Guide Pratique du NAT sous Linux 2.4

Rusty Russell
Titre original : ‘*Linux 2.4 NAT HOWTO*’
Traduction initiale : Emmanuel Roger
Dernière adaptation : Guillaume Audirac*guillaume POINT audirac CHEZ netpratique POINT fr*
Relecture : Thomas Nemeth*tnemeth CHEZ free POINT fr*
v1.18.fr.1.0, le 20 Mai 2004, traduction/adaptation

Ce document décrit comment réaliser du camouflage d'adresse IP, un serveur mandataire transparent, de la redirection de ports ou d'autres formes de Traduction d'Adresse Réseau (‘Network Address Translation’ ou NAT) avec le noyau Linux 2.4.

Table des matières

1 Introduction

Bienvenue, ami lecteur.

Vous êtes sur le point de plonger dans le monde fascinant (et parfois redoutable) de la **Traduction d'Adresse Réseau** ou NAT ('Network Address Translation'), parfois appelée Transformation d'Adresse Réseau. Ce Guide Pratique vous accompagnera aussi précisément que possible dans les noyaux de Linux 2.4 et au-delà.

Sous Linux 2.4, une infrastructure appelée 'Netfilter' a été introduite pour marquer les paquets. Une couche au-dessus de celle-ci fournit le NAT, complètement réimplémenté depuis les noyaux précédents.

2 Où se Trouvent les Sites Web Officiels et la Liste de Diffusion ?

Voici les trois sites officiels :
- Merci à *Filewatcher* `http://netfilter.filewatcher.org/` .
- Merci à *l'équipe de samba et SGI* `http://netfilter.samba.org/ter` .
- Merci à *Harald Welte* `http://netfilter.gnumonks.org/` .

Vous pouvez tous les atteindre en utilisant le DNS de type Round-Robin via

`http://www.netfilter.org/` et `http://www.iptables.org/`

Pour la liste de diffusion officielle de Netfilter :

Liste de Netfilter `http://www.netfilter.org/contact.html#list` .

2.1 Qu'est-ce Que la Traduction d'Adresse Réseau ?

Normalement, les paquets sur un réseau voyagent de leur source (par exemple, votre ordinateur de bureau) à leur destination (par exemple, www.gnumonks.org) en traversant différents liens : dans mon cas, environ 19 d'où je suis en Australie. Aucun de ces liens ne modifie vraiment votre paquet : ils le renvoient juste plus loin.

Si l'un de ces liens effectuait du NAT, alors il modifierait l'adresse source ou destination du paquet au moment où il passe. Comme on peut l'imaginer, le système n'a pas été prévu pour fonctionner comme ça, et le NAT est toujours quelque-chose de bancal. Généralement, le lien effectuant du NAT mémorise comment il a modifié un paquet, et quand une réponse arrive dans l'autre sens, il effectue la modification inverse sur ce paquet de réponse, pour que tout fonctionne bien.

2.2 Pourquoi Voudrais-Je Utiliser du NAT ?

Dans un monde parfait, vous n'en auriez pas besoin. Néanmoins, voici les raisons principales :

Les Connexions par Modem à Internet

La plupart des FAI (Fournisseurs d'Accès à Internet) vous donnent une seule adresse IP quand vous vous connectez chez eux. De ce fait, vous pouvez envoyer des paquets avec l'adresse source que vous voulez, mais seules vous seront envoyées les réponses aux paquets avec l'adresse IP source qui vous a été attribuée. Si vous voulez utiliser plusieurs machines différentes (comme dans un réseau personnel) pour vous connecter à Internet à travers ce lien unique, vous avez besoin du NAT.

C'est de loin l'utilisation la plus fréquente du NAT de nos jours, généralement connue sous le nom de camouflage d'adresse IP ('masquerading') dans le monde Linux. J'appelle ça du SNAT, parce que vous changez l'adresse **source** du premier paquet.

