

Tradução de endereço de rede em um stick

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Informações de Apoio](#)

[Exemplo 1 Diagrama e Configuração de Rede](#)

[Diagrama de Rede](#)

[Requisitos](#)

[Configuração de Roteador NAT](#)

[Exemplo 1 de saída do comando show and debug](#)

[Teste um](#)

[Teste dois](#)

[Exemplo 2 – Diagrama e Configuração de Rede](#)

[Diagrama de Rede](#)

[Requisitos](#)

[Configuração de Roteador NAT](#)

[Exemplo 2 da saída dos comandos show e debug](#)

[Teste um](#)

[Resumo](#)

[Informações Relacionadas](#)

[Introdução](#)

O que queremos dizer por Tradução de Endereço de Rede (NAT) em um cenário difícil? O termo “em um cenário difícil” implica geralmente o uso de uma única interface física de um roteador para uma tarefa. Assim como podemos usar subinterfaces da mesma interface física para realizar o truncamento do ISL (Enlace entre Switches), podemos usar uma única interface física em um roteador para realizar a NAT.

Nota: O roteador deve processar o interruptor cada pacote devido à interface de loopback. Isto degrada o desempenho do roteador.

[Pré-requisitos](#)

[Requisitos](#)

Não existem requisitos específicos para este documento.

[Componentes Utilizados](#)

Esta característica exige-o usar uma versão do Cisco IOS ® Software que apoie o NAT. Use o [Cisco Feature Navigator II \(clientes registrados somente\)](#) para determinar que Versões do IOS você pode usar com esta característica.

[Convenções](#)

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

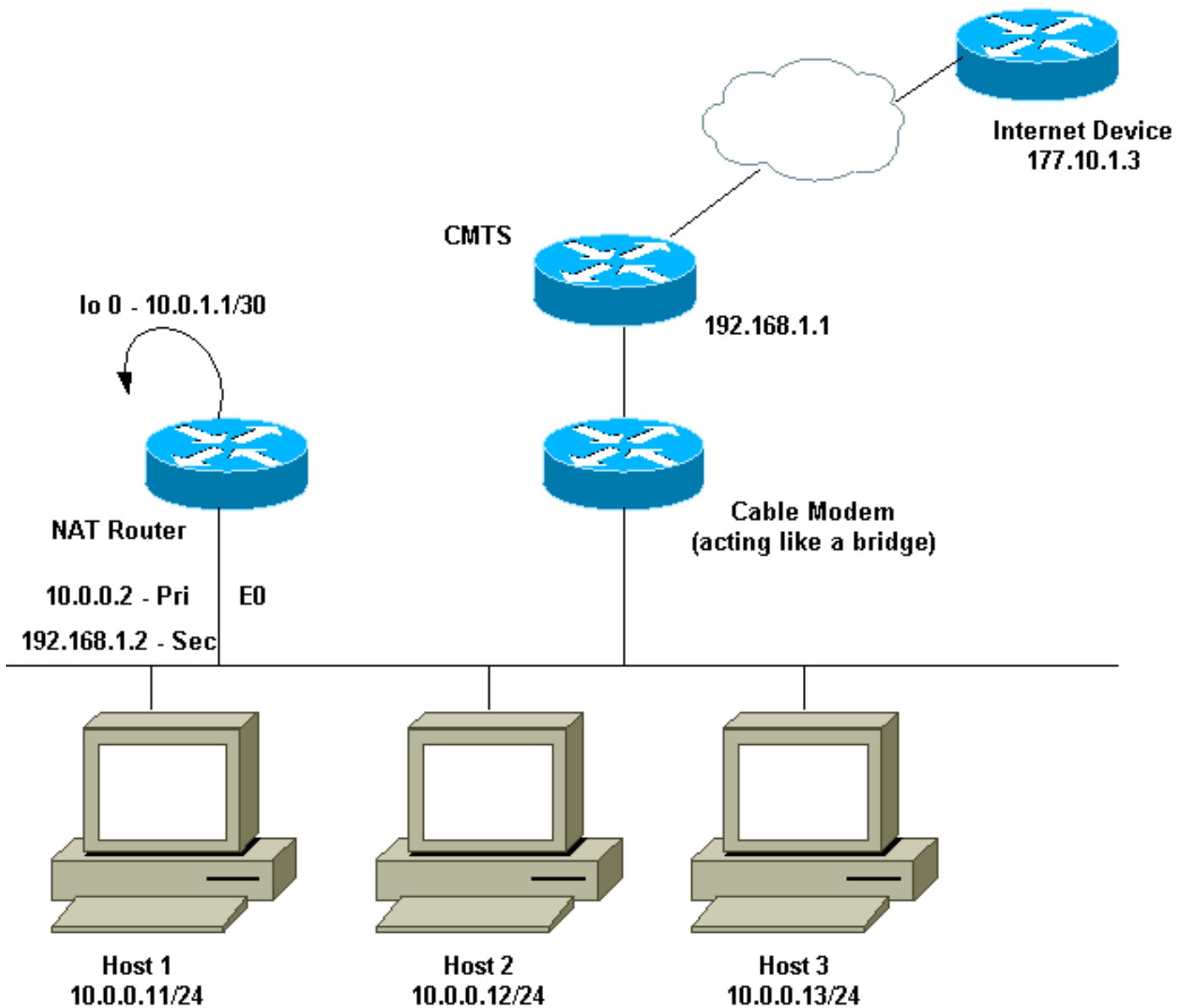
[Informações de Apoio](#)

