

Using NAT in Overlapping Networks

Document ID: 13774

Introduction

Prerequisites

- Requirements

- Components Used

- Conventions

Configure

- Network Diagram

- Configurations

Verify

Troubleshoot

Related Information

Introduction

This document demonstrates how you can use Network Address Translation (NAT) for overlapping networks. Overlapping networks result when you assign an IP address to a device on your network that is already legally owned and assigned to a different device on the Internet or outside network. Overlapping networks also result when two companies, both of whom use RFC 1918 IP addresses in their networks, merge. These two networks need to communicate, preferably without having to readdress all their devices.

Prerequisites

Requirements

A basic understanding of IP addressing, IP routing, and Domain Name System (DNS) is helpful for understanding the contents of this document.

Components Used

Support for NAT began in Cisco IOS[®] software version 11.2. For more information on platform support see NAT Frequently Asked Questions.

Conventions

For more information on document conventions, refer to Cisco Technical Tips Conventions.

Configure

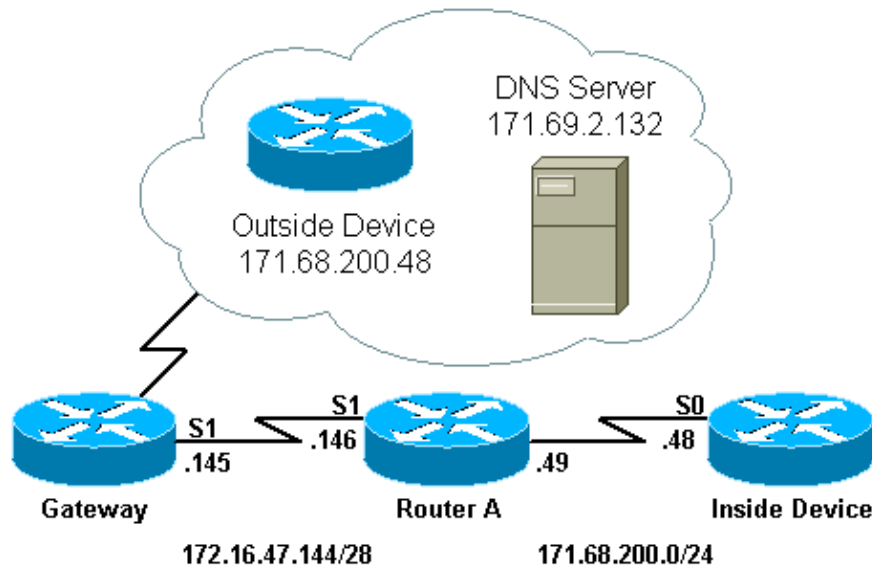
In this section, you are presented with the information to configure the features described in this document.

Note: To find additional information on the commands used in this document, use the Command Lookup Tool (registered customers only) .

Network Diagram

This document uses the network setup shown in the diagram below.

Notice that the inside device has the same IP address as the outside device with which it wishes to communicate.



Configurations

Router A is configured for NAT, such that it translates the inside device to an address from the pool "test-loop" and the outside device to an address from the pool "test-dns." An explanation of how this configuration helps with overlapping follows the configuration table below.

Router A
<pre>! version 11.2 no service udp-small-servers no service tcp-small-servers ! hostname Router-A ! ! ip domain-name cisco.com ip name-server 171.69.2.132 ! interface Loopback0 ip address 1.1.1.1 255.0.0.0 ! interface Ethernet0 ip address 135.135.1.2 255.255.255.0 shutdown ! interface Serial0 ip address 171.68.200.49 255.255.255.0 ip nat inside no ip mroute-cache no ip route-cache no fair-queue ! interface Serial1</pre>

```

ip address 172.16.47.146 255.255.255.240
ip nat outside
no ip mroute-cache
no ip route-cache
!
ip nat pool test-loop 172.16.47.161 172.16.47.165 prefix-length 28
ip nat pool test-dns 172.16.47.177 172.16.47.180 prefix-length 28
ip nat inside source list 7 pool test-loop
ip nat outside source list 7 pool test-dns
ip classless
ip route 0.0.0.0 0.0.0.0 172.16.47.145
access-list 7 permit 171.68.200.0 0.0.0.255
!
!
line con 0
  exec-timeout 0 0
line aux 0
line vty 0 4
  login
!
end

```

In order for the above configuration to help with overlapping when the inside device communicates with the outside device, it must use the outside device's domain name.

The inside device cannot use the IP address of the outside device because it is the same as the address assigned to itself (the inside device). Therefore, the inside device will send a DNS query for the outside device's domain name. The inside device's IP address will be the source of this query, and that address will be translated to an address from the "test-loop" pool because the **ip nat inside source list** command is configured.