Serveurs Multiples

Parfois, vous voulez changer la direction des paquets arrivant dans votre réseau. C'est souvent parce que vous n'avez qu'une seule adresse IP (comme ci-dessus), mais vous voulez quand même qu'on puisse accéder aux machines qui se trouvent derrière celle avec l'adresse IP 'réelle'. Vous pouvez le faire si vous remaniez la destination des paquets entrants. Ce genre de NAT est appelé 'redirection de ports' ('port-forwarding') dans les précédentes versions de Linux.

Une variante courante de ceci est le partage de charge ('load-sharing'), où les paquets sont répartis sur un parc de machines. Si vous faites ceci sur une large échelle, vous pouvez jeter un oeil à

Serveur Linux Virtuel `http://linuxvirtualserver.org/` .

Mandataire Transparent

Parfois, vous voulez faire croire que chaque paquet traversant votre machine sous Linux est destiné à un programme sur cette machine même. Ceci est utilisé pour réaliser des mandataires transparents : un mandataire (ou 'proxy') est un programme qui se situe entre votre réseau et le monde extérieur, établissant la communication entre les deux. La transparence traduit le fait que votre réseau ne sait pas qu'il communique avec un mandataire, à moins évidemment qu'il ne fonctionne pas.

Squid peut être configuré pour fonctionner de cette manière, et c'est ce qu'on appelle une redirection ou un mandataire transparent dans les versions précédentes de Linux.

3 Les Deux Types de NAT

Je sépare le NAT en deux types différents : le **NAT de Source** (SNAT) et le **NAT de Destination** (DNAT).

Le NAT de source, c'est lorsqu'on modifie l'adresse source du premier paquet : c'est-à-dire que vous changez le lieu dont est issue la connexion. Le NAT de source est toujours réalisé après le routage, juste avant que le paquet ne soit envoyé sur la ligne. Le camouflage est une forme spécialisée de SNAT.

Le NAT de destination, c'est lorsqu'on modifie l'adresse de destination du premier paquet : c'est-à-dire que vous changez le lieu où la connexion va aboutir. Le NAT de destination est toujours effectué avant le routage, aussitôt que le paquet arrive sur la ligne. La redirection de port, le partage de charge et le mandataire transparent sont tous des formes de DNAT.

4 Traduction Rapide À Partir des Noyaux 2.0 et 2.2

Désolé pour tous ceux d'entre-vous qui restent choqués par la transition du 2.0 (Ipfwadm) au 2.2 (Ipchains). Il y a de bonnes et de mauvaises nouvelles.

Tout d'abord, vous pouvez toujours utiliser Ipchains et Ipfwadm comme avant. Pour cela, vous aurez besoin d'insérer (avec insmod) les modules 'ipchains.o' ou 'ipfwadm.o' trouvés dans la dernière distribution de Netfilter. Ils sont mutuellement exclusifs (vous êtes prévenus) et ne doivent être associés avec aucun autre module de Netfilter.

Une fois un de ces modules installé, vous pouvez utiliser Ipchains ou Ipfwadm comme avant, avec les différences suivantes :

– Configurer les durées de validité du camouflage avec ipchains -M -S ou ipfwadm -M -s, est sans effet. Puisque les durées de validité sont plus longues pour la nouvelle infrastructure de NAT, ça n'a plus d'importance.
– Pour le camouflage, les champs init_seq, delta et previous_delta du listage détaillé sont toujours à zéro.
– Initialiser les compteurs et les lister en même temps avec '-Z -L' ne fonctionne plus : les compteurs ne seront pas remis à zéro.
– La rétrocompatibilité fonctionne mal pour un grand nombre de connexions : ne l'utilisez pas sur une passerelle professionnelle !

Les bidouilleurs remarqueront aussi :

– Vous pouvez à présent utiliser les ports 61000-65095 même si vous effectuez du camouflage. Le code de camouflage considérait que tout ce qui se trouvait dans cet intervalle était déloyal, donc les programmes ne pouvaient l'utiliser.
– Le bidouillage 'getsockname' (non documenté) que les programmes des mandataires transparents pouvaient utiliser pour découvrir la destination réelle des connexions ne fonctionne plus.
– Le bidouillage 'bind-to-foreign-address' (non documenté) n'est également plus implémenté ; il servait à réaliser l'illusion d'un mandataire transparent.

