Chapter 3

Configuring NAT

This chapter describes how to configure Network Address Translation (NAT) on your ERX router; it contains the following sections:

Overview

The Internet faces the challenges of conserving IP address space while continuing to provide scalability in routing. Network Address Translation (NAT) helps address these challenges by allowing the conservation of registered IP addresses within private networks and simplifying IP addressing management tasks through a form of *transparent routing*.

NAT enables you to translate IP addresses between two address realms (for example, between an intranet network that uses private, not publicly routable addresses and the Internet, or between two overlapping, private networks). When incoming traffic is received, the IP addresses are translated back for delivery within the private network.

Using NAT at the edge of your intranet provides the following advantages:

- Allows unregistered *private* addresses to connect to the Internet by translating those addresses into globally registered IP addresses
- Increases network privacy by hiding internal IP addresses from external networks

Platform Considerations

For information about modules that support NAT on ERX-14xx models, ERX-7xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support NAT.

NOTE: The E120 router and the E320 router do not support configuration of NAT.

Module Requirements

To configure NAT on ERX-7xx models, ERX-14xx models, and the ERX-310 router, you must install a Service Module (SM). For information about installing modules in E-series routers, see the *ERX Hardware Guide*.

Unlike other line modules, SMs do not pair with corresponding I/O modules that contain ingress and egress ports. Instead, they receive data from and transmit data to other line modules with access to ingress and egress ports on their own associated I/O modules.

For a list of the modules that support NAT, see *ERX Module Guide, Appendix A, Module Protocol Support*.

References

For more information about NAT, consult the following resources:

- RFC 2663—IP Network Address Translator (NAT) Terminology and Considerations (August 1999)
- RFC 2694—DNS extensions to Network Address Translators (DNS_ALG) (September 1999)
- RFC 2993—Architecture Implications of NAT (November 2000)
- RFC 3022—Traditional IP Network Address Translator (Traditional NAT) (January 2001)
- RFC 3027—Protocol Complications with the IP Network Address Translator (January 2001)

NAT Configurations

You can configure NAT in several different ways. Each of the following configuration methods provides a solution for different configuration requirements:

- Traditional NAT
- Bidirectional NAT
- Twice NAT

Traditional NAT

Traditional NAT is the most common method of using address translation. Its primary use is translating private addresses to legal addresses for use in an external network. When configured for dynamic operation, hosts within a private network can initiate access to the external (public) network, but external nodes on the outside network cannot initiate access to the private network.

Addresses on the private network and public network must not overlap. Also, route destination advertisements on the public network (for example, the Internet) can appear within the inside network, but the NAT router does not propagate advertisements of local routes that reference private addresses out to the public network.

There are two types of traditional NAT—basic NAT and NAPT.

Basic NAT

Basic NAT provides translation for IP addresses only (called a *simple* translation) and places the mapping into a NAT table. In other words, for packets outbound from the private network, the NAT router translates the source IP address and related fields (for example, IP, TCP, UDP, and ICMP header checksums). For inbound packets, the NAT router translates the destination IP address (and related checksums) for entries that it finds in its translation table.

CAUTION: Although NAT is the simplest translation method, it is the least secure. By not including port or external host information in the translation, basic NAT allows access to any port of the private host by any external host.

NAPT

Network Address Port Translation (NAPT) extends the level of translation beyond that of basic NAT; it modifies both the IP address and the transport identifier (for example, the TCP or UDP port number, or the ICMP query identifier) and places the mapping into the translation table (this entry is called an *extended* translation). This method can translate the addresses and transport identifiers of many private hosts into a few external addresses and transport identifiers, to make efficient use of globally registered IP addresses.

Similar to basic NAT, for outbound packets NAPT translates the source IP address, source transport identifier, and related checksum fields. For inbound packets NAPT translates the destination IP address, destination transport identifier, and checksum fields.

Bidirectional NAT

Bidirectional (or two-way) NAT adds support to basic NAT for the Domain Name System (DNS) so public hosts can initiate sessions into the private network, usually to reach servers intended for public access.

When an outside host attempts to resolve the name of an inside host on a private network, the NAT router intercepts the DNS reply and installs an address translation to allow the outside host to reach the inside host by using a public address. When the outside host initiates a connection with the inside host on the private network, the NAT router translates that public destination address to the private address of the inside host and, on the return path, replaces the source address with the advertised public address.

You might need to perform some additional configuration to allow public access from the Internet to a DNS server that resides in the private domain. (See *Bidirectional NAT Example* on page 99.)

The same address space requirements and routing restrictions apply to bidirectional NAT that were described for traditional NAT. The difference between these two methods is that the DNS exchange might create entries within the translation table.

Twice NAT

In twice NAT, both the source and destination addresses are subject to translation as packets traverse the NAT router in either direction. For example, you would use twice NAT if you are connecting two networks in which all or some addresses in one network overlap addresses in another network, whether the network is private or public.

Network and Address Terms

The NAT implementation defines an address realm as either *inside* or *outside*, with the router that is running NAT acting as the defining boundary between the two realms.

From a NAT perspective, an *inside* network is the local portion of a network that uses private, not publicly routable IP addresses that you want to translate. An *outside* network is the public portion of a network that uses legitimate, publicly routable IP addresses to which you want private hosts to connect.

The addresses that are translated by NAT between address realms are labeled as inside or outside, and as local or global. When reading the terms in the following sections, keep the following definitions in mind:

- The terms *inside* and *outside* refer to the host that the address is associated with.
- The terms *local* and *global* refer to the network on which the address appears.

Inside Local Addresses

The *inside local* address is a configured IP address that is assigned to a host on the inside network. Addresses may be globally unique (not requiring translation), allocated from the private address space defined in RFC 1918, or officially allocated to some other organization.

Inside Global Addresses

The *inside global* address is the *translated* IP address of an inside host as seen by an outside host and network. Addresses may be allocated from a globally unique address space (often provided by the ISP, if the inside address is connected to the global Internet).

Outside Local Addresses

The *outside local* address is the *translated* IP address of an outside host as it appears to the inside network. Addresses may be globally unique (not requiring translation), allocated from the private address space defined in RFC 1918, or officially allocated to some other organization.

Outside Global Addresses

The *outside global* address is the configured, publicly routable IP address assigned to a host on the outside network.

Understanding Address Translation

Address translation can occur one of two ways: inside or outside source translation.

