Traducción de Dirección de Red en un Solo Sentido

Contenido

Introducción

¿Qué significa la Traducción de dirección de red (NAT) en un solo sentido? El término "en un solo sentido" generalmente implica el uso de una sola interfaz física de un router para una tarea. Así como podemos usar subinterfaces de la misma interfaz física para realizar la conexión troncal del link entre switches (ISL), también podemos usar una sola interfaz física en un router para lograr la ejecución de NAT.

Nota: El router debe procesar el Switch cada paquete debido al Loopback Interface. Esto degrada el funcionamiento del router.

prerrequisitos

Requisitos

No hay requisitos específicos para este documento.

Componentes Utilizados

Esta característica le requiere utilizar una versión del Cisco IOS ® Software que soporte el NAT. Utilice el Cisco Feature Navigator II (clientes registrados solamente) para determinar que las versiones de IOS usted pueden utilizar con esta característica.

Convenciones

Para obtener más información sobre las convenciones del documento, consulte Convenciones de Consejos Técnicos de Cisco.

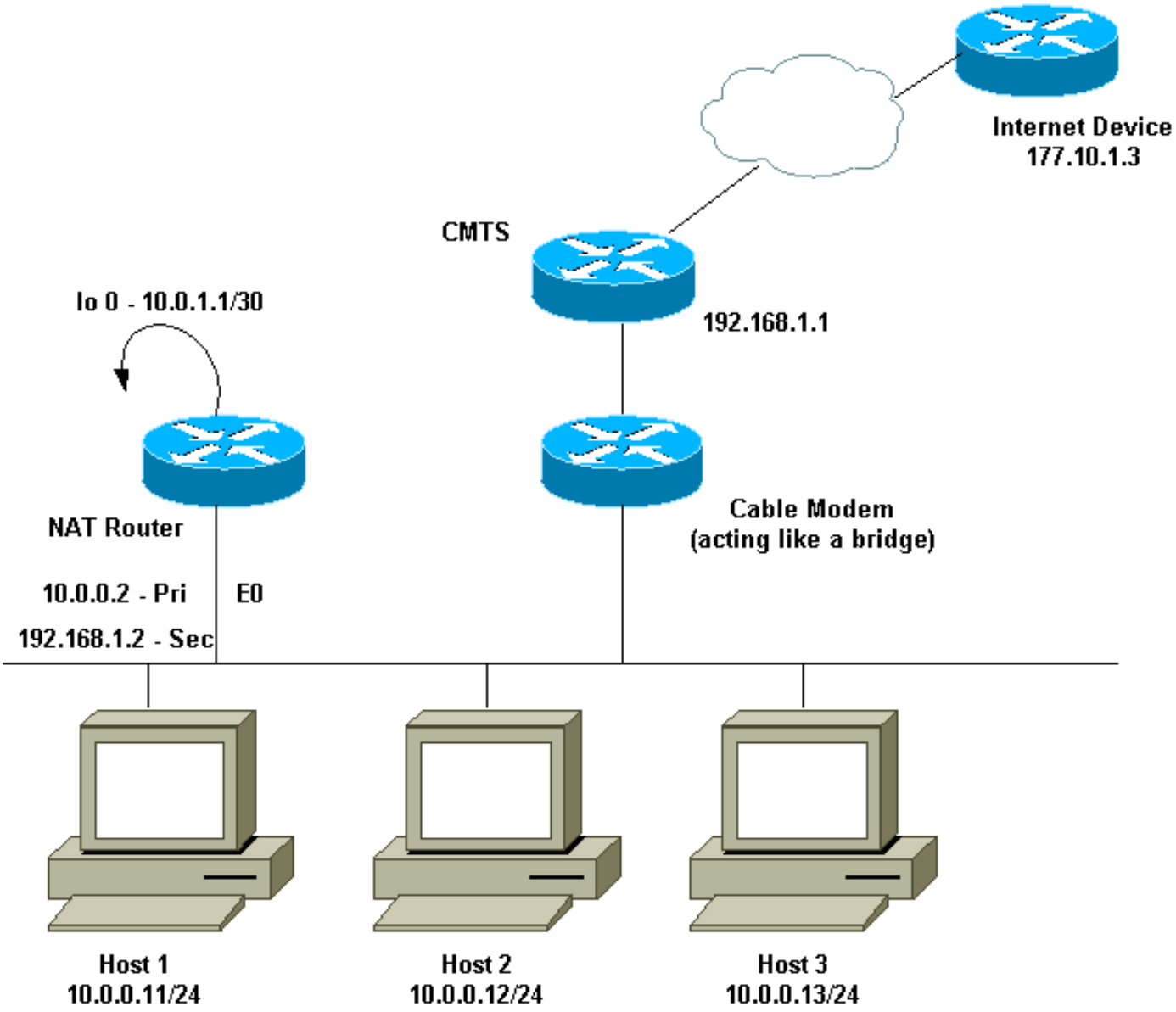
Antecedentes

Para que el NAT ocurra, un paquete se debe conmutar de una interfaz definida del “interior” NAT a una interfaz definida del “exterior” NAT o viceversa. Este requisito para NAT no ha cambiado, pero este documento demuestra cómo usted puede utilizar una interfaz virtual, si no conocido como un Loopback Interface, y Policy-Based Routing para hacer el trabajo NAT en un router con una sola interfaz física.

La necesidad del NAT en un solo sentido es rara. De hecho, los ejemplos en este documento pueden ser las únicas situaciones en las cuales esta configuración es necesaria. Aunque se presentan otras ocasiones donde los usuarios emplean el Policy Routing conjuntamente con el NAT, no consideramos esto ser NAT en un solo sentido porque estos casos todavía utilizan más de una interfaz física.

Ejemplo 1 Diagrama y configuración de la red

Diagrama de la red



El diagrama de red anterior es muy común en las configuraciones de cable módem. El sistema de terminación de cablemódem (CMTS) es un router y el cablemódem (CM) es un dispositivo que actúa como puente. El problema que hacemos frente es que nuestro Proveedor de servicios de Internet (ISP) no nos ha dado bastantes direcciones válidas para el número de host que necesitan alcanzar Internet. El ISP nos proporcionó la dirección 192.168.1.2, para ser utilizada con un dispositivo. Por el requerimiento adicional, recibimos tres más — 192.168.2.1 a 192.168.2.3 — en qué NAT traduce los host en el rango 10.0.0.0/24.

