Traduction d'adresses de réseau sur une barrette

Contenu

Introduction

Que voulons-nous dire par Traduction d'adresses de réseau (NAT) sur une barrette ? Le terme « sur une barrette » implique habituellement l'utilisation d'une interface physique unique d'un routeur pour une tâche. Tout comme nous pouvons employer des sous-interfaces de la même interface physique pour exécuter l'agrégation Inter-Switch Link (ISL), nous pouvons employer une interface physique unique sur un routeur afin d'accomplir la NAT.

Remarque: Le routeur doit commuter par processus chaque paquet en raison de l'interface de bouclage. Ceci dégrade les performances du routeur.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Composants utilisés

Cette caractéristique exige de vous d'utiliser une version du logiciel de Cisco IOS® qui prend en charge NAT. Utilisez le Navigateur de fonctionnalités Cisco II (clients inscrits seulement) pour déterminer les versions d'IOS que vous pouvez utiliser avec cette configuration.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous aux Conventions relatives aux conseils techniques Cisco.

Informations générales

Pour que la NAT ait lieu, un paquet doit être commuté d'une interface définie « intérieur » NAT à une interface définie « extérieur » NAT, ou vice-versa. Cette condition requise pour NAT n'a pas changé, mais ce document explique comment vous pouvez utiliser une interface virtuelle, également appelée interface de bouclage, et le Policy Based Routing (PBR) pour que la NAT fonctionne sur un routeur avec une interface physique unique.

Le besoin de NAT sur une barrette est rare. En fait, les exemples dans ce document peuvent être les seules situations dans lesquelles cette configuration est nécessaire. Bien qu'il existe d'autres occasions où les utilisateurs emploient le routage de stratégie en même temps que la NAT, nous considérons qu'il ne s'agit pas de NAT sur une barrette car ces instances utilisent toujours plus d'une interface physique.

Exemple 1 Diagramme et configuration de réseau

Diagramme du réseau

Le diagramme de réseau ci-dessus est très commun dans une configuration de modem câble. Le système de terminaison par modem câble (CMTS) est un routeur et le modem câble (CM) est un périphérique qui agit comme un pont. Le problème auquel nous faisons face est que notre fournisseur de services Internet ne nous a pas donné assez d'adresses valides pour le nombre d'hôtes qui doivent atteindre Internet. Le fournisseur de services Internet nous a donné l'adresse 192.168.1.2, qui devait être utilisée pour un périphérique. Suite à une demande supplémentaire, nous en avons reçu trois autres - 192.168.2.1 à 192.168.2.3 - dans lesquelles NAT traduit les hôtes dans la portée 10.0.0.0/24.

Conditions requises

Nos exigences sont les suivantes :

- Tous les hôtes sur le réseau doivent pouvoir atteindre Internet.
- L'hôte 2 doit être accessible depuis Internet avec l'adresse IP 192.168.2.1.
- Puisque nous pouvons avoir plus d'hôtes que d'adresses légales, nous utilisons le sous-réseau 10.0.0.0/24 pour notre adressage interne.

Pour les besoins de ce document, nous montrons seulement la configuration du routeur NAT. Cependant, nous mentionnons quelques remarques importantes relatives à la configuration des hôtes.