Inside Source Translation

Inside source translation is the most commonly used NAT configuration. When an inside host sends a packet to the outside network, the NAT router translates the source information (either the source address or the source address/port pair) and, in the inbound direction, restores the original information (this time operating on the destination address or address/port pair).

For outbound traffic, the NAT router translates the inside local address (or address/port) into the inside global address (or address/port), either through a statically defined translation or dynamically created translation. For inbound traffic, a translation must be found to revert the inside global address (or address/port) into the inside local address (or address/port), or the packet is not routed into the inside network.

NOTE: Dynamic inside source translations are established by outbound traffic.

You use inside source translation in traditional and bidirectional NAT configurations.

Outside Source Translation

Outside source translation is used in NAT configurations only when addresses of external hosts might create a conflict on the private network. This complementary translation process is performed on the opposite addressing fields in the IP packet. When an outside host sends a packet to the inside network, the NAT router translates the source information (either the source address or the source address/port pair) and, in the outbound direction, restores the original information (this time operating on the destination address or address/port pair).

For inbound traffic, the NAT router translates the outside global address (or address/port) into the outside local address (or address/port), either through a statically defined translation or dynamically created translation. For outbound traffic, a translation must be found to revert the outside local address (or address/port) into the outside global address (or address/port), or the packet is not routed into the outside network.

NOTE: Dynamic outside source translations are established by inbound traffic.

You use outside source translation along with inside source translation to configure twice NAT.

Address Assignment Methods

NAT uses one of two methods to assign a translated IP address: static translation or dynamic translation.

Static Translations

You enter static translations as direct configuration settings that remain in the translation table until you remove them. You use static translations when you must initiate connections from both the inside and outside interfaces, or when the translation is not subject to change.

Dynamic Translations

Dynamic translations use access list rules, to determine whether to apply NAT to incoming traffic, and NAT address pools, from which a NAT translation can obtain IP addresses. You use dynamic translation when you want the NAT router to initiate and manage address translation and session flows between address realms on demand.

Order of Operations

This section describes the order of operations for both inside-to-outside and outside-to-inside translation.

Inside-to-Outside Translation

Inside-to-outside translation occurs in the following order:

1. Inside (privately addressed) traffic enters the router on an interface marked as *inside*.
2. A route lookup is performed.
3. If the next interface is marked as *outside*, the router sends the traffic to the server module.
4. The server module performs the appropriate translation.
5. The router forwards the packet to the appropriate egress line module.
6. The line module sends the packet as outbound traffic using a globally unique source address (inside source translation), destination address (outside source translation), and ports (NAPT).

Outside-to-Inside Translation

Outside-to-inside translation occurs in the following order:

1. Traffic from the outside, public domain enters the router.
2. All traffic from an interface that is marked *outside*, whether or not it requires NAT, is sent to the server module.
3. The server module searches for an associated NAT match.
4. If the server module:
   - Finds a NAT match, and the destination interface is marked as *inside*, the server module performs the appropriate translation and sends the packet to the appropriate destination.
   - Does not find a NAT match, and the destination interface is marked as *inside*, the server module drops the packet.
   - Does not find a NAT match, and the destination interface is not marked as *inside*, the server module processes the packet normally for its destination.

PPTP and GRE Tunneling Through NAT

You can configure NAT traversal support for GRE flows using simple translations (Basic NAT). Because PPTP uses an enhanced GRE encapsulation for the PPP payload, configuring for GRE flows also supports NAT traversal for PPTP tunnels.

NOTE: Neither port translation (NAPT) nor Firewall traversal for GRE packets is supported for GRE flows.

When configured, the following types of translations are supported for GRE and PPTP tunnels:

- Inside source static simple translations (inbound and outbound)
- Outside source static simple translations (inbound and outbound)
- Inside source dynamic simple translations (inbound and outbound)
- Outside source dynamic simple translations (inbound and outbound)
- Combinations of the preceding translations (for example, twice NAT)

Packet Discard Rules

For all supported types of traffic (TCP, UDP, ICMP, and GRE), NAT discards packets in the following cases:

- When the translation table is full (that is, no more entries can be added).
- When the address pool is exhausted for outbound packets with inside source dynamic translation.
- When no match can be found for the destination addresses of inbound packets.
- When the address pool is exhausted for inbound packets with outside source dynamic translation.

In addition, NAT discards GRE packets under the following conditions:

- When the GRE packets match an NAPT rule.
- When Firewall is functioning.

Before You Begin

You can configure certain IP interfaces to participate in Network Address Translation. This chapter discusses how to configure NAT to function for certain IP interfaces. For information about general IP interface configuration, see *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*.

Configuring a NAT License

You must configure a NAT license before you can use any NAT commands on the ERX router.

license nat

- Use to specify a NAT license.
- Purchase a NAT license to allow NAT configuration on the ERX router.

NOTE: Acquire the license from Juniper Networks Customer Services and Support or from your Juniper Networks sales representative.

- Example

  host1(config)#**license nat** *license-value*

- Use the **no** version to disable the license.

Limiting Translation Entries

You can configure the maximum number of dynamic translation entries that the translation table contains in global configuration mode for a given virtual router.

ip nat translation max-entries

- Use to specify the maximum number of dynamic translation entries that the translation table can contain in global configuration mode for the given virtual router.
- Example

  host:VR1 (config-if) #**ip nat translation max-entries 1000**

- Use the **no** version to remove the configured limit and return the maximum number of translation entries to the default, which is no enforced limit, as capacity allows.

Specifying Inside and Outside Interfaces

You must mark interfaces that participate in NAT translation as residing on the inside or the outside network.

CAUTION: Only packets routed between an inside and an outside interface are subject to translation.

You can unmark an interface by using the **no** version of this command.

ip nat

- Use to mark an IP interface as participating in NAT translation.
- Use the keyword (**inside** or **outside**) to specify the side of the network on which the interface resides.
- Example

  host (config-if) # **ip nat inside**

- Use the **no** version to unmark the interface (the default) so that it does not participate in NAT translation.

Defining Static Address Translations

Static address translation establishes a one-to-one mapping between a local and global address or local and global address/port pair. When you specify a static address translation or address/port pair translation, you issue commands to indicate how the translation is applied, along with more specific variables that further define the type of translation.

CAUTION: You must mark interfaces that participate in NAT translation as on the inside or the outside network. See *Specifying Inside and Outside Interfaces* on page 89 for details.

Creating Static Inside Source Translations

You use the **ip nat inside source static** command to create static translations from a local IP address to a global IP address, and to *untranslate* the destination address when a packet returns from the outside network to the inside network. When you configure traditional NAT (both basic NAT and NAPT), you only need to use this command alone. However, when you configure twice NAT, you must also use the **ip nat outside source static** command.