Requisitos

Nuestros requisitos son:

- Todos los host en la red deben poder alcanzar Internet.
- Se debe poder alcanzar el host 2 desde Internet con la dirección IP de 192.168.2.1.
- Porque podemos tener más host que las direcciones legales, utilizamos la subred 10.0.0.0/24 para nuestra dirección interna.

A los efectos de este documento, sólo mostramos la configuración del router NAT. Sin embargo, mencionamos algunas notas de configuración importantes en cuanto a los host.

Configuración del router NAT

Configuración del router 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#
```

Nota: Todos los hosts tienen sus gateway predeterminados definidos en 10.0.0.2, que es el router NAT. El ISP así como el CMTS deben tener una ruta a 192.168.2.0/29 que señale al router NAT para que el tráfico de retorno trabaje, porque el tráfico de los host interiores aparece como llegando de esta subred. En este ejemplo, el CMTS rutearía el tráfico para 192.168.2.0/29 a 192.168.1.2 que es la driección IP secundaria configurada en el router NAT.

Ejemplo 1: Resultados de los comandos show y debug

En esta sección encontrará información que puede utilizar para comprobar que su configuración funcione correctamente.

Para ilustrar que la configuración antedicha trabaja, hemos funcionado con algunas **pruebas de ping** mientras que monitorean a la **salida de los debugs** en el router NAT. Puede observar que los comandos ping son exitosos y que el resultado de debug muestra exactamente lo que está sucediendo.

Nota: Antes de que utilice los **comandos debug**, consulte Información Importante sobre los Comandos Debug.

Prueba uno

Para nuestra primera prueba, **hacemos ping de un** dispositivo en nuestro Internet laboratorio-definido para recibir 2. recordamos que uno de los requisitos era que los dispositivos en Internet deben poder comunicar con el host 2 con la dirección IP de 192.168.2.1. Lo que sigue es la **salida de los debugs** según lo visto en el router NAT. **Los comandos debug** que se ejecutaban en el router NAT eran el **detalle del paquete 177 del IP del debug** que utiliza la **lista de acceso** definida **177**, el **IP del debug nacional**, y la **directiva del IP del debug** que nos muestra los Policy-Routed Packet.

Ésta es la salida del **comando show ip Nat translation** ejecutado en el router 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#
```