4.1 Au secours ! Je Veux Juste du Camouflage !

C'est ce que veulent la plupart des gens. Si vous avez une connexion PPP allouée dynamiquement (si vous ne savez pas, c'est le cas), vous voulez simplement dire à votre machine que tous les paquets qui viennent de votre réseau interne doivent avoir l'air de venir de la machine qui initie la connexion PPP.

```
# Charger le module du NAT (il charge tous les autres).
modprobe iptable_nat

# Dans la table du NAT (-t nat), ajouter une règle (-A) après le routage
# (POSTROUTING) pour tous les paquets qui sortent par ppp0 (-o ppp0) qui stipule
# de camoufler la connexion (-j MASQUERADE).
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE

# Activer la redirection d'IP
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Notez que vous n'allez effectuer aucun filtrage de paquets ici : pour cela, lisez le chapitre du Guide Pratique du Filtrage de Paquets : 'Mélanger le NAT et le Filtrage de Paquets'.

4.2 Et Au Sujet d'ipmasqadm ?

C'est une catégorie d'utilisateurs beaucoup plus restreinte, donc je ne m'attache pas à la compatibilité à ce point là. Vous pouvez simplement utiliser 'iptables -t nat' pour effectuer de la redirection de port. Donc par exemple, sous Linux 2.2, vous auriez entré :

```
# Linux 2.2
# Faire suivre les paquets TCP destinés au port 8080 de 1.2.3.4 vers le port 80 de 192.168.1.1
ipmasqadm portfw -a -P tcp -L 1.2.3.4 8080 -R 192.168.1.1 80
```

Maintenant, vous devriez faire :

```
# Linux 2.4
# Ajouter une règle de pré-routage (-A PREROUTING) à la table de NAT (-t nat) qui
# prend les paquets TCP (-p tcp) destinés au port 8080 (--dport 8080) de 1.2.3.4 (-d 1.2.3.4)
# et les redirige (-j DNAT) vers le port 80 de 192.168.1.1 (--to 192.168.1.1:80).
iptables -A PREROUTING -t nat -p tcp -d 1.2.3.4 --dport 8080 \
        -j DNAT --to 192.168.1.1:80
```

5 Contrôler À Quoi Appliquer le NAT

Vous devez créer des règles de NAT qui disent au noyau quelles connexions modifier et comment les modifier. Pour cela, on utilise la souplesse de l'outil `Iptables`, en lui spécifiant de modifier la table de NAT avec l'option -t nat.

La table des règles de NAT contient 3 listes appelées 'chaînes' : chaque règle est examinée dans l'ordre jusqu'à ce que l'une d'elles convienne. Les trois chaînes sont nommées PREROUTING (pour le NAT de destination, lorsque le paquet arrive), POSTROUTING (pour le NAT de source, lorsque le paquet part) et OUTPUT. La troisieme chaîne (OUTPUT) sera ignorée pour l'instant.

Le diagramme suivant devrait illustrer ceci clairement si j'avais le moindre talent artistique :

```
         -----                                      -----
        /     \                                    /     \
```