The **ip nat inside source static** command creates a simple (IP address only) or extended (IP address, port, and protocol) entry in the translation table that maps the two addresses.

ip nat inside source static

- Use to create static translations for a source address (or address/port pair) when routing a packet from the inside network to the outside network, and to *untranslate* the destination address (or address/port pair) when a packet returns from the outside network to the inside network.
- A static translation created with the **ip nat inside source static** command enables any outside host to contact the inside host by using the inside global address of the inside host. A static translation can be used by traffic that is initiated in either direction
- Example 1—Simple address translation

  host (config) # **ip nat inside source static 10.1.2.3 171.69.68.10**

- Example 2—Extended address/port translation

  host (config) # **ip nat inside source static tcp 10.1.2.3 15 171.69.68.10 30**

- Use the **no** version to remove the static translation and purge the associated translations from the translation table.

Creating Static Outside Source Translations

Less commonly used, outside source translation enables you to set up translation between two non-unique or not publicly routable networks (for example, two separate networks that use overlapping IP address blocks).

ip nat outside source static

- Use to translate the source address when routing a packet from the outside network to the inside network, and to *untranslate* the destination address when a packet travels from the inside network to the outside network.
- Creates a simple (IP address only) or extended (IP address, protocol, and port) entry in the translation table that maps the two addresses.
- A static translation created with the **ip nat outside source static** command enables any inside host to contact the outside host by using the outside local address of the outside host. A static translation can be used by traffic that is initiated in either direction.
- Example 1—Simple address translation

  host (config) # **ip nat outside source static 171.69.68.10 10.1.2.3**

- Example 2—Extended address/port translation

  host (config) # **ip nat outside source static tcp 171.69.68.10 56 10.1.2.3 24**

- Use the **no** version to remove the static translation and purge the associated translations from the translation table.

Defining Dynamic Translations

Dynamic translations use access list rules, to determine whether or not to apply NAT to incoming traffic, and NAT address pools, from which a NAT translation can allocate IP addresses. You use dynamic translation when you want the NAT router to initiate and manage address translation and session flows between address realms on demand.

To configure dynamic translations:

- Define any access list rules that the NAT router uses to decide which packets need translation.
- Define an address pool from which the NAT router obtains addresses.
- Define inside and outside source translation rules for the NAT router to create NAT translations.
- Mark interfaces as *inside* or *outside*.
- (Optional) Modify any translation timeout values.

Creating Access List Rules

Before you create a dynamic translation, create the access list rules that you plan to apply to the translation. For information about configuring access lists, see *Chapter 1, Configuring Routing Policy*.

The router evaluates multiple commands for the same access list in the order they were created. An undefined access list implicitly contains a rule to *permit any*. A defined access list implicitly ends with a rule to *deny any*.

NOTE: The access lists do not filter any packets; they determine whether the packet requires translation.

You use the **access-list** command to create an access list.

access-list

- Use to define an IP access list to permit or deny translation based on the addresses in the packets.
- Each access list is a set of permit or deny conditions for routes that are candidates for translation (that is, moving from the inside network to the outside network).
- A zero in the wildcard mask means that the route must exactly match the corresponding bit in the address. A one in the wildcard mask means that the route does not have to match the corresponding bit in the address.

- Use the **log** keyword to log an Info event in the ipAccessList log whenever matching an access list rule.
- Example

  host1(config)#**access-list bronze permit ip host any 228.0.0.0 0.0.0.255**

- Use the **no** version to delete the access list (by not specifying any other options), the specified entry in the access list, or the log for the specified access list or entry (by specifying the **log** keyword).

Defining Address Pools

Before you can configure dynamic translation, create an address pool. An address pool is a group of IP addresses from which the NAT router obtains an address when dynamically creating a new translation. You can create address pools with either a single range or multiple, nonoverlapping ranges.

When you create a single range, you specify the starting and ending IP addresses for the range in the root **ip nat pool** command. However, when you create multiple, nonoverlapping ranges, you omit the optional starting and ending IP addresses in the root **ip nat pool** command; this launches the IP NAT Pool Configuration (config-ipnat-pool) mode.

The config-ipnat-pool mode uses an **address** command to specify a range of IP addresses. You can repeat this command to create multiple, nonoverlapping ranges.

When you create or edit address pools, keep the following in mind:

- Starting and ending IP addresses for the specified range are inclusive and must reside on the same subnet.
- Address ranges are verified against other ranges in the specified pool to exclude range overlaps. Additional verification occurs when the pool is associated with a translation rule and the router can determine whether the rule is inside or outside.
- You cannot change the network mask if configured ranges already exist.
- The network mask (or prefix length) is used to recognize host addresses that end in either all zeros or all ones. These addresses are reserved as broadcast addresses and are not allocated from an address pool, even if they are included in an address pool range.
- You cannot remove an address pool if the pool is part of a translation rule or if any of the ranges within the pool are still in use. You must issue the **clear ip nat translation** command to clear any ranges before you can remove the pool to which they apply.

address

- Use to specify a range of IP addresses in config-ipnat-pool mode; you can repeat the **address** command to create multiple ranges.
- Example

  host (config-ipnat-pool)#**address 171.69.40.110 171.69.40.115**

- Use the **no** version to remove the range for the current address pool.

ip nat pool

- Use to create address pools.
- Example 1—Creating a single, continuous range

  host (config) #**ip nat pool singlerange 171.69.40.1 171.69.40.100 prefix-length 30**

- Example 2—Creating multiple, discontinuous ranges

  host (config) #**ip nat pool multiplerange prefix-length 30**
  host (config-ipnat-pool)#**address 171.69.40.110 171.69.40.112**
  host (config-ipnat-pool)#**address 171.69.40.118 171.69.40.120**
  host (config-ipnat-pool)#**exit**

- Use the **no** version to remove the address range.

Defining Dynamic Translation Rules

You can use the CLI to define dynamic translation rules for inside and outside sources.

CAUTION: You must mark interfaces that participate in NAT translation as on the inside or the outside network. See *Specifying Inside and Outside Interfaces* on page 89 for details.

You can create a dynamic translation rule to configure inside source or outside source translation. If the NAT router cannot locate a matching entry in its translation database for a given packet, it evaluates the access list of all applicable dynamic translation rules (inside source translation rules for outbound packets and outside source translation rules for inbound packets) against the packet. If an access list permits translation, the NAT router tries to allocate an address from the associated address pool to install a new translation.