Configuration du routeur NAT

Configuration du routeur NAT
interface Loopback0 ip address 10.0.1.1 255.255.255.252 ip nat outside *!--- Creates a virtual interface called Loopback 0 and assigns an !--- IP address of 10.0.1.1 to it. Defines interface Loopback 0 as !--- NAT outside. ! !* interface Ethernet0 ip address 192.168.1.2 255.255.255.0 secondary ip address 10.0.0.2 255.255.255.0 ip Nat inside *!--- Assigns a primary IP address of 10.0.0.2 and a secondary IP !--- address of 192.168.1.2 to Ethernet 0. Defines interface Ethernet 0 !--- as NAT inside. The 192.168.1.2 address will be used to communicate !--- through the CM to the CMTS and the Internet. The 10.0.0.2 address !--- will be used to communicate with the local hosts.* ip policy route-map Nat-loop *!--- Assigns route-map "Nat-loop" to Ethernet 0 for policy routing.* ! ip Nat pool external 192.168.2.2 192.168.2.3 prefix-length 29 ip Nat inside source list 10 pool external overload ip Nat inside source static 10.0.0.12 192.168.2.1 *!--- NAT is defined: packets that match* ***access-list 10*** *will be !--- translated to an address from the pool called "external". !--- A static NAT translation is defined for 10.0.0.12 to be !--- translated to 192.168.2.1 (this is for host 2 which needs !--- to be accessed from the Internet).* ip classless ! ! ip route 0.0.0.0 0.0.0.0 192.168.1.1 ip route 192.168.2.0 255.255.255.0 Ethernet0 *!--- Static default route set as 192.168.1.1, also a static !--- route for network 192.168.2.0/24 directly attached to !--- Ethernet 0* ! ! access-list 10 permit 10.0.0.0 0.0.0.255 *!---* ***Access-list 10*** *defined for use by NAT statement above.* access-list 102 permit ip any 192.168.2.0 0.0.0.255 access-list 102 permit ip 10.0.0.0 0.0.0.255 any *!---* ***Access-list 102*** *defined and used by route-map "Nat-loop" !--- which is used for policy routing.* ! Access-list 177 permit icmp any any *!---* ***Access-list 177*** *used for* ***debug***. ! route-map Nat-loop permit 10 match ip address 102 set ip next-hop 10.0.1.2 *!--- Creates route-map "Nat-loop" used for policy routing. !--- Route map states that any packets that match access-list 102 will !--- have the next hop set to 10.0.1.2 and be routed "out" the !--- loopback interface. All other packets will be routed normally. !--- We use 10.0.1.2 because this next-hop is seen as located !--- on the loopback interface which would result in policy routing to !--- loopback0. Alternatively, we could have used "set interface !--- loopback0" which would have done the same thing.* ! end NAT-router#

Remarque: tous les hôtes ont leur passerelle par défaut définie à 10.0.0.2, qui est le routeur NAT. Le fournisseur de services Internet et le CMTS doivent avoir une route à 192.168.2.0/29 qui pointe vers le routeur NAT pour que le trafic de retour fonctionne, car le trafic en provenance des hôtes internes apparaît comme provenant de ce sous-réseau. Dans cet exemple, le CMTS acheminerait

le trafic pour 192.168.2.0/29 à 192.168.1.2, qui est l'adresse IP secondaire configurée sur le routeur NAT.

Exemple 1 Sortie de commande show et debug

Cette section fournit des informations qui vous permettront de vérifier que votre configuration fonctionne correctement.

Afin d'illustrer que la configuration ci-dessus fonctionne, nous avons exécuté quelques **tests ping** pendant que la **sortie de débogage** sur le routeur NAT est contrôlé. Vous pouvez observer que les **commandes ping** réussissent et que la **sortie de débogage** montre exactement ce qui se produit.

Remarque: Avant d'utiliser les commandes **debug**, référez-vous à la section **Informations importantes sur les commandes Debug**.

Test Un

Pour notre premier test, nous exécutons une commande **ping** à partir d'un périphérique dans notre Internet défini comme laboratoire vers l'hôte 2. Souvenez-vous qu'une des conditions requises était que les périphériques Internet doivent pouvoir communiquer avec l'hôte 2 avec l'adresse IP 192.168.2.1. Voici la **sortie de débogage** telle qu'affichée sur le routeur NAT. Les commandes **debug** exécutées sur le routeur NAT étaient **debug ip packet 177 detail**, qui utilise l'**access-list 177** définie, **debug ip Nat** et **debug ip policy**, qui nous montre les paquets à routage géré par une stratégie.