```
PREROUTING -->[Décision]----------------->POSTROUTING----->
  \D-NAT/     [de routage]                 \S-NAT/
                  |                           ^
                  |                           |
                  |                           |
                  |                           |
                  |                           |
                  |                           |
                  |                           |
                  ------> Processus local ------
```

Lorsqu'un paquet traverse chacun des cercles ci-dessus, on observe la connexion à laquelle il est associé. S'il s'agit d'une nouvelle connexion, on examine la chaîne correspondante dans la table de NAT pour voir ce qu'il convient de faire du paquet. La réponse obtenue sera appliquée ensuite à tous les futurs paquets associés à cette connexion.

5.1 Sélection Simple Avec Iptables

`Iptables` accepte en standard de nombreuses options, listées ci-dessous. Toutes les options avec un double tiret peuvent être abrégées, à condition qu'`Iptables` puisse les différencier des autres. Si votre noyau contient le module de support d'Iptables, vous devrez charger le module iptables.o en premier : 'insmod ip_tables'.

L'option la plus importante ici est la sélection de table avec '-t'. Pour toutes les opérations de NAT, vous aurez à utiliser '-t nat' pour la table NAT. La seconde option indispensable est '-A' pour ajouter une nouvelle règle à la fin d'une chaîne (par exemple '-A POSTROUTING') ou '-I' pour en insérer une au début (par exemple '-I PREROUTING').

Vous pouvez spécifier l'adresse de source ('-s' ou '–source') et de destination ('-d' ou '–destination') du paquet que vous voulez transformer. Ces options peuvent être suivies par une seule adresse IP (par exemple 192.168.1.1), un nom (p.e. www.gnumonks.org), ou une adresse de réseau (par exemple 192.168.1.0/24 ou 192.168.1.0/255.255.255.0).

Pour la correspondance, vous pouvez spécifier une interface d'entrée ('-i' ou '–in-interface') ou de sortie ('-o' ou '–out-interface'), mais celle que vous avez le droit de spécifier dépend de la chaîne contenant la règle : dans PREROUTING, vous ne pouvez sélectionner qu'une interface d'entrée, et dans POSTROUTING (et OUTPUT) qu'une interface de sortie. Si vous vous trompez, `Iptables` vous renverra une erreur.

5.2 Affinage de la Sélection des Paquets à Marquer

J'ai mentionné plus haut que vous pouviez spécifier une adresse de source et de destination. Néanmoins, si vous omettez l'adresse de source, alors toutes les adresses de source conviendront. Et si vous omettez l'adresse de destination, alors toutes les adresses de destination conviendront.

Vous pouvez aussi spécifier un protocole ('-p' ou '–protocol'), comme TCP ou UDP ; seuls les paquets de ce protocole corresponderont à la règle. Le principal intérêt de spécifier un protocole TCP ou UDP est de disposer alors d'options supplémentaires : précisément '–source-port' et '–destination-port' (abbrégées en '–sport' et '–dport').

Ces options vous permettent de spécifier que seuls les paquets avec un certain port source ou destination corresponderont à la règle. C'est utile pour rediriger les requêtes web (port TCP 80 ou 8080) et laisser les autres paquets tranquilles.

Ces options doivent suivre l'option '-p' (qui a comme effet de charger l'extension de bibliothèque partagée pour ce protocole). Vous pouvez utiliser des numéros de ports ou un nom issu du fichier '/etc/services'.

Les différentes manières dont vous pouvez sélectionner un paquet sont détaillées dans la page de manuel (`man iptables`).

6 Définir Comment Marquer Les Paquets

Maintenant, nous savons comment sélectionner les paquets à marquer. Pour terminer notre règle, nous devons préciser au noyau ce qu'il doit faire des paquets.

6.1 NAT de Source

Vous voulez effectuer du NAT de source ; c'est-à-dire changer l'adresse source des connexions en quelque-chose d'autre. Ceci est réalisé dans la chaîne POSTROUTING, juste avant que le paquet ne soit définitivement envoyé à l'extérieur ; c'est une remarque d'importance, car elle signifie que toute autre fonction sur votre machine sous Linux (routage, filtrage de paquets) verra le paquet non modifié. Cela signifie aussi que l'option '-o' (interface de sortie) peut être utilisée.

Le NAT de source est spécifié en utilisant les options '-j SNAT', et '–to-source' qui spécifie une adresse IP, une plage d'adresses IP, et éventuellement un port ou une plage de ports (pour les protocoles UDP et TCP seulement).

