapping-id** *map-id*| **no-alias** | **no-payload** | **redundancy** *group-name* | **route-map** | **vrf** *name*]
4. Repeat Step 3 for each VPN being configured.
5. **ip route vrf** *vrf-name* **prefix** *prefix mask next-hop-address* **global**
6. Repeat Step 5 for each VPN being configured.
7. **exit**
8. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as 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** {**esp** *local-ip* **interface** *type number*	*local-ip global-ip*}} [**extendable**	**mapping-id** *map-id*| **no-alias**	**no-payload**	**redundancy** *group-name*	**route-map**	**vrf** *name*] **Example:** `Router(config)# ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop`	Enables inside static translation on the VRF.
Step 4	Repeat Step 3 for each VPN being configured.	--

	Command or Action	Purpose
Step 5	**ip route vrf** *vrf-name* **prefix** *prefix mask next-hop-address* **global** **Example:** `Router(config)# ip route vrf shop 0.0.0.0 0.0.0.0 168.58.88.2 global`	Allows the route to be shared by several customers.
Step 6	Repeat Step 5 for each VPN being configured.	--
Step 7	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 8	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by VRF translations.

Configuring Outside Dynamic NAT with MPLS VPNs

Perform this step to configure your NAT PE router for dynamic outside translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool outside** *global-ip local-ip* **netmask** *netmask*
4. **ip nat inside source static** *local-ip global-ip* **vrf** *vrf-name*
5. Repeat Step 4 for each VRF being configured.
6. **ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name*
7. **exit**
8. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

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

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**ip nat pool outside** *global-ip local-ip* **netmask** *netmask* **Example:** `Router(config)# ip nat pool outside 4.4.4.1 4.4.4.254 netmask 255.255.255.00`	Allows the configured VRF to be associated with the NAT translation rule.
Step 4	**ip nat inside source static** *local-ip global-ip* **vrf** *vrf-name* **Example:** `Router(config)# ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop`	Allows the route to be shared by several customers.
Step 5	Repeat Step 4 for each VRF being configured.	Allows the route to be shared by several customers.
Step 6	**ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name* **Example:** `Router(config)# ip nat outside source static 168.58.88.2 4.4.4.1 vrf shop`	Enables NAT translation of the outside source address.
Step 7	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 8	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by VRF translations.

Configuring Outside Static NAT with MPLS VPNs

Perform this task to configure your NAT PE router for static outside translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **ip nat pool inside** *global-ip local-ip* **netmask** *netmask*
4. Repeat Step 3 for each pool being configured.

5. **ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name*
6. Repeat Step 5 for each pool being configured.
7. **ip nat outside source static** *global-ip local-ip* vrf *vrf-name*
8. Repeat Step 7 for all VPNs being configured.
9. **exit**
10. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure** {**terminal** | **memory** | **network**} **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool inside** *global-ip local-ip* **netmask** *netmask* **Example:** `Router(config)# ip nat pool inside1 2.2.1.1 2.2.1.254 netmask 255.255.255.0`	Allows the configured VRF to be associated with the NAT translation rule.
Step 4	Repeat Step 3 for each pool being configured.	--
Step 5	**ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name* **Example:** `Router(config)# ip nat inside source list 1 pool inside2 vrf shop`	Allows the route to be shared by several customers.
Step 6	Repeat Step 5 for each pool being configured.	Defines the access list.
Step 7	**ip nat outside source static** *global-ip local-ip* vrf *vrf-name* **Example:** `Router(config)# ip nat outside source static 168.58.88.2 4.4.4.1 vrf shop`	Allows the route to be shared by several customers.
Step 8	Repeat Step 7 for all VPNs being configured.	--
Step 9	**exit** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config)# exit`	
Step 10	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by VRF translations.

