Network Address Translation su Memory Stick

Sommario

Introduzione

Cosa si intende per NAT (Network Address Translation) su stick? Il termine "su bastone" in genere indica l'uso di una singola interfaccia fisica di un router per un'attività. Proprio come possiamo usare sottointerfacce della stessa interfaccia fisica per eseguire il trunking ISL (Inter-Switch Link), possiamo usare una singola interfaccia fisica su un router per realizzare il NAT.

Nota: il router deve elaborare ogni pacchetto dello switch a causa dell'interfaccia di loopback. Ciò riduce le prestazioni del router.

Prerequisiti

Requisiti

Nessun requisito specifico previsto per questo documento.

Componenti usati

Per questa funzionalità è necessario utilizzare una versione del software Cisco IOS® che supporti NAT. Per stabilire quali versioni IOS è possibile usare con questa funzione, usare Cisco Feature Navigator II (solo utenti registrati).

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni nei suggerimenti tecnici.

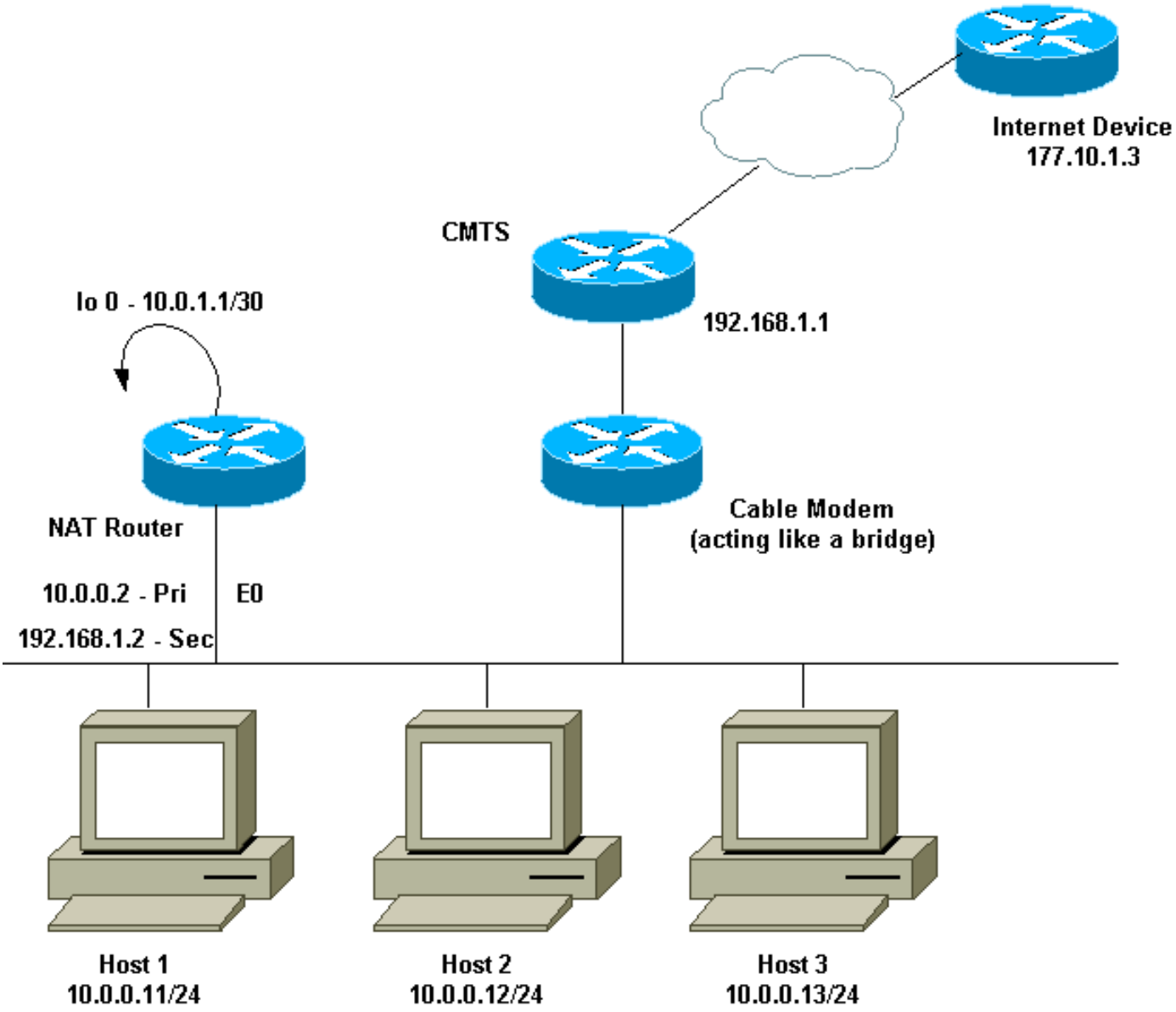
Premesse

Affinché il NAT abbia luogo, un pacchetto deve essere commutato da un'interfaccia definita "all'interno" NAT a un'interfaccia definita "all'esterno" NAT o viceversa. Questo requisito per NAT non è cambiato, ma questo documento dimostra come è possibile usare un'interfaccia virtuale, nota anche come interfaccia di loopback, e un routing basato su policy per far funzionare NAT su un router con una singola interfaccia fisica.

Il bisogno di NAT su un bastone è raro. In effetti, gli esempi riportati in questo documento potrebbero essere le uniche situazioni in cui è necessaria questa configurazione. Anche se si verificano altre occasioni in cui gli utenti utilizzano il routing di policy in combinazione con NAT, non consideriamo questo NAT su un stick perché queste istanze utilizzano ancora più di un'interfaccia fisica.

Esempio 1 - Esempio di rete e configurazione

Esempio di rete