```
## Changer l'adresse source en 1.2.3.4
# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4

## Changer l'adresse source en 1.2.3.4, 1.2.3.5 ou 1.2.3.6
# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4-1.2.3.6

## Changer l'adresse source en 1.2.3.4, port 1-1023
# iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to 1.2.3.4:1-1023
```

6.1.1 Camouflage

C'est un cas spécial de NAT de source appelé camouflage d'adresse IP : il devrait seulement être utilisé pour des adresses IP allouées dynamiquement, comme pour des connexions téléphoniques standardes (pour les adresses IP statiques, utilisez le SNAT ci-dessus).

Vous n'avez pas besoin de spécifier l'adresse source explicitement avec le camouflage : il utilisera l'adresse source de l'interface par laquelle le paquet sort. Mais plus important, si le lien est rompu, les connexions (qui sont de toute façon perdues) sont oubliées, ce qui évite des problèmes éventuels quand la connexion est rétablie avec une nouvelle adresse IP.

```
## Camoufler tout ce qui sort par ppp0
# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

6.2 NAT de Destination

Ceci est réalisé dans la chaîne PREROUTING, au moment où le paquet arrive ; cela signifie que toute autre fonction de votre machine sous Linux (routage, filtrage de paquets) verra le paquet aller vers sa destination 'réelle'. Cela signifie aussi que l'option '-i' (interface d'entrée) peut être utilisée.

Le NAT de destination est spécifié en utilisant '-j DNAT', et l'option '–to-destination' spécifie une adresse IP, une plage d'adresses IP, et éventuellement un port ou une plage de ports (pour les protocoles UDP et TCP seulement).

```
## Changer l'adresse de destination en 5.6.7.8
# iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 5.6.7.8

## Changer l'adresse de destination en 5.6.7.8, 5.6.7.9 ou 5.6.7.10
# iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 5.6.7.8-5.6.7.10

## Changer l'adresse de destination du trafic web en 5.6.7.8, port 8080
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth0 \
        -j DNAT --to 5.6.7.8:8080
```

6.2.1 Redirection

Il y a un cas spécialisé de NAT de destination appelé redirection : c'est une simple facilité qui est exactement équivalente à faire du DNAT vers l'adresse de l'interface d'entrée.

```
## Envoyer le trafic web entrant du port-80 vers notre mandataire (transparent) Squid
# iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 \
        -j REDIRECT --to-port 3128
```

Notez que Squid doit être configuré pour jouer le rôle de mandataire transparent !

6.3 Détail des Mises en Correspondance

Il y a plusieurs subtilités du NAT que la plupart des gens n'auront jamais à utiliser. Elles sont documentées ici pour les curieux.

6.3.1 Sélection d'Adresses Multiples Dans une Plage

Si une plage d'adresses IP est donnée, l'adresse choisie est basée sur l'adresse la moins utilisée, pour les connexions que la machine connaît. Ceci donne un équilibrage de charge grossier.

6.3.2 Ne Réaliser Aucune Correspondance de NAT

Vous pouvez utiliser la cible '-j ACCEPT' pour laisser passer une connexion sans réaliser de NAT.

6.3.3 Comportement Standard du NAT

Le comportement par défaut essaie de modifier la connexion le moins possible, en respectant les contraintes de la règle définie par l'utilisateur. Cela veut dire qu'on ne redirige pas les ports à moins d'y être forcé.

6.3.4 Correspondance Implicite de Port Source

Même quand aucun NAT n'est demandé pour une connexion, une transformation de port source peut être implicitement effectuée, si une autre connexion a déjà été engagée avant la nouvelle. Considérons le cas du camouflage qui est plutôt courant :

1. Une connexion web est établie par une machine 192.1.1.1 du port 1024 au port 80 de www.netscape.com.
2. Celle-ci est transformée par la machine effectuant le camouflage pour utiliser en adresse source sa propre adresse IP (1.2.3.4).
3. La machine effectuant le camouflage essaie de réaliser une connexion web vers le port 80 de www.netscape.com à partir de 1.2.3.4 (l'adresse de son interface externe) port 1024.
4. Le code du NAT va modifier le port source de la seconde connexion en 1025, pour éviter toute collision entre les deux connexions.

Quand cette correspondance implicite de source se produit, les ports sont répartis en 3 classes :
– Les ports inférieurs à 512
– Les ports entre 512 et 1023
– Les ports supérieurs à 1024.
Un port ne sera jamais implicitement mis en correspondance dans une autre classe que la sienne.

