

Sommaire

I.Rappels.....	1
a)Les différents types de filtrages : les tables.....	1
b)Fonctionnement de base : les chaînes et les règles.....	1
II.La table nat : utilisation en passerelle.....	2
a)Architecture.....	2
b)Décision de routage.....	3
c)Etapas de traversée.....	3
III.Activation du NAT.....	5
IV.Une configuration de base.....	5
a)EXEMPLE.....	7
b)Configuration de l'interface du client.....	7
c)Configuration de l'interface de la passerelle.....	7
d)Configuration iptables de la passerelle pour les connexions sortantes.....	8
e)Configuration iptables de la passerelle pour les connexions entrantes.....	8
f)Amélioration de la chaîne POSTROUTING.....	9
I.Bibliographie.....	9

I. Rappels

a) Les différents types de filtrages : les tables

Une table contient des chaînes relatives au filtrage qu'elle réalise.

Il existe 3 tables distinctes :

- filter (Tables de filtrage) : c'est la table par défaut. Elle sert pour les entrées/sorties/traversées sur la machine.
- nat (translation d'adresse) : elle sert pour le gérer le changement d'adresses IPs et de ports
- mangle (table spécifique) : elle sert pour gérer

Dans la table de filtrage (FILTER) on peut filtrer les paquets avec trois chaînes :

- Paquets entrants (INPUT) (vers des applications) (client ou server)
- Paquets sortants (OUTPUT) (émis par des applications) (client ou server)
- Paquet passant par le firewall (FORWARD) (passerelle)

Dans la table de translation (NAT) on peut gérer les paquets (dans le cas d'une passerelle) :

- Entrant dans le firewall (PREROUTING) : DNAT
- Sortant du firewall (POSTROUTING) : SNAT

b) Fonctionnement de base : les chaînes et les règles

Iptables est basé sur des chaînes de pare-feu (ou simplement chaîne). Une chaîne est un ensemble ordonné (une liste) de règles. Une règle indique quoi faire d'un paquet quand il a certaines

caractéristiques.

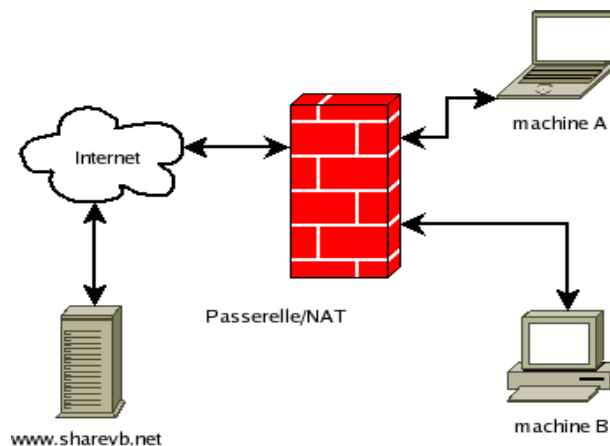
Ainsi, un paquet traverse les règles d'une chaîne jusqu'à ce qu'il corresponde à une règle. Dans ce cas, la règle indique si le paquet est transmis à sa destination (ACCEPT) ou supprimer (DROP).

Dès qu'une règle (autre que LOG) capte un paquet, elle prend une action sur le paquet et la parcourt de la chaîne s'arrête.

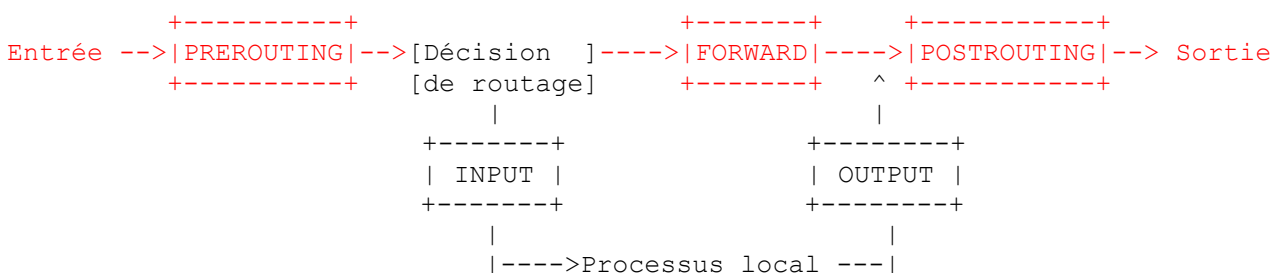
Si le paquet ne correspond à aucune règle alors on applique la police par défaut de la chaîne. Si elle est à ACCEPT, tout paquet non interdit sera délivré à sa destination. Si elle est à DROP, tout paquet non autorisé sera supprimé (Cette solution est la plus sûr).

