Deploying Network Address Translation

Session 2212
Agenda—Terminology

- Terminology Rehash
- Requirements (Hardware/Software)
- Considerations
- Configuration/Basic to Real World Examples
- Troubleshooting

Terminology—Inside

ZONE “Inside”: Intranet/Private Address

- Your company’s network
- Typically a RFC 1918 network
- “Local address” is the real IP address of the host
- Not routable on the Internet
ZONE “Outside”: Internet/Public Address

- Everyone else’s network
- Registered addresses only
- “Global address” is the virtual IP address of the inside host
- Is routable on the Internet

Terminology—Static

- Commonly used for inbound traffic
- Permanent
- “Local” address is always known by the same “global” address
Typically used for outbound (inside -> outside) traffic

Short lived

”Local” address might not always be known by the same “global” address

Network Address Translation

Layer 3

Maps one internal (local) address to one external (global) address
Terminology—PAT

- Port Address Translation
- Layer 3 and 4
- Similar to NAT, except it maps multiple internal (local) addresses to one external (global) address

Agenda—Requirements

- Terminology Rehash
- Requirements (Hardware/Software)
- Considerations
- Configuration/Basic to Real World Examples
- Troubleshooting
Cisco IOS Software

- 11.2—IP plus only
- 11.3—PAT: General availability
- 11.3—NAT: IP plus
- 12.x—Full NAT/PAT

Requirements—Hardware

Hardware

- Most platforms
- Each translation = 160 bytes
- 10,000 translation = 1.6 megabytes
- Performance/latency is negligible
Agenda—Considerations

- Terminology Rehash
- Requirements (Hardware/Software)
- Considerations
- Configuration/Basic to Real World Examples
- Troubleshooting

Considerations—Applications

Know Your Applications

- Application Layer: Embedded IP information in the payload
- Transport/Network Layer: PAT/NAT compliant
Considerations—Embedded IP

Address Translation

Inside

Outside

Considerations—IPSec-ESP

Encapsulating Security Payload (ESP): Tunnel Mode Only

Original Packet

WORKS!
Considerations—IPSec-AH

Authentication Headers (AH)
Transport Mode

Authenticated

HDR + DATA = Checksum

IP_HDR
DATA

NAT
Layer 3
Checksum

Breaks!

Considerations—Access-Lists Inbound

Inside
Outside

Packet Flow

Routing
Inbound ACL
Inbound ACL*

NAT
Decryption

Outbound ACL

*Only if the Packet is encrypted
Considerations—Access-Lists
Outbound

Packet Flow

Inside          Outside

Inbound ACL   Routing   Encryption

Policy Routing  NAT  Outbound ACL

FTP—Active

- Server initiated Data Connections
- Client tells the server on which Port to send to the Client

TCP Connection 1: Active Mode: LS Set
TCP Connection 2: Active Mode: Data

Control Connection

SYN
SYN and ACK
ACK

"Inside" Network
"Outside" Network

Port Command <Address and Port>

ACK
SYN
SYN and ACK
ACK

Data Flows Server to Client

Translate Embedded Address—Reserve Source Port, IF PAT is to be used
FTP—Passive

- Client initiates data connections
- Server tells the client on which port to send to the client

TCP Connection 1: Control Connection
- SYN
- SYN and ACK
- ACK
- PASV (Passive)

TCP Connection 2: Passive Mode: Data
- SYN
- SYN and ACK
- ACK
- Data Flows Server to Client

“Inside” Network
- ENT PASV <Address and Port>
- Translate Embedded Address

“Outside” Network

Non Standard FTP Ports

- Server (172.16.1.1) is listening on port 6000

```
router(config)# access-list 1 permit host 172.16.1.1
router(config)# ip nat service list 1 ftp tcp port 6000
```
Agenda—Configurations

- Terminology Rehash
- Requirements (Hardware/Software)
- Considerations
- Configuration/Basic to Real World Examples
- Troubleshooting

Topology—Outbound NAT

ISP Assigned: 209.165.201.0 /27
Step 1:
Create the Pool of usable Global Addresses

```
router(config)# ip nat pool natpool 209.165.201.10
209.165.201.30 netmask 255.255.255.224
```

Step 2:
Define the Local Addresses

```
router(config)# access-list 1 permit ip 10.1.1.0 0.0.0.255
```
Step 3:
Bind Local Addresses to Global Addresses

```
router(config)# ip nat inside source list 1 pool natpool
```

Or Step 3:
Bind Local Addresses to Outside Interface

```
router(config)# ip nat inside source list 1 interface serial 0 overload
```
Or even Step 3:
Bind Local Addresses to Global Addresses

```
router(config)# ip nat inside source list 1 pool natpool overload
```