Voici la sortie de la commande **show ip Nat translation** exécutée sur le routeur NAT :

```
NAT-router#show ip Nat translation Pro Inside global Inside local Outside local Outside global -
-- 192.168.2.1 10.0.0.12 --- --- NAT-router#
```

Depuis un périphérique sur Internet, dans ce cas un routeur, nous exécutons une commande **ping** sur 192.168.2.1 qui réussit, comme indiqué ici :

```
Internet-device#ping 192.168.2.1 Type escape sequence to abort. Sending 5, 100-byte ICMP Echos
to 192.168.2.1, timeout is 2 seconds: !!!!! Success rate is 100 percent (5/5), round-trip
min/avg/max = 92/92/92 ms Internet-device#
```

Pour observer ce qui se produit dans le routeur NAT, consultez ces **sorties de débogage** et commentaires :

```
IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1, len 100, policy match
    ICMP type=8, code=0
IP: route map Nat-loop, item 10, permit
IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1 (Loopback0), Len 100, policy routed
    ICMP type=8, code=0
!--- The above debug output shows the packet with source 177.10.1.3 destined !--- to
192.168.2.1. The packet matches the statements in the "Nat-loop" !--- policy route map and is
permitted and policy-routed. The Internet !--- Control Message Protocol (ICMP) type 8, code 0
indicates that this !--- packet is an ICMP echo request packet. IP: Ethernet0 to Loopback0
10.0.1.2 IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1 (Loopback0), g=10.0.1.2, Len 100, forward
ICMP type=8, code=0 !--- The packet now is routed to the new next hop address of 10.0.1.2 !---
as shown above. IP: NAT enab = 1 trans = 0 flags = 0 NAT: s=177.10.1.3, d=192.168.2.1->10.0.0.12
[52] IP: s=177.10.1.3 (Loopback0), d=10.0.0.12 (Ethernet0), g=10.0.0.12, Len 100, forward ICMP
type=8, code=0 IP: NAT enab = 1 trans = 0 flags = 0 !--- Now that the routing decision has been
made, NAT takes place. We can !--- see above that the address 192.168.2.1 is translated to
10.0.0.12 and !--- this packet is forwarded out Ethernet 0 to the local host. !--- Note: When a
packet is going from inside to outside, it is routed and !--- then translated (NAT). In the
```

```
opposite direction (outside to inside), !--- NAT takes place first. IP: s=10.0.0.12 (Ethernet0),
d=177.10.1.3, Len 100, policy match ICMP type=0, code=0 IP: route map Nat-loop, item 10, permit
IP: s=10.0.0.12 (Ethernet0), d=177.10.1.3 (Loopback0), Len 100, policy routed ICMP type=0,
code=0 IP: Ethernet0 to Loopback0 10.0.1.2 !--- Host 2 now sends an ICMP echo response, seen as
ICMP type 0, code 0. !--- This packet also matches the policy routing statements and is !---
permitted for policy routing. NAT: s=10.0.0.12->192.168.2.1, d=177.10.1.3 [52] IP: s=192.168.2.1
(Ethernet0), d=177.10.1.3 (Loopback0), g=10.0.1.2, Len 100, forward ICMP type=0, code=0 IP:
s=192.168.2.1 (Loopback0), d=177.10.1.3 (Ethernet0), g=192.168.1.1, Len 100, forward ICMP
type=0, code=0 IP: NAT enab = 1 trans = 0 flags = 0 !--- The above output shows the Host 2 IP
address is translated to !--- 192.168.2.1 and the packet that results packet is sent out
loopback 0, !--- because of the policy based routing, and finally forwarded !--- out Ethernet 0
to the Internet device. !--- The remainder of the debug output shown is a repeat of the previous
!--- for each of the additional four ICMP packet exchanges (by default, !--- five ICMP packets
are sent when pinging from Cisco routers). We have !--- omitted most of the output since it is
redundant. IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1, Len 100, policy match ICMP type=8,
code=0 IP: route map Nat-loop, item 10, permit IP: s=177.10.1.3 (Ethernet0), d=192.168.2.1
(Loopback0), Len 100, policy routed ICMP type=8, code=0 IP: Ethernet0 to Loopback0 10.0.1.2 IP:
s=177.10.1.3 (Ethernet0), d=192.168.2.1 (Loopback0), g=10.0.1.2, Len 100, forward ICMP type=8,
code=0 IP: NAT enab = 1 trans = 0 flags = 0 NAT: s=177.10.1.3, d=192.168.2.1->10.0.0.12 [53] IP:
s=177.10.1.3 (Loopback0), d=10.0.0.12 (Ethernet0), g=10.0.0.12, Len 100, forward ICMP type=8,
code=0 IP: NAT enab = 1 trans = 0 flags = 0 IP: s=10.0.0.12 (Ethernet0), d=177.10.1.3, Len 100,
policy match ICMP type=0, code=0 IP: route map Nat-loop, item 10, permit IP: s=10.0.0.12
(Ethernet0), d=177.10.1.3 (Loopback0), Len 100, policy routed ICMP type=0, code=0 IP: Ethernet0
to Loopback0 10.0.1.2 NAT: s=10.0.0.12->192.168.2.1, d=177.10.1.3 [53] IP: s=192.168.2.1
(Ethernet0), d=177.10.1.3 (Loopback0), g=10.0.1.2, Len 100, forward ICMP type=0, code=0 IP:
s=192.168.2.1 (Loopback0), d=177.10.1.3 (Ethernet0), g=192.168.1.1, Len 100, forward ICMP
type=0, code=0 IP: NAT enab = 1 trans = 0 flags = 0
```