II. La table nat : utilisation en passerelle

a) Architecture



Le fonctionnement de la table NAT (et FILTER) dans le cas d'une passerelle est le suivant :



La traversée de la passerelle par un paquet est indiquée en rouge.

La requête et la réponse passent, dans l'ordre, par PREROUTING puis FORWARD puis POSTROUTING.

b) Décision de routage

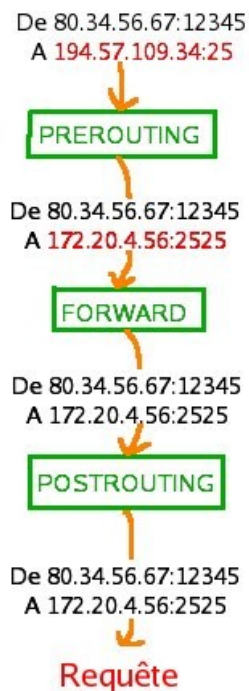
La décision de routage se fait en regardant si l'adresse IP de destination est celle de l'interface par laquelle le paquet est entré :

- si l'adresse de destination est celle d'une entrée de la machine, le paquet est pour la machine (serveur ou client) : le paquet passe dans la chaîne INPUT
- si le paquet vient d'une application : le paquet créé passe dans la chaîne OUTPUT
- **sinon le paquet ne fait que passer par la machine (passerelle) (adresse destination différente de l'interface entrante d'où vient le paquet): chaîne FORWARD. Si le forwarding/mapping n'est pas autorisé alors le paquet est détruit.**

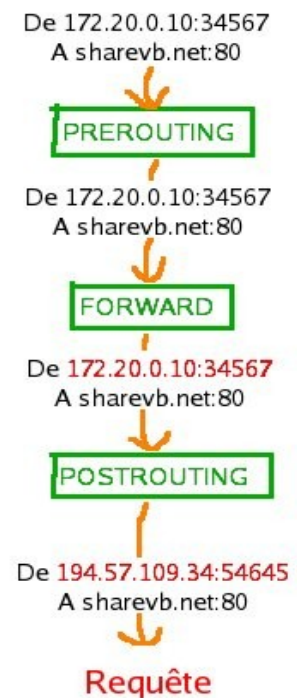
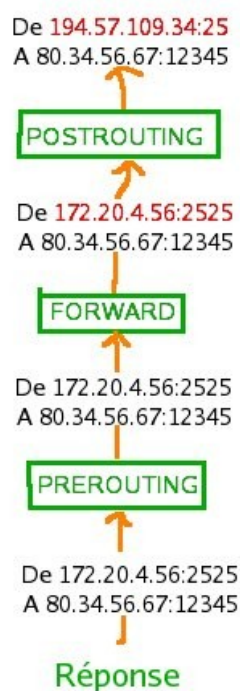
Dans le cas d'une simple passerelle, on n'utilisera que la chaîne FORWARD.

c) Etapes de traversée

Connexion entrante



Connexion sortante



Les étapes de parcourt des chaînes par un paquet sont les suivantes :

- le paquet passe dans la chaîne PREROUTING : cette chaîne permet de modifier l'adresse de destination du paquet (Port Forwarding/DNAT statique) : **Cela s'applique dans le sens extérieur -> intérieur. Pour les réponses du même sens, la table NAT assure toute seule le DNAT automatique.**
 - on peut décider de mapper un port n de la passerelle sur le port m d'un serveur interne (n et m pas forcément différents = forwarding). Ainsi, l'utilisateur aura l'impression de se connecter à la passerelle sur le port n alors qu'il sera, en fait, connecté au serveur interne sur le port m .

```
iptables -t nat -A PREROUTING -i interface_entrée -d IP_passerelle
```

```
-p tcp|udp -dport port_n [autres options] -j DNAT --to-destination serveur_interne:port_n
```

- on peut utiliser l'option *-i interface* pour faire des mapping/forwarding de ports différents suivant l'interface d'entrée
- on peut décider de forwarder tous les ports de la passerelle sur une machine interne

```
iptables -t nat -A PREROUTING -i interface_entrée -d IP_passerelle [autres options] -j DNAT --to-destination serveur_interne
```