When you create dynamic translation rules, keep the following in mind:

- You can associate a list with one pool at any given time. Associating a list with a different pool replaces the previous association.
- The optional **overload** keyword for inside source translation specifies that the router employ NAPT.
- You can configure dynamic NAPT for inside source translation only; you cannot configure dynamic NAPT for outside source translation.

- When no match occurs for any dynamic translation rule, the NAT router does not translate the packet.
- When an address pool is empty, the NAT router drops the packet.
- Access lists and pools do not have to exist when you are defining dynamic translation rules; you may create them after you define the dynamic translations.

Creating Dynamic Inside Source Translation Rules

Use the **ip nat inside source list** command to create a dynamic inside source translation rule. This command creates a translation rule that:

- Translates inside local source addresses to inside global addresses when packets from the inside network are routed to the outside network
- Translates outside local source addresses to outside global addresses when packets from the outside network are routed to the inside network.
- Use the **overload** keyword to specify that the translation create NAPT entries (protocol, port, and address) in the NAT table.

The **no** version of this command removes the dynamic translation rule, but does not remove any previously created translations (resulting from the rule evaluation) from the translation table. To remove active translations from the translation table, see *Clearing Dynamic Translations* on page 96.

ip nat inside source list

- Use to create dynamic translation rules that specify when to create a translation for a source address when routing a packet from the inside network to the outside network.
- Example

  host (config) #**ip nat inside source list translation1 pool pool1**

- Use the **overload** keyword to specify that the translation create extended entries (protocol, port, and address) in the translation table for NAPT.
- Use the **no** version to remove the dynamic translation rule; this command does not remove any dynamic translations from the translation table.

Creating Dynamic Outside Source Translation Rules

Use the **ip nat outside source list** command to create a dynamic outside source translation rule. This command dynamically translates outside global source addresses to outside local addresses when packets are routed from the outside network to the inside network (and *untranslates* the destination address when a packet returns before a translation table entry times out).

The **no** version of this command removes the dynamic translation rule, but does not remove any previously created translations from the translation table. To remove active translations from the translation table, see *Clearing Dynamic Translations* on page 96.

ip nat outside source list

- Use to create dynamic translation rules that specify when to create a translation for a source address when routing a packet from the outside network to the inside network.
- Example

  host (config) # **ip nat outside source list translation1 pool pool1**

- Use the **no** version to remove the dynamic translation rule; this command does not remove any dynamic translations from the translation table.

Defining Translation Timeouts

The router removes unused dynamic translations in the translation table. Use the **ip nat translation** command to change or disable NAT translation timeouts.

You can set the aging time (in seconds) for any of the specified timers:

- timeout—Dynamic simple translations (not for overloaded translations); default is 86400 seconds (24 hours).
- dns-timeout—DNS-created protocol translations; default is 120 seconds. These dynamic translations are installed by the DNS but not yet used; as soon as the translation is used, the router applies the timeout value mentioned above.
- udp-timeout—UDP protocol extended translations; default is 300 seconds (5 minutes).
- tcp-timeout—TCP protocol extended translations; default is 86400 seconds (24 hours).
- finrst-timeout—TCP connections terminated with reset (RST) or bidirectional finished (FIN) flags; default is 120 seconds. This timeout applies only to TCP extended translations. The timer removes unused, closed TCP translations, which allows for retransmissions.
- icmp-timeout—ICMP protocol extended translations; default is 300 seconds (5 minutes).
- gre-timeout—Aging time for GRE protocol translations; default value is 300 seconds (5 minutes)

All timeouts for this command support a maximum value of 2147483 seconds (about 25 days).

The **no** version of this command resets the timer to its default value.

ip nat translation

- Use to change translation timeouts for existing and newly created translations in the translation table.
- All timeouts for this command support a maximum value of 2147483 seconds (about 25 days).
- Example

  host1 (config) # **ip nat translation timeout 23200**

- Use the **no** version to reset the timer to its default value.

Clearing Dynamic Translations

Use the **clear ip nat translation command** to clear dynamic translations from the NAT translation table. You can remove all dynamic translations from the translation table or restrict the removal of translation entries based on the protocol, address, or port values.

clear ip nat translation

- Use to clear dynamic translations from the NAT translation table.
- Use an asterisk (*) in the **clear ip nat translation** version of this command to clear all dynamic translations from the translation table.
- Use an asterisk (*) in the **clear ip nat translation { gre | icmp | tcp | udp } inside** *insideGlobalIpAddress* ***** *insideLocalIpAddress* ***** version of this command to match any global or local port and remove inside source extended GRE, ICMP, TCP, or UDP translations for the specified global IP address and local IP address.
- Example 1—Clear all dynamic translations

  host1 #**clear ip nat translation***

- Example 2—Clear a specific port translation

  host1 #**clear ip nat translation tcp inside 171.69.68.10 10.1.2.3 55**

- There is no **no** version.

NAT Configuration Examples

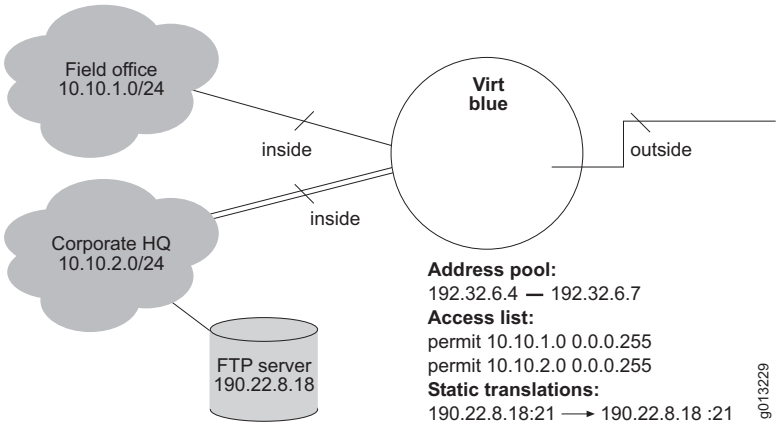
This section contains NAT configuration examples for a single virtual router configuration and NAT translation between two virtual routers.

NAPT Example

Figure 6 illustrates a NAPT configuration for a private network with two inside subnetworks, a field office, and a corporate office.

Both offices use private addresses. The corporate office has a dual T-3 link and a public FTP server that has a global address (that is, it does not need translation).