6.3.5 Que Se Passe-t'il Quand le NAT Echoue ?

S'il n'y a aucune façon de mettre en correspondance la connexion quand l'utilisateur le demande, elle sera abandonnée. Ceci s'applique aussi aux paquets ne pouvant être associés à une connexion, parce qu'ils sont malformés ou parce que la machine est saturée en mémoire, etc...

6.3.6 Mises en Correspondance Multiples, Recouvrements et Collisions

Vous pouvez avoir des règles de NAT qui mettent en correspondance des paquets sur la même plage ; le code du NAT est suffisamment malin pour éviter les collisions. Donc avoir deux règles qui mettent en correspondance les adresses sources 192.168.1.1 et 192.168.1.2 avec 1.2.3.4 fonctionnera.

De plus, vous pouvez mettre en correspondance des adresses IP réelles utilisées, à partir du moment où ces adresses passent par la machine effectuant cette mise en correspondance. Ainsi, si vous avez un réseau assigné (1.2.3.0/24), avec un réseau interne utilisant ces adresses et un autre utilisant des adresses privées (192.168.1.0/24), vous pouvez simplement faire du NAT des adresses sources 192.168.1.0/24 vers le réseau 1.2.3.0 sans crainte de collisions :

```
# iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 \
        -j SNAT --to 1.2.3.0/24
```

La même logique s'applique aux adresses utilisées par la machine effectuant elle-même le NAT : c'est comme cela que le camouflage fonctionne (en partageant l'adresse de l'interface entre des paquets camouflés et des paquets 'réels' venant de la machine elle-même).

De plus, vous pouvez mettre en correspondance les mêmes paquets sur différentes cibles et ils seront répartis. Par exemple, si vous ne voulez rien mettre en correspondance sur 1.2.3.5, vous pouvez utiliser :

```
# iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 \
        -j SNAT --to 1.2.3.0-1.2.3.4 --to 1.2.3.6-1.2.3.254
```

6.3.7 Modification de la Destination de Connexions Générées Localement

Le code du NAT vous permet d'insérer des règles de DNAT dans la chaîne OUTPUT, mais ceci n'est pas complètement soutenu dans le noyau 2.4 (il pourrait l'être, mais il nécessite une nouvelle option de configuration, quelques tests, et un bon morceau de code. Ainsi, à moins que quelqu'un n'engage Rusty pour l'écrire, je ne le prévois pas de sitôt).

La limitation actuelle vient du fait que vous ne pouvez modifier la destination que vers la machine locale (par exemple 'j DNAT –to 127.0.0.1'), et vers aucune autre machine, sinon les réponses ne seront pas traduites correctement.

7 Protocoles Spéciaux

Certains protocoles n'aiment pas être soumis au NAT. Pour chacun d'entre-eux, deux extensions doivent être écrites ; une pour le traçage de connexion du protocole et une pour le NAT lui-même.

Dans la distribution actuelle de Netfilter, il y a des modules pour FTP : ip_conntrack_ftp.o et ip_nat_ftp.o. Si vous les insérez dans votre noyau avec insmod (ou si vous les intégrez à la compilation), alors effectuer tout type de NAT sur une connexion FTP devrait fonctionner. Si vous ne le faites pas, alors seul le FTP passif sera utilisable, et encore, il pourrait ne pas fonctionner de façon fiable si vous réalisez plus que du simple NAT de source.

8 Avertissements sur le NAT

Si vous effectuez du NAT sur une connexion, tous les paquets passant **dans les deux sens** (vers l'intérieur et l'extérieur du réseau) doivent traverser la machine effectuant le NAT, sinon ça ne fonctionnera pas correctement. En effet, le code de traçage de connexion réassemble les fragments, ce qui veut dire que non seulement le suivi de connexion ne sera pas fiable, mais aussi que vos paquets pourraient ne pas passer du tout, puisque des fragments seront retenus.

