CCNA Exam Prep: Back to Networking Basics with Hank Preston and Patrick Gargano -- Season 2

Navigating NAT: Bridging Private Networks to the Internet

Network Address Translation (NAT) is the process where a network device, such as a Cisco router or Cisco firewall, assigns a public address to host devices inside a private network. The main reason to use NAT is to reduce the number of public IP addresses that an organization uses because the number of available IPv4 public addresses is limited. In this lab, you will configure various types of NAT. You will test, view, and verify that the translations are taking place, and you will interpret the NAT/PAT statistics to monitor the process.

Related CCNA v1.1 exam topic:

- 4.1 Configure and verify inside source NAT using static and pools

In this lab, we will explore how to:

- Configure and verify dynamic NAT for IPv4
- Configure and verify PAT for IPv4
- Configure and verify static NAT for IPv4

Setup and Scenario

In this set of lab-based demonstrations, you are a network engineer tasked with exploring and testing different types of NAT deployments that your enterprise network could use.

You've been asked to:

- Configure the EDGE router with dynamic NAT, PAT, and static NAT.

*Be sure to **START** the lab before continuing to the demo lab.*

> Note: The credentials for all devices are **cisco / cisco**

Part 1: Reviewing the current state of the network

Before we jump into configuring NAT, let's check the current state of the network and how it's operating.

Step 1

Open a console connection to the EDGE router and verify its configuration.

```
EDGE# show run | section dhcp
ip dhcp excluded-address 192.168.10.1 192.168.10.5
ip dhcp pool VLAN10
 network 192.168.10.0 255.255.255.0
 default-router 192.168.10.1
 dns-server 192.168.255.1
 domain-name example.com

EDGE# show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
```

```
Ethernet0/0            209.165.200.6   YES TFTP   up                    up
Ethernet0/1            192.168.10.1    YES manual up                    up
Ethernet0/2            192.168.30.1    YES manual up                    up
Ethernet0/3            unassigned      YES TFTP   administratively down down
Loopback0              192.168.255.1   YES TFTP   up                    up

EDGE# sh run | section ip host | dns
 dns-server 192.168.255.1
ip host server.example.com 192.168.30.30
ip host www_server.example.com 209.165.201.10
ip dns server
ip dns primary example.com soa 192.168.255.1 admin@example.com 21600 900 7776000 86400

EDGE# show ip route static
<...output omitted...>

Gateway of last resort is 209.165.200.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 209.165.200.1
```

Notice that the EDGE router is configured as a DHCP server for the 192.168.10.0/24 network. EDGE is also configured as the primary DNS server for the example.com domain, using its Loopback0 IPv4 address as the DNS server address. EDGE is already configured with a default route pointing to the ISP router to ensure that all traffic to unknown destinations is sent to the ISP router. ### Step 2 Verify internal connectivity. From PC1, ping the EDGE Loopback0 interface, ping the internal SERVER, ping the ISP router, and ping the external WWW_SERVER.

```
PC1:~$ ping 192.168.255.1
PING 192.168.255.1 (192.168.255.1): 56 data bytes
64 bytes from 192.168.255.1: seq=0 ttl=42 time=1.333 ms
64 bytes from 192.168.255.1: seq=1 ttl=42 time=1.172 ms
^C

PC1:~$ ping server
PING server (192.168.30.30): 56 data bytes
64 bytes from 192.168.30.30: seq=0 ttl=42 time=1.340 ms
64 bytes from 192.168.30.30: seq=1 ttl=42 time=1.547 ms
^C

PC1:~$ ping 209.165.200.1
PING 209.165.200.1 (209.165.200.1): 56 data bytes
^C
--- 209.165.200.1 ping statistics ---
9 packets transmitted, 0 packets received, 100% packet loss

PC1:~$ ping 209.165.201.10
PING 209.165.201.10 (209.165.201.10): 56 data bytes
^C
--- 209.165.201.10 ping statistics ---
6 packets transmitted, 0 packets received, 100% packet loss
PC1:~$
```

The first two test will succeed. Notice in the second test the use of the DNS name for the internal SERVER (192.168.30.30). The last two test will fail since the ISP router does not know how to route packets back to the source internal 192.168.10.0/24 network. We will correct this in the next part of the lab.

Part 2: Configure and verify dynamic NAT for IPv4 on EDGE

An ISP has allocated the public IP address space of 209.165.200.0/29 to your company. This network is used to address the link between the ISP router and the company EDGE router. The first address (209.165.200.1) is assigned to the E0/0 interface on ISP and the last address (209.165.200.6) is assigned to the E0/0 interface on EDGE. The remaining addresses (209.165.200.2-5) will be used to provide internet access to the company hosts.

