Network Address Translation auf einem Stick

Inhalt

Einführung

Was bedeutet Network Address Translation (NAT) auf einem Stick? Der Begriff "auf einem Stick" impliziert in der Regel die Verwendung einer einzigen physischen Schnittstelle eines Routers für eine Aufgabe. So wie wir Subschnittstellen derselben physischen Schnittstelle für Inter-Switch Link (ISL)-Trunking verwenden können, können wir eine einzige physische Schnittstelle auf einem Router verwenden, um NAT zu erreichen.

Hinweis: Aufgrund der Loopback-Schnittstelle muss der Router jedes Paket verarbeiten. Dies beeinträchtigt die Leistung des Routers.

Voraussetzungen

Anforderungen

Für dieses Dokument bestehen keine speziellen Anforderungen.

Verwendete Komponenten

Für diese Funktion müssen Sie eine Version der Cisco IOS® Software verwenden, die NAT unterstützt. Bestimmen Sie mithilfe des Cisco Feature Navigator II (nur registrierte Kunden), welche IOS-Versionen Sie mit dieser Funktion verwenden können.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter Cisco Technical Tips Conventions.

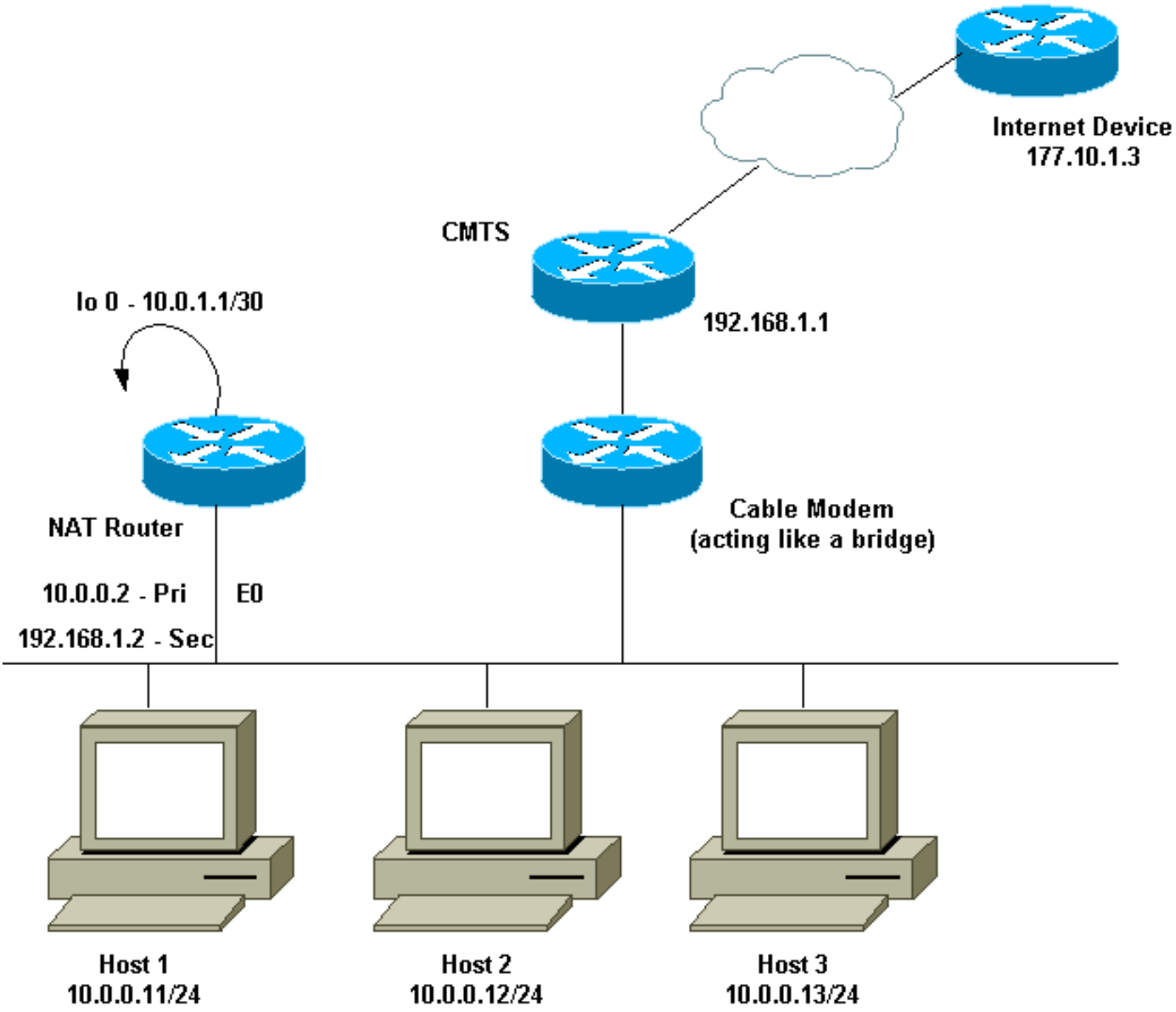
Hintergrundinformationen

Damit NAT durchgeführt werden kann, muss ein Paket von einer "internen", definierten NAT-Schnittstelle zu einer "externen" NAT-definierten Schnittstelle gewechselt werden oder umgekehrt. Diese NAT-Anforderung wurde nicht geändert. Dieses Dokument zeigt jedoch, wie Sie eine virtuelle Schnittstelle (auch Loopback-Schnittstelle genannt) und richtlinienbasiertes Routing verwenden können, um NAT auf einem Router mit einer einzigen physischen Schnittstelle zu verwenden.

NAT auf einem Stick ist selten erforderlich. Die Beispiele in diesem Dokument sind möglicherweise die einzigen Fälle, in denen diese Konfiguration erforderlich ist. Auch wenn andere Fälle auftreten, in denen Benutzer Richtlinienrouting in Verbindung mit NAT verwenden, wird dies nicht als NAT auf einem Stick betrachtet, da diese Instanzen immer noch mehr als eine physische Schnittstelle verwenden.

Beispiel 1 Netzwerkdiagramm und -konfiguration

Netzwerkdiagramm