De un dispositivo en Internet, en este caso un router, **hacemos ping** 192.168.2.1 que sea acertado como se muestra aquí:

```
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#
```

Para ver qué sucede en el router NAT, refiera a esta **salida de los debugs** y comentarios:

```
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
```

Prueba dos

Otro de nuestros requisitos es permitir que los host tengan la capacidad de comunicarse con Internet. Para esta prueba, **hacemos ping el** dispositivo de Internet del host 1. A continuación, se detallan los comandos show y debug obtenidos como resultado.

La tabla de traducción de NAT en el router NAT está inicialmente como sigue:

```
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#
```

Una vez que publicamos el **ping del** host 1, vemos:

```
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#
```

Arriba se observa que el ping fue satisfactorio. La tabla NAT en el router NAT ahora parece:

```
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 tabla de traducción NAT anterior ahora muestra las traducciones adicionales que son resultado de la configuración NAT dinámica (en contraposición con la configuración estática NAT).

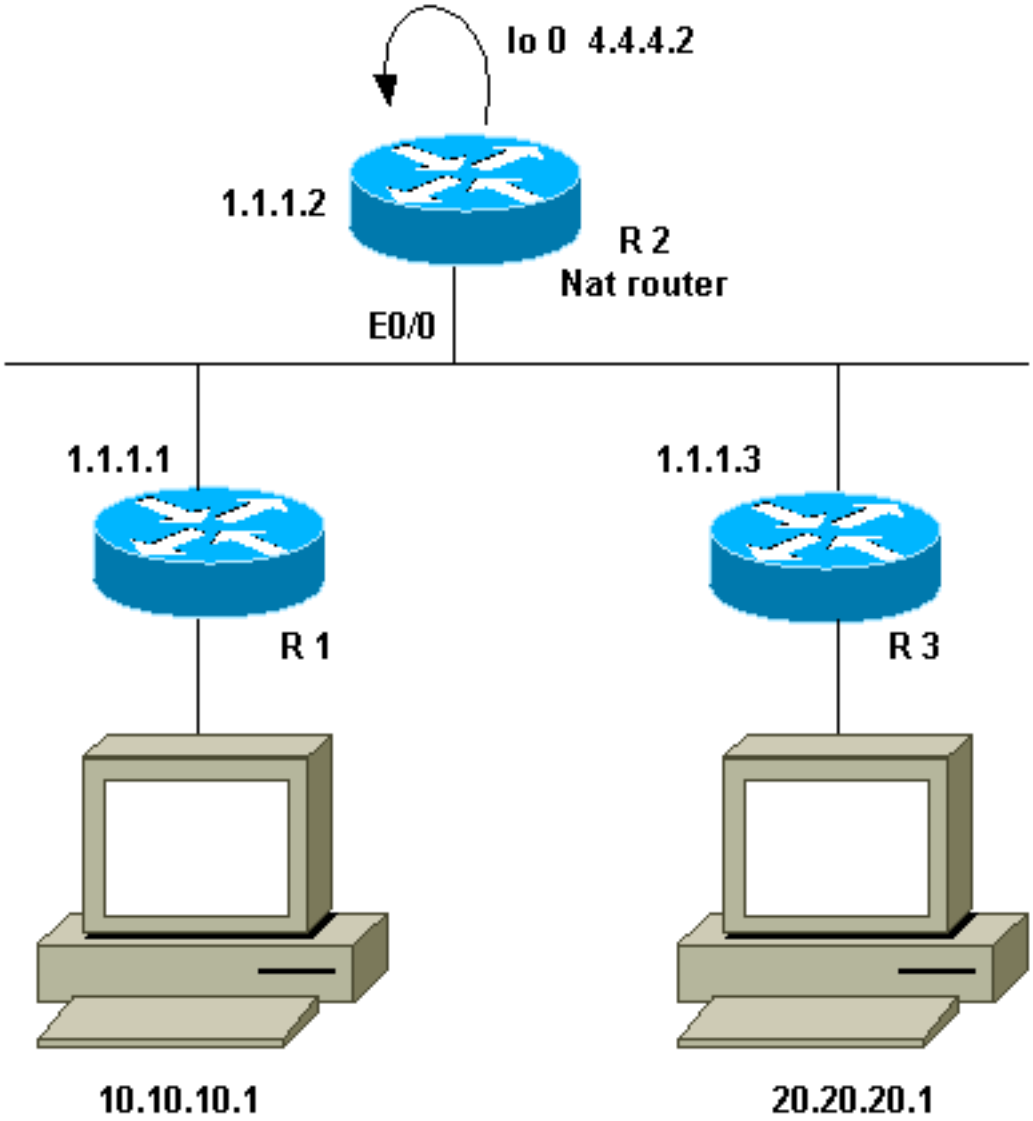
La salida de los debugs debajo de las demostraciones qué ocurre en el router NAT.