Step 1

Configure a simple access list that defines what hosts are going to be allowed for translation. In this case, all devices on the 192.168.10.0/24 LAN and 192.168.30.0/24 LAN are eligible for translation.

```
EDGE(config)# access-list 1 permit 192.168.10.0 0.0.0.255
EDGE(config)# access-list 1 permit 192.168.30.0 0.0.0.255
```

Step 2

Create the NAT pool, and give it a name and a range of addresses to use. Use the first three available addresses from the ISP range.

```
EDGE(config)# ip nat pool MYPOOL 209.165.200.2 209.165.200.4 netmask 255.255.255.248
```

The pool contains three public IP addresses. ### Step 3 Configure the translation, associating the ACL and pool to the translation process.

```
EDGE(config)# ip nat inside source list 1 pool MYPOOL
```

This command enables inside NAT for addresses matching ACL 1. ### Step 4 Define the inside and outside interfaces on EDGE.

```
EDGE(config)# interface e0/1
EDGE(config-if)# ip nat inside
EDGE(config-if)# interface e0/2
EDGE(config-if)# ip nat inside
EDGE(config-if)# interface e0/0
EDGE(config-if)# ip nat outside
```

These commands tell the NAT process which interfaces are private (inside) and which are public (outside). ### Step 5 From PC1, ping the ISP E0/0 interface (209.165.200.1) and ping the external WWW_SERVER. On EDGE, display the NAT translation table by using the ``show ip nat translations`` command.

```
PC1:~$ ping 209.165.200.1
PING 209.165.200.1 (209.165.200.1): 56 data bytes
64 bytes from 209.165.200.1: seq=1 ttl=42 time=1.665 ms
64 bytes from 209.165.200.1: seq=2 ttl=42 time=1.405 ms

PC1:~$ ping www_server
PING www_server (209.165.201.10): 56 data bytes
64 bytes from 209.165.201.10: seq=0 ttl=42 time=1.956 ms
64 bytes from 209.165.201.10: seq=1 ttl=42 time=2.274 ms

EDGE# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 209.165.200.2:14  192.168.10.6:14    209.165.201.10:14  209.165.201.10:14
icmp 209.165.200.2:15  192.168.10.6:15    209.165.200.1:15   209.165.200.1:15
--- 209.165.200.2      192.168.10.6       ---                ---
```

Notice the two ICMP entries, one for the ISP 209.165.200.1 address and one for the WWW_SERVER 209.165.201.10 address, as well as a generic entry indicating the mapping of the PC1 IP address to the first available address from the pool. The EDGE router has a DNS entry for the external WWW_SERVER which allows you to ping that device more easily. The output from the ``show ip nat translations`` command displays four types of addresses: *Inside local address*: The IP address assigned to a host on the inside network. *Inside global address*: The translated inside local address. *Outside global address*: The IPv4 address that the host owner assigns to a host on the outside network. *Outside local address*: The IPv4 address of an outside host as it appears to the inside network. ### Step 6 From the internal SERVER, ping the ISP E0/0 interface (209.165.200.1) and ping the external WWW_SERVER. On EDGE, display the NAT translation table by using the ``show ip nat translations`` command.

```
SERVER:~$ ping 209.165.201.1
PING 209.165.201.1 (209.165.201.1): 56 data bytes
64 bytes from 209.165.201.1: seq=0 ttl=42 time=1.665 ms
64 bytes from 209.165.201.1: seq=1 ttl=42 time=2.114 ms

SERVER:~$ ping www_server
PING www_server (209.165.201.10): 56 data bytes
64 bytes from 209.165.201.10: seq=0 ttl=42 time=2.303 ms
64 bytes from 209.165.201.10: seq=1 ttl=42 time=2.299 ms

EDGE# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 209.165.200.2:16  192.168.10.6:16    209.165.200.1:16   209.165.200.1:16
icmp 209.165.200.2:17  192.168.10.6:17    209.165.201.10:17  209.165.201.10:17
--- 209.165.200.2      192.168.10.6       ---                ---
icmp 209.165.200.3:1   192.168.30.30:1    209.165.201.10:1   209.165.201.10:1
icmp 209.165.200.3:2   192.168.30.30:2    209.165.201.1:2    209.165.201.1:2
--- 209.165.200.3      192.168.30.30      ---                ---
```

You should have a total of four ICMP translations. Two triggered by traffic from PC1, and two triggered by traffic from the internal SERVER. You will also see two generic NAT entries, one for the PC1 mapping and one for the internal SERVER mapping. ### Step 7 To verify how long these translations are allocated, issue the ``show ip nat translations verbose`` command.

```
EDGE#sh ip nat tr ver
Pro Inside global      Inside local       Outside local      Outside global
--- 209.165.200.2      192.168.10.6       ---                ---
    create 00:28:18, use 00:05:30, left 23:54:29, Map-Id(In): 2,
    flags:
none, use_count: 0, entry-id: 1, lc_entries: 0
--- 209.165.200.3      192.168.30.30      ---                ---
    create 00:20:31, use 00:01:19, left 23:58:40, Map-Id(In): 2,
    flags:
none, use_count: 0, entry-id: 6, lc_entries: 0
```