Figure 6: NAPT Example



The address pool consists of three addresses (the number of addresses is small, because NAPT is used). Addresses matching the private address spaces of the corporate and field subnetworks are translated to global addresses from the pool through NAPT.

To configure this example:

1. Enter the correct virtual router context.

   host1(config)#**virtual-router blue**

2. Mark the inside interfaces.

   a. Mark the field office:

      host1:blue(config)#**interface serial 2/1:1/1**
      host1:blue(config-interface)#**ip nat inside**
      host1:blue(config-interface)#**exit**

   b. Mark the two corporate T-3 links:

      host1:blue(config)#**interface serial 1/1**
      host1:blue(config-interface)#**ip nat inside**
      host1:blue(config-interface)#**exit**

```
host1:blue(config)#interface serial 1/2
host1:blue(config-interface)#ip nat inside
host1:blue(config-interface)#exit
```

3. Mark the outside interface.

```
host1:blue(config)#interface gigabitEthernet 3/0.1
host1:blue(config-interface)#ip nat outside
host1:blue(config-interface)#exit
```

4. Create a static nil-translation for the FTP server on the corporate network.

```
host1:blue(config)#ip nat inside source static tcp 190.22.8.18 21 190.22.8.18
21
```

5. Create the address pool for dynamic translations.

```
host1:blue(config)#ip nat pool corpxyz 192.32.6.4 192.32.6.7 prefix-length 24
```

6. Create the access list for addresses eligible for dynamic translation.

```
host1:blue(config)#access-list justcorp permit 10.10.1.0 0.0.0.255
host1:blue(config)#access-list justcorp permit 10.10.2.0 0.0.0.255
```

7. Create the NAPT dynamic translation rule.

```
host1:blue(config)#ip nat inside source list justcorp pool corpxyz overload
```

8. Configure a default route to the outside interface.

```
host1:blue(config)#ip route 0.0.0.0 0.0.0.0 gigabitEthernet 3/0.1
```

9. Configure a null route for the inside global addresses to prevent routing loops when no matching translation exists.

```
host1:blue(config)#ip route 192.32.6.0 255.255.255.248 null 0
```

NOTE: Null route applies to 192.32.6.0–192.32.6.3, which do not exist in the address pool

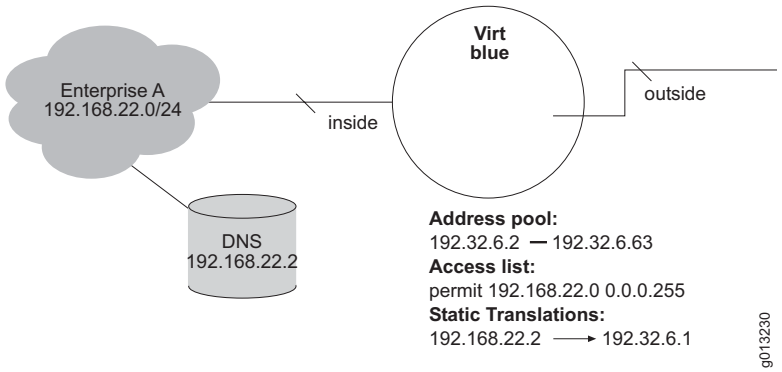
All hosts that use private addresses in both the field office and the corporate office must have their addresses translated to one of the three addresses in the pool. Because this example uses NAPT, the interface can use only one pool address, depending on the number of inside hosts attempting to access the outside at any given time.

Bidirectional NAT Example

Figure 7 illustrates how outside hosts can initiate conversations with inside hosts through the use of a DNS server that resides on the inside network.

The inside realm uses basic NAT. The inside network uses a mix of private subnetwork address space (192.168.22/24) and registered public addresses.

Figure 7: Bidirectional NAT Example



To configure this example:

1. Enter the correct virtual router context.

   host1(config)#**virtual-router blue**

2. Mark the inside interface.

   host1:blue(config)#**interface serial 1/1:1/1**
   host1:blue(config-interface)#**ip nat inside**
   host1:blue(config-interface)#**exit**

3. Mark the outside interface.

   host1:blue(config)#**interface gigabitEthernet 3/0.1**
   host1:blue(config-interface)#**ip nat outside**
   host1:blue(config-interface)#**exit**

4. Create the translation for the DNS.

   host1:blue(config)#**ip nat inside source static 192.168.22.2 192.32.6.1**

5. Create the address pool for dynamic translations.

   host1:blue(config)#**ip nat pool entA192 192.32.6.2 192.32.6.63 prefix-length 24**

6. Create the access list for addresses eligible for dynamic translation (that is, private addresses).

   host1:blue(config)#**access-list entA permit 192.168.22.0 0.0.0.255**

7. Create the dynamic translation rule.

   host1:blue(config)#**ip nat inside source list entA pool entA192**

8. Configure a default route to the outside interface.

   host1:blue(config)#**ip route 0.0.0.0 0.0.0.0 gigabitEthernet 3/0.1**

9. Configure a null route for the inside global addresses, to prevent routing loops when no matching translation exists.

   host1:blue(config)#**ip route 192.32.6.0 255.255.255.192 null 0**

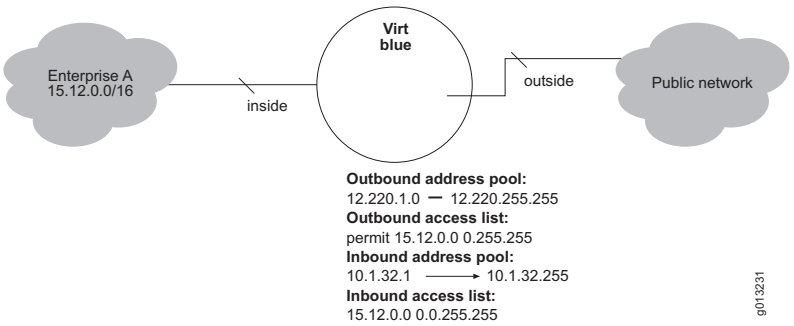
NOTE: Null route applies to 192.32.6.0 and 192.32.6.1, which do not exist in the address pool.

Twice NAT Example

Twice NAT is often useful when the inside network is using a nonprivate address space (unregistered usage of global address space) and you want it to connect to the public network. Inside local addresses need to be translated to legal global addresses. Legal addresses from the outside that overlap those used on the inside network need to be translated to unused and recognizable addresses in the inside network. Both inside source and outside source translations must be configured on the NAT router.