Test Deux

Une autre de nos conditions requises est de permettre aux hôtes de communiquer avec Internet. Pour ce test, nous exécutons une commande **ping** sur le périphérique Internet à partir de l'hôte 1. Les commandes **show** et **debug** résultantes sont indiquées ci-dessous.

Initialement, la table de traduction NAT dans le routeur NAT est la suivante :

```
NAT-router#show ip Nat translation Pro Inside global Inside local Outside local Outside global -
-- 192.168.2.1 10.0.0.12 --- --- NAT-router#
```

Une fois que nous émettons la commande **ping** à partir de l'hôte 1, nous voyons :

```
Host-1#ping 177.10.1.3 Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to
177.10.1.3, timeout is 2 seconds: !!!!! Success rate is 100 percent (5/5), round-trip
min/avg/max = 92/92/96 ms Host-1#
```

Nous observons ci-dessus que la commande **ping** a réussi. La table NAT dans le routeur NAT ressemble maintenant à ceci :

```
NAT-router#show ip Nat translation Pro Inside global Inside local Outside local Outside global
icmp 192.168.2.2:434 10.0.0.11:434 177.10.1.3:434 177.10.1.3:434 icmp 192.168.2.2:435
10.0.0.11:435 177.10.1.3:435 177.10.1.3:435 icmp 192.168.2.2:436 10.0.0.11:436 177.10.1.3:436
177.10.1.3:436 icmp 192.168.2.2:437 10.0.0.11:437 177.10.1.3:437 177.10.1.3:437 icmp
192.168.2.2:438 10.0.0.11:438 177.10.1.3:438 177.10.1.3:438 --- 192.168.2.1 10.0.0.12 --- ---
NAT-router#
```

La table de traduction NAT ci-dessus contient maintenant des traductions supplémentaires qui sont le résultat de la configuration NAT dynamique (par opposition à la configuration NAT statique).