The output shows that the NAT allocations are for 24 hours. ### Step 8 Enable the ``debug ip nat`` command on the EDGE router and ping from PC1 to the external WWW_SERVER to observe the translations that take place.

```
EDGE#
*Mar  3 16:54:04.365: NAT: Entry assigned id 38
*Mar  3 16:54:04.365: NAT*: s=192.168.10.6->209.165.200.2, d=209.165.201.10 [61076]
*Mar  3 16:54:04.367: NAT*: s=209.165.201.10, d=209.165.200.2->192.168.10.6 [10157]
*Mar  3 16:54:05.365: NAT*: s=192.168.10.6->209.165.200.2, d=209.165.201.10 [61119]
EDGE#
*Mar  3 16:54:05.366: NAT*: s=209.165.201.10, d=209.165.200.2->192.168.10.6 [10177]
*Mar  3 16:54:06.365: NAT*: s=192.168.10.6->209.165.200.2, d=209.165.201.10 [61169]
EDGE#
*Mar  3 16:54:06.367: NAT*: s=209.165.201.10, d=209.165.200.2->192.168.10.6 [10234]
EDGE#
*Mar  3 16:54:18.320: NAT: expiring 209.165.200.2 (192.168.10.6) icmp 25 (25)
*Mar  3 16:54:18.320: NAT: Freeing nat entry, id 37
```

Look for the -> symbol in each NAT translation line. This shows when NAT is occurring, either as traffic is leaving the private network, or as traffic is returning. You will get a pair of debug entries for each ping transmitted to the WWW_SERVER since a ping is two messages: echo request and echo reply. After a few moments you will see the NAT entry expiring.

Disable NAT debugging with the `no debug ip nat` command.

Step 9

A pool of three addresses isn't sufficient for an enterprise network with thousands of hosts. Clear the NAT translations and statistics and configure PAT in the next part of the lab.

```
EDGE# clear ip nat translations *

EDGE# clear ip nat statistics
```

Part 3: Configure and verify PAT for IPv4 on EDGE

Dynamic NAT with a pool is useful when the number of host devices is small, but that solution doesn't scale for large enterprise networks. Instead, you will deploy PAT, also called NAT Overload. The configuration steps are basically the same for NAT and PAT. An access list identifies addresses eligible to be translated, interfaces are identified as either inside or outside, and a pool is defined or an interface is chosen to "overload". In Part 3, we will start by enable PAT for the pool used in Part 2, and then we will migrate to interface PAT.

Step 1

Remove the NAT pool command on EDGE and then enter it again using the `overload` keyword.

```
EDGE(config)# no ip nat inside source list 1 pool MYPOOL
EDGE(config)# ip nat inside source list 1 pool MYPOOL overload
```

This is the only change required to enable PAT translations, since the access list and the pool are already configured. ### Step 2 From PC1 and the internal SERVER, ping the external WWW_SERVER. USe the ``show ip nat translations`` command to verify the results.

```
PC1:~$ ping www_server
PING www_server (209.165.201.10): 56 data bytes
64 bytes from 209.165.201.10: seq=0 ttl=42 time=3.815 ms
64 bytes from 209.165.201.10: seq=1 ttl=42 time=2.086 ms

SERVER:~$ ping www_server
PING www_server (209.165.201.10): 56 data bytes
64 bytes from 209.165.201.10: seq=0 ttl=42 time=1.810 ms
64 bytes from 209.165.201.10: seq=1 ttl=42 time=2.226 ms

EDGE# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 209.165.200.4:1024 192.168.10.6:18   209.165.201.10:18  209.165.201.10:1024
icmp 209.165.200.4:1025 192.168.30.30:4   209.165.201.10:4   209.165.201.10:1025
```

Notice that the inside global address is identical for both PC1 and internal SERVER traffic. In the output above, the EDGE router used the last address in the pool. Your output might differ. The NAT process is now "overloading" one public address from the pool. This means that NAT is using Layer 4 port numbers to create unique NAT entries for each host IPv4 address being translated. Generate more traffic from PC1 and the internal SERVER to the external WWW_SERVER and ISP router to verify that the EDGE router continues to overload the same public IP address. If you quickly verify the output from the ``show ip nat translations verbose`` command you will notice that NAT entries for ICMP now timeout after 1 minute, as compared to 24 hours in Part 2. ### Step 3 From PC1, SSH to the external WWW_SERVER and then verify the NAT translation table.

```
PC1:~$ ssh www_server
The authenticity of host 'www_server (209.165.201.10)' can't be established.
ED25519 key fingerprint is SHA256:9sjYtoCKDaHJGYMCz38aLjFu4aLBlgJzdul3HwyTMlU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'www_server' (ED25519) to the list of known hosts.
```

```
cisco@www_server's password: cisco
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See .

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

WWW_SERVER:~$

EDGE# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 209.165.200.4:1024 192.168.10.6:50812 209.165.201.10:22  209.165.201.10:22
```

When prompted to continue, answer `yes` to trust the connection. The password for the external WWW_SERVER is `cisco`.