Das obige Netzwerkdiagramm wird häufig bei der Einrichtung eines Kabelmodems verwendet. Das Cable Modem Termination System (CMTS) ist ein Router, und das Cable Modem (CM) ist ein Gerät, das wie eine Bridge funktioniert. Das Problem besteht darin, dass unser Internet Service Provider (ISP) uns nicht genügend gültige Adressen für die Anzahl der Hosts zur Verfügung gestellt hat, die ins Internet gelangen müssen. Der ISP gab uns die Adresse 192.168.1.2, die für ein Gerät verwendet werden sollte. Auf weitere Anfrage erhielten wir drei weitere 192.168.2.1 bis 192.168.2.3, in die NAT die Hosts im Bereich 10.0.0.0/24 übersetzt.

Anforderungen

Unsere Anforderungen sind:

- Alle Hosts im Netzwerk müssen in der Lage sein, das Internet zu erreichen.
- Host 2 muss über das Internet mit der IP-Adresse 192.168.2.1 erreichbar sein.
- Da wir mehr Hosts als legale Adressen haben können, verwenden wir das Subnetz 10.0.0.0/24 für unsere interne Adressierung.

Für die Zwecke dieses Dokuments wird nur die Konfiguration des NAT-Routers angezeigt. Wir erwähnen jedoch einige wichtige Konfigurationshinweise für die Hosts.

NAT-Router-Konfiguration

NAT-Router-Konfiguration

```
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#
```

Hinweis: Auf allen Hosts ist das Standard-Gateway auf 10.0.0.2 festgelegt, d. h. den NAT-Router. Sowohl der ISP als auch das CMTS müssen über eine Route zu 192.168.2.0/29 verfügen, die auf den NAT-Router zeigt, damit der Rückverkehr funktioniert, da der Datenverkehr von den internen Hosts scheinbar von diesem Subnetz ankommt. In diesem Beispiel leitet das CMTS den Datenverkehr für 192.168.2.0/29 an 192.168.1.2 weiter. Dies ist die sekundäre IP-Adresse, die auf dem NAT-Router konfiguriert wurde.

Beispiel 1: Ausgabe des Befehls show and debug

Dieser Abschnitt enthält Informationen zur Bestätigung, dass Ihre Konfiguration ordnungsgemäß funktioniert.

Um zu verdeutlichen, dass die obige Konfiguration funktioniert, haben wir einige **Ping-**Tests ausgeführt, während die **Debug-**Ausgabe auf dem NAT-Router überwacht wird. Sie können sehen, dass die **Ping-**Befehle erfolgreich sind und die **Debug-**Ausgabe genau anzeigt, was geschieht.

Hinweis: Bevor Sie **Debugbefehle** verwenden, lesen Sie Wichtige Informationen über Debug-Befehle.