The DNS server replies to the address which came from the pool "test-loop" with the IP address associated with the outside device's domain name in the payload of the packet. The destination address of the reply packet is translated back to the inside device's address, and the address in the payload of the reply packet is then translated to an address from the pool "test-dns" because of the **ip nat outside source list** command. Therefore the inside device learns that the IP address for the outside device is one of the addresses from the "test-dns" pool, and it will use this address when communicating with the outside device. The router running NAT takes care of the translations at this point.

This process can be seen in detail in the Troubleshoot section. Devices using overlapping addresses can communicate with each other without the use of DNS, but in this case, static NAT would have to be configured. An example of how this might be done follows.

Router A
<pre> ! version 11.2 no service udp-small-servers no service tcp-small-servers ! hostname Router-A ! ! ip domain-name cisco.com ip name-server 171.69.2.132 ! interface Loopback0 ip address 1.1.1.1 255.0.0.0 ! </pre>

```

interface Ethernet0
 ip address 135.135.1.2 255.255.255.0
 shutdown
!
interface Serial0
 ip address 171.68.200.49 255.255.255.0
 ip nat inside
 no ip mroute-cache
 no ip route-cache
 no fair-queue
!
interface Serial1
 ip address 172.16.47.146 255.255.255.240
 ip nat outside
 no ip mroute-cache
 no ip route-cache
!
ip nat pool test-loop 172.16.47.161 172.16.47.165 prefix-length 28
ip nat inside source list 7 pool test-loop
ip nat outside source static 171.68.200.48 172.16.47.177
ip classless
ip route 0.0.0.0 0.0.0.0 172.16.47.145
ip route 172.16.47.160 255.255.255.240 Serial0

!--- This line is necessary to make NAT work for return traffic.
!--- The router needs to have a route for the pool to the inside
!--- NAT interface so it knows that a translation is needed.

access-list 7 permit 171.68.200.0 0.0.0.255
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
!
end

```

With the above configuration, when the inside device wants to communicate with the outside device it can now use IP address 172.16.47.177, and DNS is not necessary. As shown above, translation of the inside device's address is still done dynamically, which means that the router must get packets from the inside device before a translation is created. For this reason, the inside device must initiate all connections in order for the inside device and outside device to communicate. If it were required that the outside device initiate connections to the inside device, then the address for the inside device must also be statically configured.

Verify

There is currently no verification procedure available for this configuration.

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

The process by which the inside device used DNS to communicate with the outside device, as described above, can be viewed in detail with the following troubleshooting process.

Currently there are no translations in the translation table that can be seen with the **show ip nat translations** command. The examples below use the **debug ip packet** and **debug ip nat** commands instead.

Note: The **debug** commands generate a significant amount of output. Use it only when traffic on the IP network is low, so other activity on the system is not adversely affected.

```
Router-A# show ip nat translations
Router-A# show debug
Generic IP:
  IP packet debugging is on (detailed)
  IP NAT debugging is on
```

When the inside device sends its DNS query to the DNS server, which resides outside the NAT domain, the source address of the DNS query (the address of the inside device) gets translated due to the **ip nat inside** commands. This can be seen in the debug output below.

```
NAT: s=171.68.200.48->172.16.47.161, d=171.69.2.132 [0]
IP: s=172.16.47.161 (Serial0), d=171.69.2.132 (Serial1), g=172.16.47.145, len 66, forward
UDP src=6988, dst=53
```

When the DNS server sends a DNS reply, the payload of the DNS reply gets translated due to the **ip nat outside** commands.

Note: NAT does not look at the payload of the DNS reply unless translation occurs on the IP header of the reply packet. See the **ip nat outside source list 7 pool** command in the router configuration above.

The first NAT message in the debug output below shows that the router recognizes the DNS reply and translates the IP address within the payload to 172.16.47.177. The second NAT message shows the router translating the destination of the DNS reply so that it can forward a reply back to the inside device that performed the initial DNS query. The destination portion of the header, the inside global address, is translated to the inside local address.

The payload of the DNS reply is translated:

```
NAT: DNS resource record 171.68.200.48 -> 172.16.47.177
```

The destination portion of the IP header in the DNS reply packet is translated:

```
NAT: s=171.69.2.132, d=172.16.47.161->171.68.200.48 [65371]
IP: s=171.69.2.132 (Serial1), d=171.68.200.48 (Serial0), g=171.68.200.48, len 315, forward
UDP src=53, dst=6988
```

Let us look at another DNS query and reply:

```
NAT: s=171.68.200.48->172.16.47.161, d=171.69.2.132 [0]
IP: s=172.16.47.161 (Serial0), d=171.69.2.132 (Serial1), g=172.16.47.145, len 66, forward
UDP src=7419, dst=53
NAT: DNS resource record 171.68.200.48 -> 172.16.47.177
NAT: s=171.69.2.132, d=172.16.47.161->171.68.200.48 [65388]
IP: s=171.69.2.132 (Serial1), d=171.68.200.48 (Serial0), g=171.68.200.48, len 315, forward
UDP src=53, dst=7419
```

Now that the payload of the DNS has been translated, our translation table has an entry for the outside local and global addresses of the outside device. With these entries in the table, we can now fully translate the header of the ICMP packets exchanged between the inside device and outside device. Let's look at this exchange in the debug output below.