```
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
```

Ejemplo 2, configuración y diagrama de la red

Diagrama de la red



Requisitos

Queremos que ciertos dispositivos detrás de los dos sitios (R1 y R3) se comuniquen. Los dos sitios utilizan los IP Addresses no colocados, así que debemos traducir los direccionamientos cuando comunican con uno a. En nuestro caso, el host 10.10.10.1 se traduce a 200.200.200.1 y el host 20.20.20.1 será traducido a 100.100.100.1. Por lo tanto, necesitamos que la traducción se efectúe en ambas direcciones. Para los fines de la contabilidad, el tráfico entre estos dos sitios debe pasar a través de R2. Para resumir, nuestros requisitos son:

- Reciba 10.10.10.1, detrás del r1, las necesidades de comunicar con el host 20.20.20.1 detrás del R3 con el uso de sus direcciones globales.
- El tráfico entre estos hosts debe configurarse a través del R2.
- Para nuestro caso, necesitamos traducciones NAT estática como se muestra en la siguiente configuración.

Configuración del router NAT

Configuración del router 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
```

Ejemplo 2 de salida del comando show and debug

Nota: La herramienta del Output Interpreter soportan a los ciertos comandos show, que permite que usted vea una análisis de la salida del comando show. Antes de que utilice los **comandos debug**, consulte Información Importante sobre los Comandos Debug.

Prueba uno

Como se muestra en la configuración anterior, tenemos dos traducciones NAT estáticas que pueden verse en el R2 con el comando show ip Nat translation.

Ésta es la salida del **comando show ip Nat translation** ejecutado en el router 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#
```

Para esta prueba, nosotros originados un **ping de un** dispositivo (10.10.10.1) detrás del r1 destinada para la dirección global de un dispositivo (100.100.100.1) detrás del R3. **El IP** corriente del **debug nacional** y el **paquete del IP del debug** en el r2 dieron lugar a esta salida:

```
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.
```

Ésta es la salida como resultado del paquete de respuesta originado del dispositivo detrás del router3 destinado para el dispositivo detrás del router1:

```
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.
```

Resumen

Este documento ha demostrado cómo puede utilizarse NAT y el ruteo basado en políticas para crear un escenario "NAT en un solo sentido". Es importante tener presente que esta configuración puede reducir el funcionamiento en el router que ejecuta el NAT porque los paquetes pueden ser process-switched a través del router.

Información Relacionada

- Página de Soporte de NAT
- Soporte Técnico - Cisco Systems