---

Step 4:
Apply NAT reference points

```
router(config)# interface ethernet0
router(config-if)# ip nat inside
router(config-if)# interface serial 0
router(config-if)# ip nat outside
```
Inbound NAT

Internet needs to get to 10.1.1.10:
Build a static translations

```
router(config)# ip nat inside source static
10.1.1.10 209.165.201.5
```

Topology—NAT by Destination
NAT by Destination—to Partners

192.168.1.0/24

Available Addresses:
172.16.1.0/24

router(config)# access-list 110 permit
ip 10.0.0.0 0.255.255.255
192.168.1.0 0.0.0.255

NAT

Ethernet 0

Serial 0

Partners

router(config)# ip nat pool
partners 172.16.1.3
172.16.1.254 netmask
255.255.255.0

NAT by Destination—to Internet

router(config)# ip nat pool internet
209.165.201.10 209.165.201.30
netmask 255.255.255.224

Internet

Available Addresses:
209.165.201.0/27

router(config)# access-list 100 deny
ip 10.0.0.0 0.255.255.255
192.168.1.0 0.0.0.255

router(config)# access-list 100 permit
ip 10.0.0.0 0.255.255.255
any
NAT by Destination—Route Map Declaration

router(config)# route-map
topartner permit 10
topartner pool partners

router(config-map)# match
ip address 110

router(config)# route-map
tointernet permit 10
tointernet pool internet

router(config-map)# match
ip address 100

NAT by Destination—Bindings

router(config)# ip nat inside
source route-map
topartner pool partners

router(config)# ip nat inside
source route-map
tointernet pool internet
Translating—Simple vs. Extended

**SIMPLE using access-lists**

```
Router#show ip nat translation
Pro Inside global          Inside local          Outside local      Outside global
--- 172.16.4.1              10.1.1.1                ---                       ---                 --- 172.16.4.1              10.1.1.1                ---                       ---
```

**EXTENDED using route-maps**

```
Router#show ip nat translation
Pro Inside global          Inside local          Outside local      Outside global
tcp 172.16.4.1:11012   10.1.1.1:11012     172.17.1.1:23      172.17.1.1:23
tcp 172.16.3.1:11011   10.1.1.1:11011     172.16.1.1:23      172.16.1.1:23
```

VPNs—The Issues

```
VPN Gateway
ISP
Internet
IPSec Tunnel
Your Company
NAT:
Access-list deny ip
10.0.0.0 0.255.255.255
??
VPN Gateway

Mode Config Pool: 172.16.1.1-.254

Your Company

10.0.0.0/8

access-list 100 deny
ip 10.0.0.0 0.255.255.255
172.16.1.0 0.0.0.255

access-list 100 permit
ip 10.0.0.0 0.255.255.255
any

VPNs—Mode Configuration

VPNs—Static NATs

ISP

NAT:
ip nat inside source
static 10.1.1.1
209.165.201.5

Roaming User

NAT:
10.1.1.1/8

Internet

VPN Gateway

IPSec Tunnel

IPSec Tunnel

ISP

NAT:
VPNs—Policy Routing

```
router (config)# access-list 100
  permit ip 10.0.0.0 0.255.255.255
  172.16.1.0 0.0.0.255
router(config)# route-map bypassnat permit 10
router(config-map)# match ip address 100
router(config-map)# set ip next-hop 172.31.1.2
router(config)# interface Ethernet 0
router(config-if)# ip policy route-map bypassnat
```

TCP Load Balancing

```
router(config)# ip nat pool tcpload 10.1.1.1 10.1.1.3
  netmask 255.255.255.0 type rotary
router(config)# access-list 1 permit host 209.165.201.5
router(config)# ip nat inside destination list 1 pool tcpload
```
Overlapping Networks

```
router-nat(config)# ip nat outside source static network 192.168.1.0 10.1.1.0 /24
router-nat(config)# ip nat inside source static network 10.1.1.0 172.16.1.0/24
```

Overlapping Networks—DNS Query

```python
RED-3
```
Overlapping Networks—
DNS Response

RED-3
192.168.1.3

Response = 10.1.1.3

DNS Response modified via address translation

Overlapping Networks—
The Packet Is Sent

SA:10.1.1.3
DA:192.168.1.3
Overlapping Networks—Source Translation

SA: 172.16.1.3
DA: 192.168.1.3

Overlapping Networks—Destination Translation

SA: 172.16.1.3
DA: 10.1.1.3
**Network Statics**

Network Statics—Cisco IOS Syntax

Router(config)# ip nat inside source static network 10.1.1.0 172.18.1.0 /24 no-alias

-OR-

drouter(config)# ip nat pool natpool 172.18.1.0 172.18.1.255 netmask 255.255.255.0 type match-host

drouter(config)# ip nat inside source list 1 pool natpool

SA= 10.1.1.20 -> 172.18.1.20

SA= 10.1.1.10 -> 172.18.1.10
Network Statics—Show Commands