The following output shows the source address (inside device's address) being translated.

```
NAT: s=171.68.200.48->172.16.47.161, d=172.16.47.177 [406]
```

Here, the destination address (outside device's outside local address) is translated.

```
NAT: s=172.16.47.161, d=172.16.47.177->171.68.200.48 [406]
```

After translation, the IP packet looks like this:

```
IP: s=172.16.47.161 (Serial0), d=171.68.200.48 (Serial1), g=172.16.47.145, len 100, forward
ICMP type=8, code=0
```

The following output shows the source address (outside device's address) being translated on the return packet.

```
NAT*: s=171.68.200.48->172.16.47.177, d=172.16.47.161 [16259]
```

Now the destination address (inside device's global address) of the return packet is translated.

```
NAT*: s=172.16.47.177, d=172.16.47.161->171.68.200.48 [16259]
```

After translation, the return packet looks like this:

```
IP: s=172.16.47.177 (Serial1), d=171.68.200.48 (Serial0), g=171.68.200.48, len 100, forward
ICMP type=0, code=0
```

The exchange of packets continues between the inside device and the outside device.

```
NAT: s=171.68.200.48->172.16.47.161, d=172.16.47.177 [407]
NAT: s=172.16.47.161, d=172.16.47.177->171.68.200.48 [407]
IP: s=172.16.47.161 (Serial0), d=171.68.200.48 (Serial1), g=172.16.47.145, len 100, forward
ICMP type=8, code=0
NAT*: s=171.68.200.48->172.16.47.177, d=172.16.47.161 [16262]
NAT*: s=172.16.47.177, d=172.16.47.161->171.68.200.48 [16262]
IP: s=172.16.47.177 (Serial1), d=171.68.200.48 (Serial0), g=171.68.200.48, len 100, forward
ICMP type=0, code=0
NAT: s=171.68.200.48->172.16.47.161, d=172.16.47.177 [408]
NAT: s=172.16.47.161, d=172.16.47.177->171.68.200.48 [408]
IP: s=172.16.47.161 (Serial0), d=171.68.200.48 (Serial1), g=172.16.47.145, len 100, forward
ICMP type=8, code=0
NAT*: s=171.68.200.48->172.16.47.177, d=172.16.47.161 [16267]
NAT*: s=172.16.47.177, d=172.16.47.161->171.68.200.48 [16267]
IP: s=172.16.47.177 (Serial1), d=171.68.200.48 (Serial0), g=171.68.200.48, len 100, forward
ICMP type=0, code=0
NAT: s=171.68.200.48->172.16.47.161, d=172.16.47.177 [409]
NAT: s=172.16.47.161, d=172.16.47.177->171.68.200.48 [409]
IP: s=172.16.47.161 (Serial0), d=171.68.200.48 (Serial1), g=172.16.47.145, len 100, forward
ICMP type=8, code=0
NAT*: s=171.68.200.48->172.16.47.177, d=172.16.47.161 [16273]
NAT*: s=172.16.47.177, d=172.16.47.161->171.68.200.48 [16273]
IP: s=172.16.47.177 (Serial1), d=171.68.200.48 (Serial0), g=171.68.200.48, len 100, forward
ICMP type=0, code=0
NAT: s=171.68.200.48->172.16.47.161, d=172.16.47.177 [410]
NAT: s=172.16.47.161, d=172.16.47.177->171.68.200.48 [410]
IP: s=172.16.47.161 (Serial0), d=171.68.200.48 (Serial1), g=172.16.47.145, len 100, forward
ICMP type=8, code=0
NAT*: s=171.68.200.48->172.16.47.177, d=172.16.47.161 [16277]
NAT*: s=172.16.47.177, d=172.16.47.161->171.68.200.48 [16277]
IP: s=172.16.47.177 (Serial1), d=171.68.200.48 (Serial0), g=171.68.200.48, len 100, forward
ICMP type=0, code=0
```

Once the exchange of packets between outside and inside is complete, we can look at the translation table, which has three entries. The first entry was created when the inside device sent a DNS query. The second entry was created when the payload of the DNS reply was translated. The third entry was created when the

ping was exchanged between the inside device and the outside device. The third entry is a summary of the first two entries, and is used for more efficient translations.

```
Router-A# show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 172.16.47.161      171.68.200.48    ---
--- ---                ---                172.16.47.177     171.68.200.48
--- 172.16.47.161      171.68.200.48    172.16.47.177     171.68.200.48
```

It is important to note when you are trying to establish connectivity between two overlapping networks by running dynamic NAT on a single Cisco router, you must use DNS to create an outside local to outside global translation. If you do not use DNS, connectivity can be established with static NAT, but it is more difficult to manage.

Related Information

- [NAT Support Page](#)
- [Technical Support – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Aug 10, 2005

Document ID: 13774