Test 1

Für den ersten Test **pingen** wir von einem Gerät im Übungs-Internet zu Host 2. Denken Sie daran, dass eine der Anforderungen darin bestand, dass Geräte im Internet mit Host 2 mit der IP-Adresse 192.168.2.1 kommunizieren können müssen. Im Folgenden sehen Sie die **Debug-**Ausgabe, die auf dem NAT-Router angezeigt wird. Die **Debug**-Befehle, die auf dem NAT-Router ausgeführt wurden, **debug ip packet 177 detail** die die definierte **Zugriffsliste 177**, **debug ip Nat** und **Debug ip ip-Richtlinie** verwendet**, die die richtliniengesteuerten Pakete anzeigt.**

Dies ist die Ausgabe des Befehls **show ip Nat translation**, der auf dem NAT-Router ausgeführt wird:

```
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#
```

Von einem Gerät im Internet, in diesem Fall von einem Router, **pingen** wir 192.168.2.1, was erfolgreich ist, wie hier gezeigt:

```
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#
```

Weitere Informationen zu den Vorgängen im NAT-Router finden Sie in der **Debug-**Ausgabe und in

den folgenden Kommentaren:

```
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 2

Eine weitere unserer Anforderungen besteht darin, den Hosts die Kommunikation mit dem Internet zu ermöglichen. Für diesen Test **pingen** wir das Internetgerät von Host 1 aus. Die resultierenden Befehle **show** und **debug** sind unten aufgeführt.

Zunächst ist die NAT-Übersetzungstabelle im NAT-Router wie folgt:

```
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#
```

Sobald der **Ping** von Host 1 ausgegeben wurde, wird Folgendes angezeigt:

```
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#
```

Wir sehen oben, dass der **Ping** erfolgreich war. Die NAT-Tabelle im NAT-Router sieht nun wie folgt aus:

```
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#
```

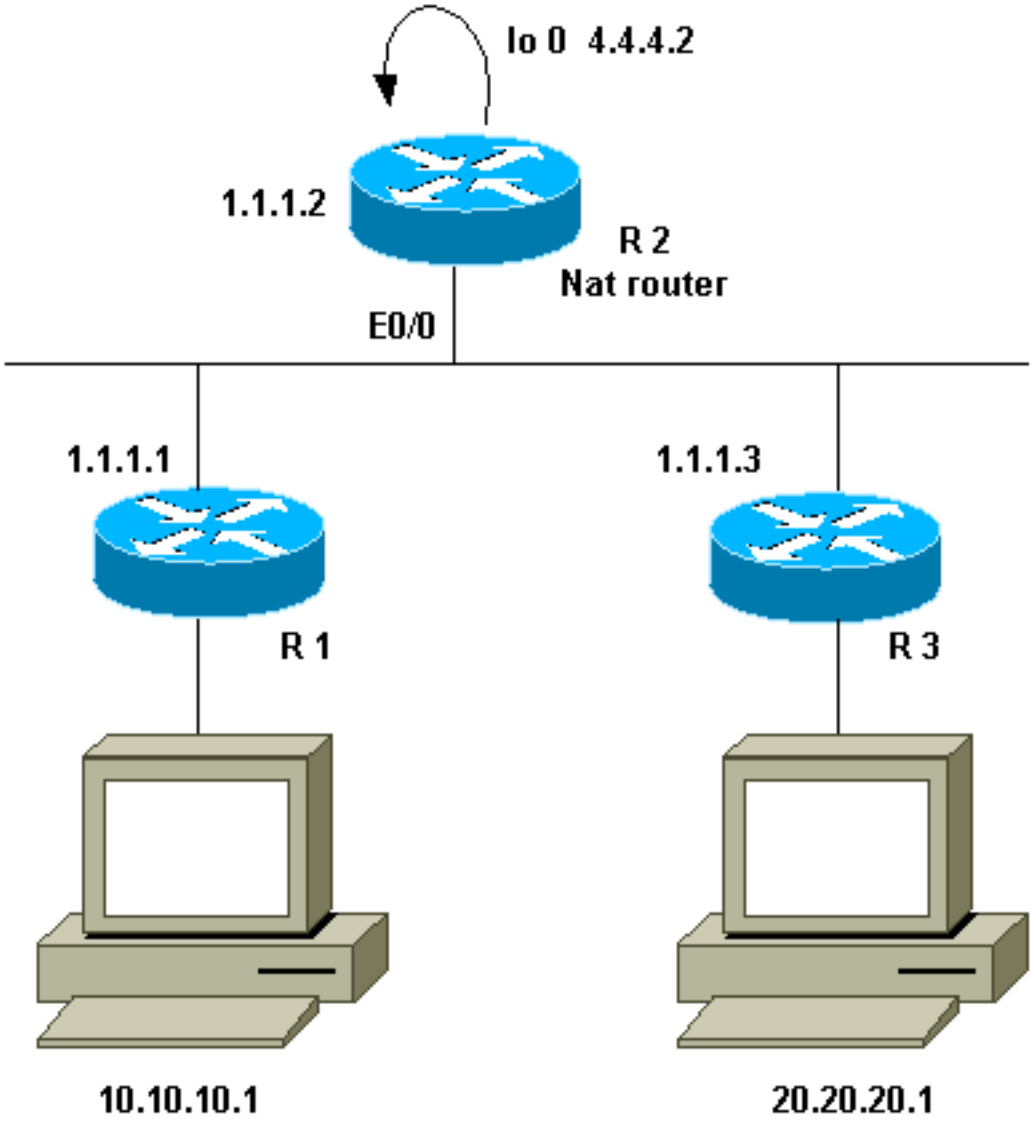
Die obige NAT-Übersetzungstabelle zeigt nun weitere Übersetzungen, die das Ergebnis der dynamischen NAT-Konfiguration sind (im Gegensatz zur statischen NAT-Konfiguration).

Die folgende **Debugausgabe** zeigt, was auf dem NAT-Router geschieht.