Figure 8 illustrates how the inside network is using the unregistered global address space of 15.12.0.0/16. Outside hosts whose addresses overlap with this subnetwork that want to access the inside network need their global addresses translated.

Figure 8: Twice NAT Example



To configure this example:

1. Enter the correct virtual router context.

   host1(config)#**virtual-router blue**

2. Mark the inside interface.

   host1:blue(config)#**interface fast-ethernet 6/1**
   host1:blue(config-interface)#**ip nat inside**
   host1:blue(config-interface)#**exit**

3. Mark the outside Interface.

   host1:blue(config)#**interface atm 3/0.20**
   host1:blue(config-interface)#**ip nat outside**
   host1:blue(config-interface)#**exit**

4. Create the address pool for inside source translations.

   host1:blue(config)#**ip nat pool entAoutpool 12.220.1.0 12.220.255.255 prefix-length 16**

NOTE: This pool is purposely smaller than the size of the company network because not all private hosts are likely to access the public network at the same time.

5. Create the access list for addresses eligible for dynamic translation.

   host1:blue(config)#**access-list entAout permit 15.12.0.0 0.0.255.255**

6. Create the dynamic translation rule for outbound traffic.

   host1:blue(config)#**ip nat inside source list entAout pool entAoutpool**

7. Create the address pool for outside source translations.

   Using an address range of 10.1.32.0/8 prevents any overlap with the private network (15.12.0.0/16).

   host1:blue(config)#**ip nat pool entAinpool 10.1.32.1 10.1.32.255 prefix-length 16**

NOTE: This pool is purposely small, allowing for only a few connections.

8. Configure the access list for global addresses that overlap with inside addresses.

   host1:blue(config)#**access-list entAin permit 15.12.0.0 0.0.255.255**

9. Create the dynamic translation rule for inbound traffic.

   host1:blue(config)#**ip nat outside source list entAin pool entAinpool**

10. Create one of the following:

    - A route to the outside interface for inside hosts to access outside hosts that have overlapping addresses.

      host1:blue(config)#**ip route 10.1.32.0 255.255.255.0 atm 3/0.1**

NOTE: An inside host cannot directly access hosts on the outside network that use addresses that overlap with the inside subnetwork. However, by using outside source translation and DNS name resolution, the NAT router can install translations so inside hosts can access these outside hosts by using nonoverlapping addresses.

    - A default route to the outside interface.

      host1:blue(config)#**ip route 0.0.0.0 0.0.0.0 atm 3/0.1**

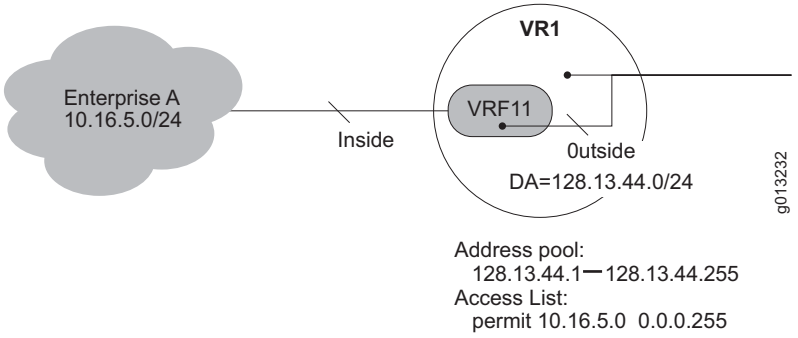
11. Configure a null route for the inside global addresses to prevent routing loops when no matching translation exists.

    host1:blue(config)#**ip route 12.220.1.0 255.255.0.0 null 0**

Cross-VRF Example

In MPLS VPN configurations, you might want to offer public Internet access to VPN subscribers. MPLS VPNs are enabled through the use of VRFs. If a VPN is using a private or overlapping address space, you can use NAT to enable access to the public network because the NAT implementation is both VR and VRF aware. Figure 9 illustrates how the subscriber interface feature of the router is used in conjunction with NAT to connect the VPNs to the public network.

Figure 9: Cross-VRF Example



VRF11 is the local (this PE) representation of the MPLS VPN and connects enterpriseA to the VPN. Enterprise A communicates to VRFs in other PE devices (the rest of the VPN) through RFC2547bis (MPLS VPNs). VR1, of which the VRF is administratively a member, represents the public network. The interface to EnterpriseA is marked as an inside interface. The normal steps for configuring inside source translation are applied. A subscriber interface is created off the uplink to the core network and anchored in the VRF. A DA-based demultiplexer matching the inside global address range is configured on the subscriber interface. The subscriber interface is marked as an outside interface.

To configure this example:

1. Enter the correct virtual routing and forwarding instance.

   host1(config)#**virtual-router vr1:vrf11**

2. Mark the inside interfaces.

   host1:vr1:vrf11(config)#**interface fast-ethernet 6/1**
   host1:vr1:vrf11 (config-interface)#**ip nat inside**
   host1:vr1:vrf11 (config-interface)#**exit**

3. Set the primary interface to DA-type demultiplexer (for subsequent shared interfaces).

   host1:vr1(config)#**interface atm 12/0.101**
   host1:vr1(config-interface)#**ip demux-type da-prefix**
   host1:vr1(config-interface)#**exit**

4. Create the address pool for dynamic translations.

   host1:vr1(config)#**virtual-router vr1:vrf11**
   host1:vr1:vrf11(config)#**ip nat pool entApool 128.13.44.0 128.13.44.255 prefix-length 24**

5. Create the access list for addresses eligible for dynamic translation.

   host1:vr1:vrf11(config)#**access-list entA permit 10.16.5.0 0.0.0.255**

6. Create the dynamic translation rule.

   host1:vr1:vrf11(config)#**ip nat inside source list entA pool entApool**

7. Create the subscriber interface off the uplink.

   host1:vr1:vrf11(config)#**interface ip vrf11vr1**
   host1:vr1:vrf11(config-interface)#**ip share-interface atm 12/0.101**
   host1:vr1:vrf11(config-interface)#**ip unnumbered loopback 1**

8. Configure a group of destination prefixes with which the device can communicate on the public network.

   host1:vr1:vrf11(config-interface)#**ip destination-prefix 128.13.44.0 255.255.255.0**

9. Mark the subscriber interface as outside.

   host1:vr1:vrf11(config-interface)#**ip nat outside**
   host1:vr1:vrf11(config-interface)#**exit**