Once the SSH connection established, you should see a TCP entry in the NAT table pointing to port 22 of the outside 209.165.201.10 address. If you verify the `verbose` output, you will see that a TCP connection timeout is 24 hours.

On PC1, type `exit` to terminate the SSH session.

Step 4

PAT to a pool is a very effective solution for small-to-midsize organizations. However, there could be unused IPv4 addresses in this scenario. In the next step you will enable PAT with interface overload to eliminate this waste of IPv4 addresses. Clear translations and translation statistics in preparation.

```
EDGE# clear ip nat translations *
EDGE# clear ip nat statistics
```

Step 5

For interface PAT, we still need the access list and inside and outside interfaces, but we can remove the NAT overload command and the NAT pool.

```
EDGE(config)# no ip nat inside source list 1 pool MYPOOL overload
EDGE(config)# no ip nat pool MYPOOL
```

Step 6

Add the PAT command to overload the EDGE E0/0 outside interface.

```
EDGE(config)# ip nat inside source list 1 interface E0/0 overload
```

Step 7

From PC1 and the internal SERVER, ping the external WWW_SERVER and the ISP. Verify the NAT translation table to confirm that the PAT feature is working.

```
EDGE# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 209.165.200.6:1024 192.168.10.6:3   209.165.201.1:3     209.165.201.1:1024
icmp 209.165.200.6:1025 192.168.10.6:4   209.165.201.10:4    209.165.201.10:1025
icmp 209.165.200.6:1026 192.168.30.30:7   209.165.201.1:7    209.165.201.1:1026
icmp 209.165.200.6:1027 192.168.30.30:8   209.165.201.10:8   209.165.201.10:1027
```

Notice that the inside global address 209.165.200.6 is the same for every entry and represents the IPv4 address of the E0/0 interface on the EDGE router that is being "overloaded". Quickly verify the `verbose` output of the command to confirm that the timeout value is 1 minute for ICMP traffic. Similar to PAT with a pool, interface PAT also uses Layer 4 port numbers to create unique entries for each internal host.

Part 4: Configure and verify static NAT for IPv4 on EDGE

In Part 4, you will configure static NAT so that the internal SERVER is directly reachable from the internet. The internal SERVER will be reachable from the EDGE router via the address 209.165.200.5.

Step 1

On the EDGE router, clear the current NAT translations and statistics.

```
EDGE# clear ip nat translations *
EDGE# clear ip nat statistics
```

Step2

On the EDGE router, configure a static NAT mapping between the 192.168.30.30 private IPv4 address of the internal SERVER and the 209.165.200.5 public address that is part of the range of addresses provided by the ISP.

```
EDGE(config)# ip nat inside source static 192.168.30.30 209.165.200.5
```

Step 3

On the EDGE router, verify that the static NAT entry is working. Use the `show ip nat translations` command.

```
EDGE# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 209.165.200.5      192.168.30.30      ---                ---
```

The translation table shows the static translation is in effect. Verify this by pinging from the external WWW_SERVER to 209.165.200.5. The pings should work.

```
WWW_SERVER:~$ ping 209.165.200.5
PING 209.165.200.5 (209.165.200.5): 56 data bytes
64 bytes from 209.165.200.5: seq=0 ttl=42 time=3.375 ms
64 bytes from 209.165.200.5: seq=1 ttl=42 time=2.782 ms
```

Verify the NAT translation table again.

```
EDGE# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 209.165.200.5:1   192.168.30.30:1    209.165.201.10:1   209.165.201.10:1
--- 209.165.200.5      192.168.30.30      ---                ---
```

You will see a new ICMP entry triggered by the successful ping from the 209.165.201.10 device to the 209.165.200.5 inside global address that then gets translated to the 192.168.30.30 address. The `verbose` output of the command will show that these entries have a timeout of 1 minute.

Perform the same SSH test from Part 3, but this time initiate the SSH session from the external WWW_SERVER towards the internal private SERVER. Accept the certificate and log into the internal SERVER. Then, verify the

NAT translation table. You will see a TCP entry pointing to destination port 22.

Congratulations! You have completed the lab. You learned how to configure and verify dynamic NAT, PAT with a pool, interface PAT, and static NAT.