La sortie **de débogage** ci-dessous montre ce qui se produit sur le routeur NAT.

```
IP: NAT enab = 1 trans = 0 flags = 0
IP: s=10.0.0.11 (Ethernet0), d=177.10.1.3, Len 100, policy match
```

```
    ICMP type=8, code=0
IP: route map Nat-loop, item 10, permit
IP: s=10.0.0.11 (Ethernet0), d=177.10.1.3 (Loopback0), Len 100, policy routed
    ICMP type=8, code=0
IP: Ethernet0 to Loopback0 10.0.1.2
!--- The above output shows the ICMP echo request packet originated by !--- Host 1 which is
policy-routed out the loopback interface. NAT: s=10.0.0.11->192.168.2.2, d=177.10.1.3 [8] IP:
s=192.168.2.2 (Ethernet0), d=177.10.1.3 (Loopback0), g=10.0.1.2, Len 100, forward ICMP type=8,
code=0 IP: s=192.168.2.2 (Loopback0), d=177.10.1.3 (Ethernet0), g=192.168.1.1, Len 100, forward
ICMP type=8, code=0 IP: NAT enab = 1 trans = 0 flags = 0 !--- After the routing decision has
been made by the policy routing, !--- translation takes place, which translates the Host 1 IP
address of 10.0.0.11 !--- to an address from the "external" pool 192.168.2.2 as shown above. !--
- The packet is then forwarded out loopback 0 and finally out Ethernet 0 !--- to the Internet
device. IP: s=177.10.1.3 (Ethernet0), d=192.168.2.2, Len 100, policy match ICMP type=0, code=0
IP: route map Nat-loop, item 10, permit IP: s=177.10.1.3 (Ethernet0), d=192.168.2.2 (Loopback0),
Len 100, policy routed ICMP type=0, code=0 IP: Ethernet0 to Loopback0 10.0.1.2 IP: s=177.10.1.3
(Ethernet0), d=192.168.2.2 (Loopback0), g=10.0.1.2, Len 100, forward ICMP type=0, code=0 !---
The Internet device sends an ICMP echo response which matches our !--- policy, is policy-routed,
and forward out the Loopback 0 interface. IP: NAT enab = 1 trans = 0 flags = 0 NAT:
s=177.10.1.3, d=192.168.2.2->10.0.0.11 [8] IP: s=177.10.1.3 (Loopback0), d=10.0.0.11
(Ethernet0), g=10.0.0.11, Len 100, forward ICMP type=0, code=0 !--- The packet is looped back
into the loopback interface at which point !--- the destination portion of the address is
translated from 192.168.2.2 !--- to 10.0.0.11 and forwarded out the Ethernet 0 interface to the
local host. !--- The ICMP exchange is repeated for the rest of the ICMP packets, some of !---
which are shown below. IP: NAT enab = 1 trans = 0 flags = 0 IP: s=10.0.0.11 (Ethernet0),
d=177.10.1.3, Len 100, policy match ICMP type=8, code=0 IP: route map Nat-loop, item 10, permit
IP: s=10.0.0.11 (Ethernet0), d=177.10.1.3 (Loopback0), Len 100, policy routed ICMP type=8,
code=0 IP: Ethernet0 to Loopback0 10.0.1.2 NAT: s=10.0.0.11->192.168.2.2, d=177.10.1.3 [9] IP:
s=192.168.2.2 (Ethernet0), d=177.10.1.3 (Loopback0), g=10.0.1.2, Len 100, forward ICMP type=8,
code=0 IP: s=192.168.2.2 (Loopback0), d=177.10.1.3 (Ethernet0), g=192.168.1.1, Len 100, forward
ICMP type=8, code=0 IP: NAT enab = 1 trans = 0 flags = 0 IP: s=177.10.1.3 (Ethernet0),
d=192.168.2.2, Len 100, policy match ICMP type=0, code=0 IP: route map Nat-loop, item 10, permit
IP: s=177.10.1.3 (Ethernet0), d=192.168.2.2 (Loopback0), Len 100, policy routed ICMP type=0,
code=0 IP: Ethernet0 to Loopback0 10.0.1.2 IP: s=177.10.1.3 (Ethernet0), d=192.168.2.2
(Loopback0), g=10.0.1.2, Len 100, forward ICMP type=0, code=0 IP: NAT enab = 1 trans = 0 flags =
0 NAT: s=177.10.1.3, d=192.168.2.2->10.0.0.11 [9] IP: s=177.10.1.3 (Loopback0), d=10.0.0.11
(Ethernet0), g=10.0.0.11, Len 100, forward ICMP type=0, code=0
```