Para que o NAT ocorra, um pacote deve ser comutado de uma interface definida do “interior” NAT a uma interface definida da “parte externa” NAT ou vice-versa. Este requisito de NAT não mudou, mas este documento demonstra como você pode usar uma interface virtual, se não sabido como uma interface de loopback, e o roteamento baseado em política para fazer o trabalho NAT em um roteador com uma única interface física.

A necessidade para o NAT em um cabo é rara. De fato, os exemplos neste documento podem ser as únicas situações em que esta configuração é precisada. Embora outras ocasiões elevaram onde os usuários empregam o roteamento de política conjuntamente com o NAT, nós não consideramos este ser NAT em um cabo porque estes exemplos ainda usam mais de uma interface física.

[Exemplo 1 Diagrama e Configuração de Rede](#)

[Diagrama de Rede](#)



O diagrama de rede acima é muito comum em uma configuração de modem a cabo. O sistema CMTS é um roteador, e o Modem a Cabo (CM) é um dispositivo que funciona como ponte. O problema que nós enfrentamos é que nosso provedor de serviço do Internet (ISP) não nos deu bastante endereços válidos para o número de anfitriões que precisam de alcançar o Internet. O ISP nos deu o endereço 192.168.1.2, que deveria ser usado para um dispositivo. Em cima de um pedido mais adicional, nós recebemos três mais — 192.168.2.1 a 192.168.2.3 — em que NAT traduz os anfitriões na escala 10.0.0.0/24.

Requisitos

Nossas exigências são:

- Todos os anfitriões na rede devem poder alcançar o Internet.
- O host 2 deve ser capaz de ser alcançado pela Internet com o endereço IP 192.168.2.1.
- Porque nós podemos ter mais anfitriões do que endereços legais, nós usamos a sub-rede 10.0.0.0/24 para nosso endereçamento interno.

Para os objetivos deste documento, mostramos somente a configuração do roteador NAT. Contudo, nós mencionamos algumas notas de configuração importantes no que diz respeito aos anfitriões.

Configuração de Roteador NAT

Configuração de Roteador 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: A configuração do gateway padrão de todos os hosts é 10.0.0.2, que é o roteador NAT. O ISP assim como o CMTS devem ter uma rota a 192.168.2.0/29 que aponte ao roteador NAT para que o tráfego de retorno trabalhe, porque o tráfego dos host internos aparece como chegando desta sub-rede. Neste exemplo, o CMTS distribuiria o tráfego para 192.168.2.0/29 a 192.168.1.2 que é o endereço IP secundário configurado no roteador NAT.

Exemplo 1 de saída do comando show and debug

Esta seção fornece informações que você pode usar para confirmar se sua configuração funciona adequadamente.

A fim ilustrar que a configuração acima trabalha, nós executamos alguns **testes de ping** quando o **resultado do debug no roteador NAT** for monitorado. Você pode ver se os comandos de ping são concluídos com êxito e se o resultado da depuração mostra exatamente o que está acontecendo.