Il diagramma di rete sopra riportato è molto comune nella configurazione di un modem via cavo. Il sistema di terminazione del modem via cavo (CMTS) è un router e il modem via cavo (CM) è un dispositivo che funziona come un bridge. Il problema che dobbiamo affrontare è che il nostro provider di servizi Internet (ISP) non ci ha fornito abbastanza indirizzi validi per il numero di host che devono raggiungere Internet. L'ISP ci ha dato l'indirizzo 192.168.1.2, che doveva essere usato per un dispositivo. Su ulteriore richiesta, abbiamo ricevuto altri tre—da 192.168.2.1 a 192.168.2.3—in cui NAT traduce gli host nell'intervallo 10.0.0.0/24.

Requisiti

I nostri requisiti sono:

- Tutti gli host della rete devono essere in grado di connettersi a Internet.
- L'host 2 deve essere raggiungibile da Internet con l'indirizzo IP 192.168.2.1.
- Poiché possiamo avere più host che indirizzi legali, utilizziamo la subnet 10.0.0.0/24 per i nostri indirizzi interni.

Ai fini di questo documento, viene mostrata solo la configurazione del router NAT. Tuttavia, vengono menzionate alcune importanti note di configurazione relative agli host.

Configurazione router NAT

Configurazione 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: su tutti gli host il gateway predefinito è impostato su 10.0.0.2, ossia sul router NAT. L'ISP e il CMTS devono avere un percorso verso 192.168.2.0/29 che punti al router NAT per consentire il traffico di ritorno per funzionare, in quanto il traffico proveniente dagli host interni sembra provenire da questa subnet. Nell'esempio, il CMTS instrada il traffico da 192.168.2.0/29 a 192.168.1.2, che è l'indirizzo IP secondario configurato sul router NAT.

Esempio 1: output del comando show e debug

Le informazioni contenute in questa sezione permettono di verificare che la configurazione funzioni correttamente.

Per dimostrare che la configurazione sopra descritta funziona, abbiamo eseguito alcuni test **ping** mentre viene monitorato l'output del **debug** sul router NAT. Come si può notare, i comandi **ping** sono stati eseguiti correttamente e l'output del comando **debug** mostra esattamente cosa sta succedendo.

Nota: prima di usare i comandi di **debug**, consultare le informazioni importanti sui comandi di debug.