Exemple 2 Diagramme et configuration de réseau

Diagramme du réseau

Conditions requises

Nous voulons que certains périphériques derrière les deux sites (R1 et R3) communiquent. Les deux sites utilisent des adresses IP non-inscrites, par conséquent nous devons traduire les adresses quand ils communiquent l'un avec l'autre. Dans notre cas, l'hôte 10.10.10.1 est traduit en 200.200.200.1 et l'hôte 20.20.20.1 sera traduit en 100.100.100.1. Par conséquent, il faut que la traduction se produise dans les deux directions. Pour des raisons de comptabilité, le trafic entre ces deux sites doit passer par R2. Pour récapituler, nos exigences sont les suivantes :

- L'hôte 10.10.10.1, derrière R1, doit communiquer avec l'hôte 20.20.20.1 derrière R3 par le biais de leurs adresses globales.
- Le trafic entre ces hôtes doit être envoyé par l'intermédiaire de R2.
- Pour notre cas, nous avons besoin de traductions NAT statiques comme indiqué dans la configuration ci-dessous.

Configuration du routeur NAT

Configuration du routeur NAT
interface Loopback0 ip address 4.4.4.2 255.255.255.0 ip Nat inside *!--- Creates a virtual interface called "loopback 0" and assigns IP address !--- 4.4.4.2 to it. Also defines for it a NAT inside interface. !* Interface Ethernet0/0 ip address 1.1.1.2 255.255.255.0 no ip redirects ip Nat outside ip policy route-map Nat *!--- Assigns IP address 1.1.1.1/24 to e0/0. Disables redirects so that packets !--- which arrive from R1 destined toward R3 are not redirected to R3 and !--- visa-versa. Defines the interface as NAT outside interface. Assigns !--- route-map "Nat" used for policy-based routing. !* ip Nat inside source static 10.10.10.1 200.200.200.1 *!--- Creates a static translation so packets received on the inside interface !--- with a source address of 10.10.10.1 will have their source address !--- translated to 200.200.200.1. **Note**: This implies that the packets received !--- on the outside interface with a destination address of 200.200.200.1 !--- will have the destination translated to 10.10.10.1.* ip Nat outside source static 20.20.20.1 100.100.100.1 *!--- Creates a static translation so packets received on the outside interface !--- with a source address of 20.20.20.1 will have their source address !--- translated to 100.100.100.1. **Note**: This implies that packets received on !--- the inside interface with a destination address of 100.100.100.1 will !--- have the destination translated to 20.20.20.1.* ip route 10.10.10.0 255.255.255.0 1.1.1.1 ip route 20.20.20.0 255.255.255.0 1.1.1.3 ip route 100.100.100.0 255.255.255.0 1.1.1.3 ! access-list 101 permit ip host 10.10.10.1 host 100.100.100.1 route-map Nat permit 10 match ip address 101 set ip next-hop 4.4.4.2

Exemple 2 Sortie de commande show et debug

Remarque: certaines commandes show sont prises en charge par l'outil Interpréteur de sortie, qui vous permet d'afficher une analyse de la sortie de la commande show. Avant d'utiliser les commandes **debug**, référez-vous à la section **Informations importantes sur les commandes Debug**.

Test Un

Comme indiqué dans la configuration ci-dessus, nous avons deux traductions NAT statiques qui peuvent être observées sur R2 avec la commande show **ip Nat translation**.