Configuration Examples for Integrating NAT with MPLS VPNs

Configuring Inside Dynamic NAT with MPLS VPNs Example

The following example shows configuring inside Dynamic NAT with MPLS VPNs.

```
!
ip nat pool inside 2.2.2.10 2.2.2.10 netmask 255.255.255.0
ip nat inside source list 1 pool inside vrf bank overload
ip nat inside source list 1 pool inside vrf park overload
ip nat inside source list 1 pool inside vrf shop overload
!
ip route vrf shop 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
ip route vrf bank 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
ip route vrf park 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
!
access-list 1 permit 192.168.0.0 0.0.255.255
```

Configuring Inside Static NAT with MPLS VPNs Example

The following example shows configuring inside static NAT with MPLS VPNs.

```
!
ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop
ip nat inside source static 192.168.122.49 2.2.2.2 vrf shop
ip nat inside source static 192.168.121.113 2.2.2.3 vrf bank
ip nat inside source static 192.168.22.49 2.2.2.4 vrf bank
ip nat inside source static 192.168.121.113 2.2.2.5 vrf park
ip nat inside source static 192.168.22.49 2.2.2.6 vrf park
ip nat inside source static 192.168.11.1 2.2.2.11 vrf shop
ip nat inside source static 192.168.11.3 2.2.2.12 vrf shop
ip nat inside source static 140.48.5.20 2.2.2.13 vrf shop
!
ip route 2.2.2.1 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.2 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.3 255.255.255.255 Serial2/1.1 192.168.121.113
ip route 2.2.2.4 255.255.255.255 Serial2/1.1 192.168.121.113
ip route 2.2.2.5 255.255.255.255 FastEthernet0/0 192.168.121.113
ip route 2.2.2.6 255.255.255.255 FastEthernet0/0 192.168.121.113
ip route 2.2.2.11 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.12 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.13 255.255.255.255 Ethernet1/0 192.168.121.113
```

Configuring Outside Dynamic NAT with MPLS VPNs Example

The following example shows configuring outside dynamic NAT with MPLS VPNs.

```
!
ip nat pool outside 4.4.4.1 4.4.4.254 netmask 255.255.255.0
ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop
ip nat inside source static 192.168.122.49 2.2.2.2 vrf shop
ip nat inside source static 192.168.121.113 2.2.2.3 vrf bank
ip nat inside source static 192.168.22.49 2.2.2.4 vrf bank
ip nat inside source static 192.168.121.113 2.2.2.5 vrf park
ip nat inside source static 192.168.22.49 2.2.2.6 vrf park
ip nat outside source list 1 pool outside
!
```

Configuring Outside Static NAT with MPLS VPNs Example

The following example shows configuring outside static NAT with MPLS VPNs.

```
!
ip default-gateway 10.1.15.1
ip nat pool inside1 2.2.1.1 2.2.1.254 netmask 255.255.255.0
ip nat pool inside2 2.2.2.1 2.2.2.254 netmask 255.255.255.0
ip nat pool inside3 2.2.3.1 2.2.3.254 netmask 255.255.255.0
ip nat inside source list 1 pool inside2 vrf bank
ip nat inside source list 1 pool inside3 vrf park
ip nat inside source list 1 pool inside1 vrf shop
ip nat outside source static 168.58.88.2 4.4.4.1 vrf bank
ip nat outside source static 18.68.58.1 4.4.4.2 vrf park
ip nat outside source static 168.58.88.1 4.4.4.3 vrf shop
ip classless
ip route 192.170.10.0 255.255.255.0 Ethernet1/0 192.168.121.113
ip route 192.170.11.0 255.255.255.0 Serial2/1.1 192.168.121.113
ip route 192.170.12.0 255.255.255.0 FastEthernet0/0 192.168.121.113
ip route vrf shop 0.0.0.0 0.0.0.0 168.58.88.2 global
ip route vrf bank 0.0.0.0 0.0.0.0 168.58.88.2 global
ip route vrf park 0.0.0.0 0.0.0.0 168.58.88.2 global
no ip http server
!
access-list 1 permit 192.168.0.0 0.0.255.255
```

Where to Go Next

- To learn about Network Address Translation 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 use NAT with application level gateways, see the “Using Application Level Gateways with NAT” module.
- To configure NAT for high availability, see the “Configuring NAT for High Availability” module.

Additional References for Integrating NAT with MPLS VPNs

Related Documents

Related Topic	Document Title
IOS Commands	Cisco IOS Master Command List
NAT commands	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard & RFC	Title
RFC 2547	*BGP/MPLS VPNs*

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 Integrating NAT with MPLS VPNs

Table 4: Feature Information for Integrating NAT with MPLS VPNs

Feature Name	Releases	Feature Configuration Information
Integrating NAT with MPLS VPNs	12.1(13)T 15.1(1)SY	The Integrating NAT with MPLS VPNs feature allows multiple Multiprotocol Label Switching (MPLS) VPNs to be configured on a single device to work together.