Nota: Antes de utilizar **comandos debug**, consulte [Informações Importantes sobre Comandos Debug](#).

Teste um

Para nosso primeiro teste, nós **sibilamos de um** dispositivo em nosso Internet laboratório-definido para hospedar 2. recordamos que uma das exigências era que os dispositivos no Internet devem poder se comunicar com o host 2 com o endereço IP de Um ou Mais Servidores Cisco ICM NT de 192.168.2.1. O seguinte é o **resultado do debug** como visto no roteador NAT. **Os comandos debug** que estavam sendo executado no roteador NAT eram **debugam o detalhe do pacote 177 IP** que usa a **lista de acesso** definida **177**, **debugam o IP Nat**, e **debugam a política IP** que nos mostra os pacotes de informação roteado de política.

Esta é a saída do **comando show ip Nat translation** executado no roteador 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 um dispositivo no Internet, neste caso um roteador, nós **sibilamos 192.168.2.1** que seja bem sucedido como mostrado aqui:

```
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 o que acontece no roteador NAT, refira estes **resultado do debug** e comentários:

```
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

```

Teste dois

Outro de nossos requisitos é permitir aos hosts a capacidade de se comunicar com a Internet. Para este teste, nós **sibilamos** o dispositivo de Internet do host 1. Os comandos show e debug são exibidos abaixo.

Inicialmente a tabela de tradução NAT no roteador NAT é como segue:

```

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#

```

Uma vez que nós emitimos o **sibilo** do host 1, nós 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#

```

Vemos acima que o ping foi executado com sucesso. A tabela NAT no roteador NAT olha agora como:

```

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#

```

A tabela de traduções NAT acima agora mostra mais traduções que resultam da configuração de NAT dinâmica (em oposição à configuração de NAT estática).