9 NAT de Source et Routage

Si vous effectuez du SNAT, vous devriez vérifier que chaque machine qui reçoit des paquets modifiés par SNAT renverra les réponses à la machine de NAT. Par exemple, si vous mettez en correspondance des paquets sortants sur l'adresse source 1.2.3.4, alors le routeur extérieur doit savoir qu'il doit renvoyer les paquets de réponse (qui auront comme **destination** 1.2.3.4) à cette machine. Ceci peut être fait des manières suivantes :

1. Si vous effectuez du SNAT vers la propre adresse de la machine (pour laquelle le routage et le reste fonctionnent), vous n'avez rien à faire.
2. Si vous effectuez du SNAT sur une adresse inutilisée du réseau local (par exemple, vous mettez en correspondance sur 1.2.3.99, une adresse IP libre sur votre réseau 1.2.3.0/24), votre machine de NAT devra répondre aux requêtes ARP sur cette adresse autant que sur la sienne : la façon la plus facile de faire cela est de créer un alias IP, par exemple :

```
# ip address add 1.2.3.99 dev eth0
```

3. Si vous effectuez du SNAT vers une adresse complètement différente, assurez-vous que la machine atteinte par les paquets modifiés routera cette adresse vers la machine de NAT. Ceci est déjà réalisé si la machine de NAT est leur passerelle par défaut, sinon vous devrez publier une route (si vous utilisez un protocole de routage) ou ajouter manuellement les routes sur chaque machine concernée.

10 NAT de Destination Vers le Même Réseau

Si vous effectuez de la redirection de port vers le même réseau, vous devrez vous assurer que les paquets futurs et leurs réponses passeront par la machine de NAT (pour qu'ils soient modifiés). Le code du NAT (depuis le noyau 2.4.0-test6) bloquera le paquet de redirection ICMP sortant qui est généré lorsque le paquet modifié sort par l'interface par laquelle il est entré, mais le serveur de réception essaiera toujours de répondre directement au client (qui ne reconnaitra pas la réponse).

Dans le cas classique, les machines internes essaient d'accéder à votre serveur web 'public', qui en réalité est redirigé par DNAT de l'adresse publique (1.2.3.4) vers une machine interne (192.168.1.1), comme ceci :

```
# iptables -t nat -A PREROUTING -d 1.2.3.4 \
        -p tcp --dport 80 -j DNAT --to 192.168.1.1
```

Une solution est d'utiliser un serveur DNS interne qui connaît l'adresse IP réelle (interne) de votre site web public, et de rediriger toutes les autres requêtes vers un serveur DNS externe. Ceci signifie qu'une connexion locale sur le serveur web montrera les adresses IP internes correctement.

L'autre solution pour ces connexions est de forcer la machine de NAT à mettre en correspondance les adresses IP sources avec la sienne, trompant le serveur en répondant à travers lui. Dans cet exemple, nous devrions faire ceci (en supposant que l'adresse IP interne de la machine de NAT est 192.168.1.250) :

```
# iptables -t nat -A POSTROUTING -d 192.168.1.1 -s 192.168.1.0/24 \
        -p tcp --dport 80 -j SNAT --to 192.168.1.250
```

Comme la règle de **PREROUTING** est exécutée en premier, les paquets seront déjà destinés pour le serveur web interne : nous pouvons dire lesquels sont générés en interne grâce aux adresses IP sources.

11 Remerciements

Tout d'abord, je remercie WatchGuard et David Bonn, qui ont cru à l'idée de Netfilter suffisamment pour me soutenir lorsque je travaillais dessus.

Je remercie aussi tous ceux qui ont été patients avec mes extravagances, alors que j'apprenais la laideur du NAT, et je remercie spécialement ceux qui lisent mon journal.

Rusty.

12 Commentaires et Corrections

Merci de faire parvenir en anglais à l'auteur vos questions et commentaires relatifs à la version originale de ce document à l'adresse *netfilter@lists.samba.org* .

N'hésitez pas à faire parvenir tout commentaire relatif à la version française de ce document à *commentaires CHEZ traduc POINT org*

en précisant le titre et la version de ce document.