10. Point the default route to the shared interface.

    host1:vr1:vrf11(config)#**ip route 0.0.0.0 0.0.0.0 ip vrf11vr1**

11. Install a null route to avoid routing loops to the inside global address.

    host1:vr1:vrf11(config)#**ip route 128.13.44.0 255.255.255.0 null 0**

Tunnel Configuration Through NAT Examples

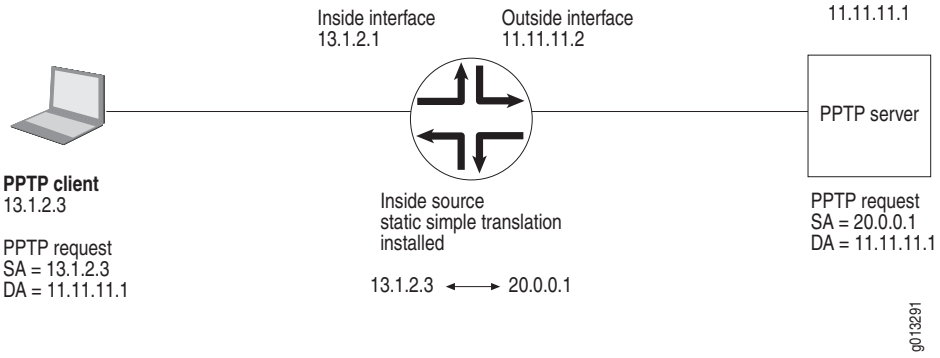
PPTP uses enhanced GRE encapsulation for PPP payloads. After the PPTP tunnel setup process, PPP packets are exchanged using GRE encapsulation. It is critical that a NAT device that resides between PPTP client and PPTP server allow GRE flows.

This section contains NAT configuration examples for both inside and outside PPTP tunnel setup through NAT.

Clients on an Inside Network

In this example, a subscriber on the inside network is initiating PPTP tunnels to a PPTP server located in the outside network. The PPTP connection to the server traverses an E-series router that has NAT enabled.

Figure 10: PPTP Tunnels on an Inside Network



The router has installed an inside source static simple translation in its translation table as follows:

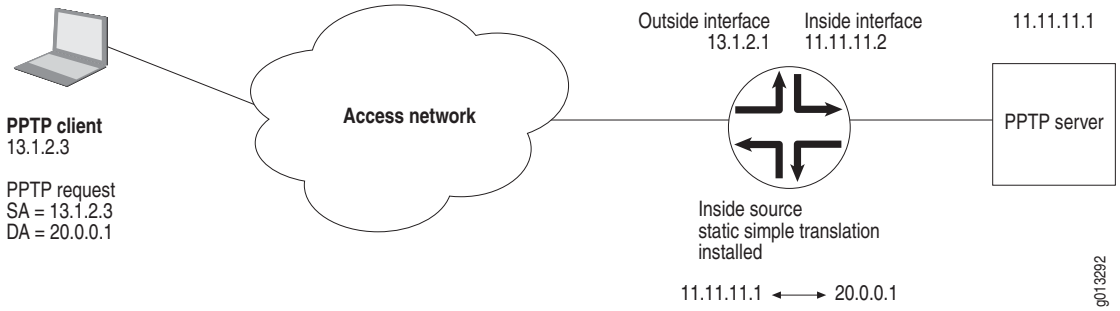
Inside Local Address	Inside Global Address
13.1.2.3	20.0.0.1

The PPTP client initiates its tunnels to the server at 11.11.11.1. The E-series router translates the SA from inside local 13.1.2.3 to inside global SA 20.0.0.1. Because GRE traffic can pass through NAT, all matching PPTP control packets are translated and forwarded to the destination.

Clients on an Outside Network

In this example, an outside subscriber initiates PPTP tunnels to a PPTP server located in the service provider network. The PPTP connection to the server traverses an E-series router that has NAT enabled.

Figure 11: PPTP Tunnels on an Outside Network



The router has installed an inside source static simple translation in its translation table as follows:

Inside Local Address	Inside Global Address
11.11.11.1	20.0.0.1

The PPTP client initiates its tunnels to the inside global address 20.0.0.1. The E-series router translates packets destined for address 20.0.0.1 and forwards them to the inside local address of 11.11.11.1. Because GRE traffic can pass through NAT, all matching PPTP control packets are translated and forwarded to the destination.

GRE Flows Through NAT

Because PPTP requires the use of GRE flows, the examples in the previous section also work for any GRE traffic flows that traverse NAT.

GRE flows can terminate at an E-series router if NAT is or is not enabled. When the router receives locally terminating inbound GRE packets, the router transmits the packets to the tunnel server module for GRE processing. If the packets require translating, they are again sent through the tunnel server module.

NOTE: Only inner IP headers are translated for terminating GRE flows; outer IP headers are never translated.

For outbound GRE packets, the process works in reverse. If the packets require translation, the router transmits the packets to the tunnel server module for translation. If the packets are destined for a GRE tunnel, they are again sent through the tunnel server module where an outer header is prepended to the packet and the packet is then sent to the appropriate GRE tunnel.

Monitoring NAT

This section explains how to view NAT license information, NAT statistics, NAT translation entries, NAT address pool information, and NAT inside and outside rule settings.

Displaying the NAT License Key

The **show license nat** command displays the NAT license key.

show license nat

- Use to display the NAT license key configured on the router.
- Example

```
host1#show license nat
Nat license is nat_license
```

Displaying Translation Statistics

The **show ip nat statistics** command displays internal statistics that apply to NAT operation.

show ip nat statistics

- Use to display internal NAT statistics.
- Field descriptions
  - Last dynamic allocation failure—Completion level of any dynamic allocation failures; the number of times the router attempted dynamic allocation but reached the dynamic allocation entry limit
  - Current static translation entries
    - Inside Source Simple—Number of inside source simple static translations
    - Outside Source Simple—Number of outside source simple static translations
    - Inside Source Extended—Number of inside source extended static translations
    - Outside Source Extended—Number of outside source extended static translations
  - Dynamic Translation Type—Type of dynamic translation (inside source simple, outside source simple, inside source extended)
  - Current—Current number of dynamic translations of the associated translation type

- Peak—Peak number of dynamic translations of the associated translation type
- Accumulated—Accumulated number of dynamic translations of the associated type; this value reflects the accumulation of dynamic translations since the last router reboot operation
- Failed—Total number of installation attempts that failed for an associated translation type
- Forwarding statistics for packets received on inside or outside interfaces
  - forwarded directly—Number of packets forwarded directly (that is, without the need of translation)
  - forwarded through translator—Number of packets forwarded through the NAT translator
  - discarded—Number of packets discarded immediately upon receipt
  - discarded by translator—Number of packets discarded by the NAT translator when no matching translation could be located