Voici la sortie de la commande **show ip Nat translation** exécutée sur le routeur NAT :

```
NAT-router#show ip Nat translation Pro Inside global Inside local Outside local Outside global --- --- --- 100.100.100.1 20.20.20.1 --- 200.200.200.1 10.10.10.1 --- --- R2#
```

Pour ce test, nous avons exécuté une commande **ping** à partir d'un périphérique (10.10.10.1) derrière R1 destinée à l'adresse globale d'un périphérique (100.100.100.1) derrière R3. L'exécution de **debug ip Nat** et **debug ip packet** sur R2 a généré la sortie suivante :

```
IP: NAT enab = 1 trans = 0 flags = 0
IP: s=10.10.10.1 (Ethernet0/0), d=100.100.100.1, Len 100, policy match
    ICMP type=8, code=0
IP: route map Nat, item 10, permit
IP: s=10.10.10.1 (Ethernet0/0), d=100.100.100.1 (Loopback0), Len 100, policy
routed
    ICMP type=8, code=0
IP: Ethernet0/0 to Loopback0 4.4.4.2
!--- The above output shows the packet source from 10.10.10.1 destined !--- for 100.100.100.1
arrives on E0/0, which is defined as a NAT !--- outside interface. There is not any NAT that
needs to take place at !--- this point, however the router also has policy routing enabled for
!--- E0/0. The output shows that the packet matches the policy that is !--- defined in the
policy routing statements. IP: s=10.10.10.1 (Ethernet0/0), d=100.100.100.1 (Loopback0),
g=4.4.4.2, Len 100, forward ICMP type=8, code=0 IP: NAT enab = 1 trans = 0 flags = 0 !--- The
above now shows the packet is policy-routed out the loopback0 !--- interface. Remember the
loopback is defined as a NAT inside interface. NAT: s=10.10.10.1->200.200.200.1, d=100.100.100.1
[26] NAT: s=200.200.200.1, d=100.100.100.1->20.20.20.1 [26] !--- For the above output, the
packet is now arriving on the loopback0 !--- interface. Since this is a NAT inside interface, it
is important to !--- note that before the translation shown above takes place, the router !---
will look for a route in the routing table to the destination, which !--- before the translation
is still 100.100.100.1. Once this route look up !--- is complete, the router will continue with
translation, as shown above. !--- The route lookup is not shown in the debug output. IP:
s=200.200.200.1 (Loopback0), d=20.20.20.1 (Ethernet0/0), g=1.1.1.3, Len 100, forward ICMP
type=8, code=0 IP: NAT enab = 1 trans = 0 flags = 0 !--- The above output shows the resulting
translated packet that results is !--- forwarded out E0/0.
```

Voici la sortie du paquet de réponse provenant du périphérique derrière le routeur 3 destiné au périphérique derrière le routeur 1 :

```
NAT: s=20.20.20.1->100.100.100.1, d=200.200.200.1 [26]
NAT: s=100.100.100.1, d=200.200.200.1->10.10.10.1 [26]
 !--- The return packet arrives into the e0/0 interface which is a NAT !--- outside interface.
In this direction (outside to inside), translation !--- occurs before routing. The above output
shows the translation takes place. IP: s=100.100.100.1 (Ethernet0/0), d=10.10.10.1
(Ethernet0/0), Len 100, policy rejected -- normal forwarding ICMP type=0, code=0 IP:
s=100.100.100.1 (Ethernet0/0), d=10.10.10.1 (Ethernet0/0), g=1.1.1.1, Len 100, forward ICMP
type=0, code=0 !--- The E0/0 interface still has policy routing enabled, so the packet is !---
check against the policy, as shown above. The packet does not match the !--- policy and is
forwarded normally.
```

Résumé

ce document a expliqué comment la NAT et le routage basé sur la stratégie peuvent être utilisés pour créer un scénario « NAT sur barrette ». Il est important de se souvenir que cette configuration peut réduire les performances sur le routeur exécutant NAT car les paquets peuvent être commutés par processus par le routeur.

Informations connexes

- Page de support NAT
- Support technique - Cisco Systems