```
router#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 172.18.1.10        10.1.1.10            ---                       ------ 172.18.1.10        10.1.1.10            ---                       ---
Subnet translation:
Inside global   Inside local    Outside local   Outside global /prefix
172.18.1.0      10.1.1.0          ---                     ---                   /24 172.18.1.0      10.1.1.0          ---                     ---                   /24
```

Network Statics—Debugs

```
router#debug ip nat detailed
IP NAT detailed debugging is on
router#
00:12:30: NAT: i: icmp (10.1.1.10, 2458) -> (10.1.2.2, 2458) [20]
00:12:30: NAT: Create inside host entry from network translation:
00:12:30:   10.1.1.10 -> 172.18.1.10 (10.1.1.0 -> 172.18.1.0)
00:12:30: NAT*: o: icmp (10.1.2.2, 2458) -> (172.18.1.10, 2458) [20]
```
Agenda—Troubleshooting

- Terminology Rehash
- Requirements (Hardware/Software)
- Considerations
- Configuration/Basic to Real World Examples
- Troubleshooting

Show Commands—Translations

```
show ip nat translation

router# show ip nat translation
Pro Inside global          Inside local          Outside local      Outside global
tcp 172.16.4.1:11012       10.1.1.1:11012     172.17.1.1:23      172.17.1.1:23
tcp 172.16.3.1:11011       10.1.1.1:11011     172.16.1.1:23      172.16.1.1:23
```
Show Commands—Statistics

**show ip nat statistics**

```python
cisco# show ip nat statistics
Total active translations: 1 (0 static, 1 dynamic; 0 extended)
Outside interfaces:
  Ethernet1
Inside interfaces:
  Ethernet0
Hits: 191  Misses: 9
Expired translations: 1
Dynamic mappings:
  -- Inside Source
  access-list 1 pool natpool refcount 1
  pool natpool: netmask 255.255.255.0
  start 172.16.2.1 end 172.16.2.10
  type generic, total addresses 10, allocated 1 (10%), misses 0
```

Global Commands—Time Out

**ip nat translation**

```python
cisco(config)# ip nat translation
  dns-timeout Specify timeout for NAT DNS flows
  finrst-timeout Specify timeout for NAT TCP flows after a FIN or RST
  icmp-timeout Specify timeout for NAT ICMP flows
  max-entries Specify maximum number of NAT entries
  port-timeout Specify timeout for NAT TCP/UDP port specific flows
  syn-timeout Specify timeout for NAT TCP flows after a SYN and no further data
  tcp-timeout Specify timeout for NAT TCP flows
  timeout Specify timeout for dynamic NAT translations
  udp-timeout Specify timeout for NAT UDP flows
```
Debug Example—Working Translation

debug ip nat

00:01:54: NAT: s=10.1.1.1->172.16.2.1, d=172.16.1.1 [0]
00:01:58: NAT: s=172.16.1.1, d=172.16.2.1->10.1.1.1 [0]
00:01:58: NAT: s=10.1.1.1->172.16.2.1, d=172.16.1.1 [1]
00:01:58: NAT*: s=172.16.1.1, d=172.16.2.1->10.1.1.1 [1]

debug ip nat detailed

00:03:18: NAT: i: tcp (10.1.1.1, 11018) -> (172.16.1.1, 23) [0]
00:03:18: NAT: o: tcp (172.16.1.1, 23) -> (172.16.2.1, 11018) [0]
00:03:18: NAT: i: tcp (10.1.1.1, 11018) -> (172.16.1.1, 23) [1]
00:03:18: NAT*: o: tcp (172.16.1.1, 23) -> (172.16.2.1, 11018) [1]

Debug Example—Non-Working Translation

router#debug ip nat
IP NAT debugging is on
router#debug ip nat detailed
IP NAT detailed debugging is on
router#
NAT: i: tcp (172.16.1.2, 11010) -> (172.18.1.2, 23) [0]
NAT: failed to allocate address for 172.16.1.2, list/map 1
NAT: translation failed (A), dropping packet s=172.16.1.2 d=172.18.1.2
NAT: o: icmp (172.16.1.1, 23) -> (172.16.1.2, 11010) [4]

(A)—Means the packet was dropped after the translation
Clear Commands

- clear ip nat translation
- clear ip nat statistics

Summary

- NAT/PAT (overload) -> one-to-one/many-to-one address mappings
- Can solve IP address shortages and/or conflicts
- Can hide your network address space from the “OUTSIDE” world
- Is flexible by utilizing route-maps and access-lists to determine what traffic needs to be translated.
- Only is performed if the packet traverses from the INSIDE to OUTSIDE “ip nat” interfaces and is permitted via the access-list
Deploying Network Address Translation

Session 2212

Please Complete Your Evaluation Form

Session 2212