- Example

```
host1#show ip nat statistics
NAT database statistics for virtual router vr1:
---------------------------------------------------------------
Last dynamic allocation failure: normal, successful completion
Dynamic entry limit was reached 10318 times

Current static translation entries:
-----------------------------------------
Inside Source Simple:               10
Outside Source Simple:              3
Inside Source Extended:             8
Outside Source Extended:            12

       Dynamic
   Translation Type      Current       Peak     Accumulated    Failed
----------------------  ----------  ----------  -----------  ----------
Inside Source Simple         69999       69999        69999       12568
Outside Source Simple         4518        4518         4518          25
Inside Source Extended       70000       70000        70000         568
Fully Extended               26855       26855        26855        2565


Forwarding statistics for virtual router vr1:
-------------------------------------------------------------------------
Packets received on inside interface and
    forwarded directly             8
    forwarded through translator   111763104
    discarded                      2
    discarded by translator        28524565

Bytes received on inside interface and
    forwarded directly             544
    forwarded through translator   5141098074
```

```
Packets received on outside interface and
    forwarded directly             7
    forwarded through translator   1031624
    discarded                      3
    discarded by translator        578961

Bytes received on outside interface and
    forwarded directly             476
    forwarded through translator   47454704
```

Displaying Translation Entries

The **show ip nat translations** command displays current translations that reside in the translation table.

Simple translation entries appear with inside/outside and local/global address information. Extended entries appear with added protocol and port numbers (or query IDs).

Using verbose mode additionally provides the time since creation and time since last use for each translation entry.

show ip nat translations

- Use to display current translations that reside in the NAT translation table.
- Field descriptions
  - Prot—Protocol (TCP, UDP, ICMP, or GRE) for this translation entry; this field appears only for extended table entries
  - Inside local—Inside local IP address for this translation entry; this field also provides the port number, separated by a colon (:) for extended entries
  - Inside global—Inside global IP address for this translation entry; this field also provides the port number, separated by a colon (:) for extended entries
  - Outside global—Outside global IP address for this translation entry; this field also provides the port number, separated by a colon (:) for extended entries
  - Outside local—Outside local IP address for this translation entry; this field also provides the port number, separated by a colon (:) for extended entries
  - Time since creation—Amount of time elapsed since the translation entry appeared in the translation table
  - Time since last use—Amount of time elapsed since the translation entry was used
- Example 1

```
host1#show ip nat translations
Prot    Inside local      Inside global     Outside global   Outside local
----    ---------------   ---------------   --------------   ----------------
GRE     13.1.2.1:*        20.0.0.1:*            ---              ---
ICMP    13.1.2.2:4        20.0.0.2:4            ---              ---
TCP     13.1.2.3:20       20.0.0.3:50           ---              ---
```

NOTE: Because they are not NAPT translations, port numbers for GRE translations appear as asterisks (*).

- Example 2

```
host1#show ip nat translations verbose
                                                                 Time        Time
                        Inside      Outside     Outside      since       since
Prot Inside local   global      global       local      creation    last use
---- ------------ ----------- ----------- ----------- ------------ ------------
     20.0.0.3     30.0.0.3    ---         ---         00:04:50     00:00:01
     21.0.0.3     30.208.0.3  ---         ---         00:02:12     00:00:01
     21.0.0.4     30.208.0.4  ---         ---         00:02:12     00:00:01
     ---          ---         50.0.0.3    70.0.0.3    00:03:24     Never
     ---          ---         51.0.0.3    70.208.0.3  00:01:44     00:00:01
     ---          ---         51.0.0.4    70.208.0.4  00:01:44     00:00:01
UDP  ---          ---         50.50.0.3:8 70.50.0.3:8 00:03:10     Never
                              7           108
UDP  22.0.0.4:63  30.224.0.3: ---         ---         00:02:12     00:00:01
                  4097
UDP  22.0.0.3:63  30.224.0.3: ---         ---         00:02:12     00:00:01
                  4096
TCP  ---          ---         50.50.0.3:8 70.50.0.3:8 00:03:10     Never
                              0           008
UDP  20.50.0.3:87 30.50.0.3:8 ---         ---         00:03:35     Never
                  108
```

Displaying Address Pool Information

The **show ip nat pool** command displays NAT address pool information. The command output displays configuration (mask and address ranges) of all address pools, unless you supply a specific pool name.

show ip nat pool

- Use to display NAT address pool information.
- Field descriptions
  - pool—Name of the address pool
  - netmask—Network prefix associated with the NAT address pool
  - prefix length—Prefix length associated with the NAT address pool
  - range—Address ranges used by this NAT address pool
- Example 1

```
host1#show ip nat pool

pool: pool1 netmask: 255.255.255.0  prefix length: 24
   range: 3.3.3.1 to 3.3.3.255
   range: 4.4.4.1 to 4.4.4.32

pool: pool2 netmask: 255.255.255.0  prefix length: 24
   range: 1.1.1.1 to 1.1.1.24
   range: 2.2.2.1 to 2.2.2.55
```

- Example 2

```
host1#show ip nat pool pool1
pool: pool1 netmask: 255.255.255.0  prefix length: 24
   range: 3.3.3.1 to 3.3.3.255
   range: 4.4.4.1 to 4.4.4.32
```

Displaying Inside and Outside Rule Settings

The **show ip nat inside rule** and **show ip nat outside rule** commands display access list and pool usage for all dynamic translation rules configured for the virtual router. If you do not specify an access list, the output displays address pool associations for each of the access lists for either inside or outside translation rules in the virtual router. Specifying an access list filters the output to display only the address pool associated with the specified list.

show ip nat inside rule

- Use to display NAT access list and pool usage information for inside source translation rules.
- Field descriptions
  - access list name—Name of the access list
  - pool name—Name of the address pool
  - rule type—Type of rule assigned
- Example

```
host1#show ip nat inside rule
access list name: list1  pool name: poolA rule type: inside source
access list name: list2  pool name: poolB rule type: inside source
access list name: list3  pool name: poolC rule type: inside source overload
```

show ip nat outside rule

- Use to display NAT access list and pool usage information for outside source translation rules.
- Field descriptions
  - access list name—Name of the access list
  - pool name—Name of the address pool
  - rule type—Type of rule assigned
- Example

```
host1#show ip nat outside rule
access list name: list4  pool name: poolD rule type: outside source
```