```
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
```

Beispiel 2 Netzwerkdiagramm und -konfiguration

Netzwerkdiagramm



Anforderungen

Wir möchten, dass bestimmte Geräte hinter den beiden Standorten (R1 und R3) kommunizieren. Die beiden Standorte verwenden nicht registrierte IP-Adressen, daher müssen wir die Adressen übersetzen, wenn sie miteinander kommunizieren. In unserem Fall wird Host 10.10.10.1 in 200.200.200.1 übersetzt und Host 20.20.20.1 in 100.100.1 übersetzt. Daher müssen wir Übersetzungen in beide Richtungen durchführen. Zu Buchhaltungszwecken muss der Datenverkehr zwischen diesen beiden Standorten R2 passieren. Zusammenfassend lassen sich folgende Anforderungen feststellen:

- Host 10.10.10.1, hinter R1, muss mit Host 20.20.20.1 hinter R3 unter Verwendung ihrer globalen Adressen kommunizieren.
- Der Datenverkehr zwischen diesen Hosts muss über R2 gesendet werden.
- Für unseren Fall benötigen wir statische NAT-Übersetzungen, wie in der unten stehenden Konfiguration gezeigt.

NAT-Router-Konfiguration

NAT-Router-Konfiguration

```
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
```

Beispiel 2 Ausgabe des Befehls show and debug

Hinweis: Bestimmte Anzeigebefehle werden vom Tool Output Interpreter unterstützt, mit dem Sie eine Analyse der Ausgabe des Befehls show anzeigen können. Bevor Sie **Debug-**Befehle verwenden, lesen Sie die Informationen unter Wichtige Informationen über Debug-Befehle.

Test 1

Wie in der obigen Konfiguration gezeigt, gibt es zwei statische NAT-Übersetzungen, die auf R2 mit dem Befehl **show ip Nat translation** angezeigt werden.

Dies ist die Ausgabe des Befehls **show ip Nat translation**, der auf dem NAT-Router ausgeführt wird:

```
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#
```

Für diesen Test haben wir einen **Ping** von einem Gerät (10.10.10.1) hinter R1 bezogen, das für die globale Adresse eines Geräts (100.100.100.1) hinter R3 bestimmt ist. Die Ausführung von **debug ip Nat** und **debug ip paket** auf R2 führte zu dieser Ausgabe:

```
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.
```

Dies ist die Ausgabe als Ergebnis des Antwortpakets, das vom Gerät hinter Router 3 stammt und für das Gerät hinter Router 1 bestimmt ist:

```
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.
```

Zusammenfassung

Dieses Dokument hat gezeigt, wie mithilfe von NAT und richtlinienbasiertem Routing ein "NAT-on-a-Stick"-Szenario erstellt werden kann. Es ist wichtig zu beachten, dass diese Konfiguration die

Leistung auf dem Router, auf dem NAT ausgeführt wird, verringern kann, da die Pakete möglicherweise über den Router prozessgeschaltet werden.

Zugehörige Informationen

- NAT-Support-Seite
- Technischer Support - Cisco Systems