Test 1

Per il primo test, è stato eseguito il **ping** da un dispositivo nell'Internet definito dal laboratorio all'host 2. Tenere presente che uno dei requisiti era che i dispositivi in Internet dovevano essere in grado di comunicare con l'host 2 con l'indirizzo IP 192.168.2.1. Di seguito viene riportato l'output del comando **debug** come mostrato sul router NAT. I comandi di debug in esecuzione sul router NAT sono **debug ip packet 177 detail** che utilizza il **file access-list 177** definito, **debug ip Nat** e **debug ip policy** che mostra i pacchetti con routing basato su criteri.

Questo è l'output del comando **show ip Nat translation** eseguito sul 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#
```

Da un dispositivo su Internet, in questo caso un router, è possibile eseguire il **ping** 192.168.2.1, che ha esito positivo, come mostrato di seguito:

```
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#
```

Per vedere cosa succede nel router NAT, fare riferimento a questo output del comando **debug** e ai seguenti commenti:

```
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 due

Un altro dei nostri requisiti è consentire agli host di comunicare con Internet. Per questo test, viene eseguito il **ping** del dispositivo Internet dall'host 1. I comandi **show** e **debug** risultanti sono riportati di seguito.

Inizialmente la tabella di conversione NAT nel router NAT è la seguente:

```
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#
```

Dopo aver inviato il comando **ping** dall'host 1, vengono visualizzati:

```
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#
```

Come si può vedere sopra, il **ping** è riuscito. La tabella NAT nel router NAT è ora simile alla seguente:

```
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#
```

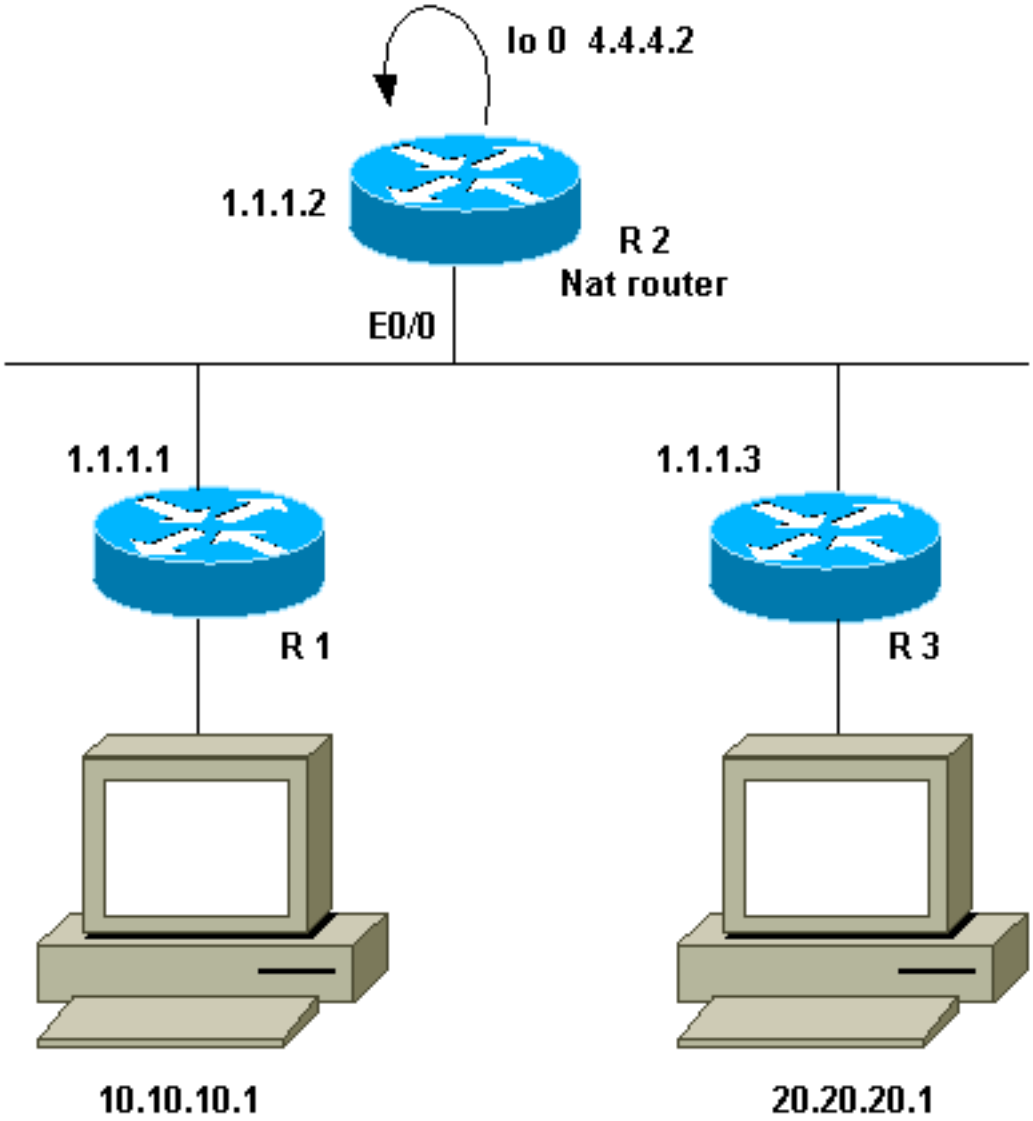
La tabella di conversione NAT sopra riportata mostra ulteriori conversioni risultanti dalla configurazione NAT dinamica (a differenza della configurazione NAT statica).