O resultado do debug abaixo das mostras o que ocorre no roteador NAT.

```

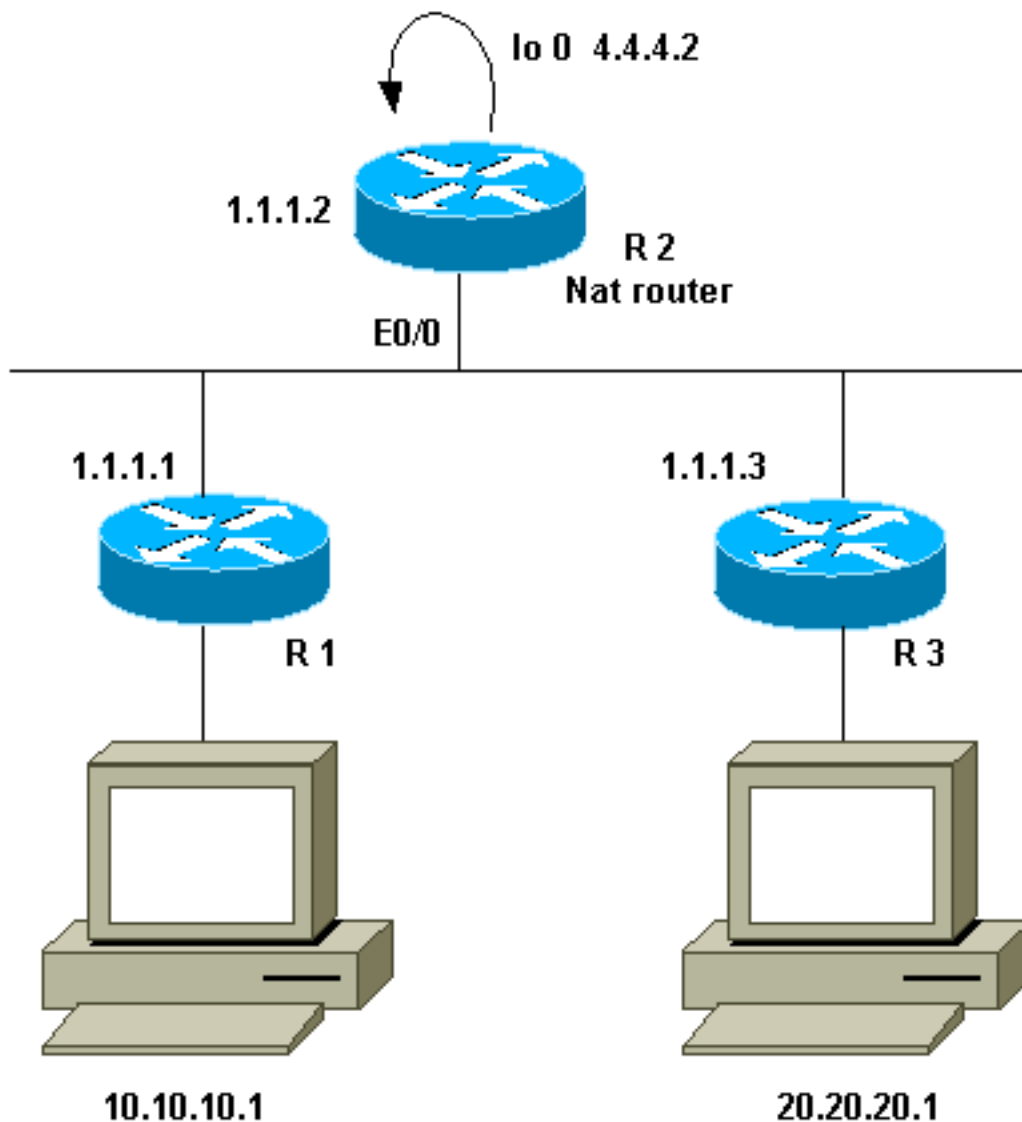
IP: NAT enab = 1 trans = 0 flags = 0
IP: s=10.0.0.11 (Ethernet0), d=177.10.1.3, Len 100, policy match
    ICMP type=8, code=0
IP: route map Nat-loop, item 10, permit
IP: s=10.0.0.11 (Ethernet0), d=177.10.1.3 (Loopback0), Len 100, policy routed
    ICMP type=8, code=0
IP: Ethernet0 to Loopback0 10.0.1.2
!--- 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

Exemplo 2 – Diagrama e Configuração de Rede

Diagrama de Rede



Requisitos

Queremos que alguns dispositivos atrás de dois locais (R1 e R3) se comuniquem. Os dois locais usam endereços IP de Um ou Mais Servidores Cisco ICM NT não-registrados, assim que nós devemos traduzir os endereços quando se comunicam um com o outro. Em nosso caso, o host 10.10.10.1 é traduzido a 200.200.200.1 e o host 20.20.20.1 será traduzido a 100.100.100.1. Portanto, precisamos que a conversão ocorra em ambas as direções. Para propósitos de contabilidade, o tráfego entre esses dois locais deve passar pelo R2. Para resumir, nossas exigências são:

- Hospede 10.10.10.1, atrás do r1, necessidades de comunicar-se com o host 20.20.20.1 atrás do R3 com o uso de seus endereços globais.
- O tráfego entre esses hosts deve ser enviado por meio de R2.
- Neste caso, a conversão de NAT estática é necessária, como mostrado na configuração a seguir.

Configuração de Roteador NAT

Configuração de Roteador 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

```

Exemplo 2 da saída dos comandos show e debug

Nota: Determinados comandos show são suportados pela Ferramenta Output Interpreter, que permite que você veja uma análise do resultado do comando show. Antes de utilizar **comandos debug**, consulte [Informações Importantes sobre Comandos Debug](#).

Teste um

Como mostra na configuração acima, temos duas conversões NAT que podemos ver em R2 com o comando show ip Nat translation.

Esta é a saída do **comando show ip Nat translation** executado no roteador 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 este teste, nós originado um **sibilo de um** dispositivo (10.10.10.1) atrás do r1 destinado para o endereço global de um dispositivo (100.100.100.1) atrás do R3. Ser executado **debuga o IP Nat e debuga o pacote IP no R2** conduziu a esta saída:

```

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.
```

Esta é a saída em consequência do pacote de resposta originado do dispositivo atrás do roteador3 destinado para o dispositivo atrás do roteador1:

```
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.
```

Resumo

Esse documento demonstrou como o uso do NAT e do roteamento baseado em política pode ser usado para criar um "NAT em um cenário difícil". É importante manter-se na mente que esta configuração pode reduzir o desempenho no roteador que executa o NAT porque os pacotes podem ser comutados por processo através do roteador.

Informações Relacionadas

- [Página de suporte de NAT](#)
- [Suporte Técnico - Cisco Systems](#)