L'output del comando **debug** riportato di seguito mostra ciò che accade sul 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
```

Esempio 2 Esempio di diagramma di rete e configurazione

Esempio di rete



Requisiti

Si desidera che determinati dispositivi dietro i due siti (R1 e R3) comunichino. I due siti utilizzano indirizzi IP non registrati, pertanto è necessario tradurre gli indirizzi quando comunicano tra loro. Nel nostro caso, l'host 10.10.10.1 viene tradotto in 200.200.200.1 e l'host 20.20.20.1 verrà tradotto in 100.100.100.1. Pertanto, è necessario che la traduzione avvenga in entrambe le direzioni. A fini contabili, il traffico tra questi due siti deve passare attraverso R2. Per riassumere, i nostri requisiti sono:

- L'host 10.10.10.1, dietro R1, deve comunicare con l'host 20.20.20.1 dietro R3 con l'utilizzo dei relativi indirizzi globali.
- Il traffico tra questi host deve essere inviato tramite R2.
- Per il nostro caso, abbiamo bisogno di traduzioni NAT statiche come mostrato nella configurazione qui sotto.

Configurazione router NAT

Configurazione 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
```

Esempio 2: output del comando show e debug

Nota: alcuni comandi show sono supportati dallo strumento Output Interpreter, che consente di visualizzare un'analisi dell'output del comando show. Prima di usare i comandi di **debug**, consultare le informazioni importanti sui comandi di debug.

Test 1

Come mostrato nella configurazione sopra, abbiamo due traduzioni NAT statiche che possono essere viste su R2 con il comando **show ip Nat translation**.

Questo è l'output del comando **show ip Nat translation** eseguito sul 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#
```

Per questo test, è stato eseguito il **ping** da un dispositivo (10.10.10.1) dietro R1 destinato all'indirizzo globale di un dispositivo (100.100.100.1) dietro R3. L'esecuzione del comando **debug ip Nat** e del **pacchetto debug ip** su R2 ha prodotto questo output:

```
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.
```

Questo è l'output del pacchetto di risposta inviato dal dispositivo dietro il router 3 destinato al dispositivo dietro il router 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.
```

Riepilogo

Questo documento ha dimostrato come l'uso del NAT e il routing basato su policy possono essere usati per creare uno scenario "NAT on a stick". È importante tenere presente che questa configurazione può ridurre le prestazioni sul router che esegue NAT in quanto i pacchetti possono

essere commutati in base al processo attraverso il router.

Informazioni correlate

- Pagina di supporto NAT
- Supporto tecnico – Cisco Systems