- par exemple, si vous avez une passerelle connectée à Internet et un serveur web interne que vous voulez rendre publique, vous utiliserait un DNAT du port 80 de la passerelle sur le port 80 du serveur interne.
- **Lorsqu'il s'agit de réponse à une requête, l'adresse destination est automatiquement remplacée par le NAT dynamique par l'adresse de la source de la requête. Les paquets dans la chaîne suivante ont donc toujours une adresse de destination du réseau d'où est partie la requête.**
- Le paquet passe dans la chaîne FORWARD : cette chaîne sert à autoriser ou interdire le passage des paquets par la passerelle **dans les deux sens** :
 - il faut (pour l'exemple, mais on peut être plus restrictif) autoriser les connexions établies dans les deux

```
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

- autoriser un mapping de port : *interface_entrée* publique et *interface_sortie* LAN

```
iptables -A FORWARD -i interface_entrée -d serveur_interne_LAN -p tcp|udp -dport port_m -m state --state NEW -j ACCEPT
```

- fermer un mapping de port : même chose mais avec DROP
- interdire certaines connexions sortantes (port *n*)

```
iptables -A FORWARD -i interface_LAN -s adresse(s)_LAN/masque -p tcp|udp -dport port_n -j DROP
```

- interdire l'accès au web (cas particulier du précédent)...

```
iptables -A FORWARD -i interface_LAN -s adresse(s)_LAN/masque -p tcp -dport 80 -j DROP
```

- le paquet passe dans la chaîne POSTROUTING : cette chaîne sert à changer l'IP source du paquet par l'IP une certaine interface. (NAT dynamique/SNAT). **Cela s'applique dans le sens intérieur -> extérieur. Pour les réponses du même sens, la table NAT assure toute seule le SNAT automatique.**
 - Par exemple, si l'on a une machine A qui partage une connexion internet, toutes les machines du réseau local se connecteront à Internet à travers A. Cependant, une fois sortis de A par la connexion Internet, les paquets doivent porter comme adresse source, l'adresse IP de la connexion Internet sans quoi le serveur destinataire ne saurait pas à qui répondre (car les IP LAN ne sont pas routables).
 - Il existe deux cas :
 - on connaît à l'avance l'adresse IP de l'interface de sortie des paquets. On utilise la cible SNAT :

```
iptables -t nat -A POSTROUTING -o interface_de_sortie -j SNAT --
```

```
to-source IP_de_sortie
```

- on ne connaît pas l'adresse IP de l'interface de sortie des paquets. On utilise la cible MASQUERADE:

```
iptables -t nat -A POSTROUTING -o interface_de_sortie -j  
MASQUERADE
```

III. Activation du NAT

Enfin, il faut activer l'IP forwarding par : `echo "1" > /proc/sys/net/ipv4/ip_forward`

IV. Une configuration de base

On va s'intéresser à la mise en place d'une passerelle/firewall et donc la chaîne FORWARD de la table FILTER et la table NAT. Nous allons par exemple nous intéresser à l'architecture suivante :

```
Machine 1----->Passerelle/firewall----->Internet  
192.168.0.10      192.168.0.1 - 197.24.19.1      Toutes les IPs
```

Notre passerelle doit donc autoriser un certain nombre de paquets à passer, mais aussi effectuer ce que l'on appelle une translation d'adresse, c'est à dire remplacer l'adresse IP du client avec la sienne avant d'envoyer le paquet sur internet.

Il faut d'abord activer la fonction "passerelle" dans le noyau, pour cela il faut utiliser la commande suivante :

```
[root@passerelle ~]# echo "1" > /proc/sys/net/ipv4/ip_forward
```

Il faut dans un deuxième temps autoriser certaines connexions à "traverser" notre passerelle, le plus simple (et le plus dangereux !!!) est d'utiliser la commande suivante :

```
[root@passerelle ~]# iptables -P FORWARD ACCEPT
```

Nous allons maintenant nous occuper de la translation d'adresses, pour consulter la table nat, on utilise la commande suivante :

```
[root@passerelle ~]# iptables -L -t nat  
Chain OUTPUT (policy ACCEPT)  
target      prot opt source                destination  
  
Chain POSTROUTING (policy ACCEPT)  
target      prot opt source                destination  
  
Chain PREROUTING (policy ACCEPT)  
target      prot opt source                destination
```

Il faut donc ajouter la règle de "masquering" avec la commande suivante :

```
[root@passerelle ~]# iptables -t nat -A POSTROUTING -j MASQUERADE
```

En effet, les modifications s'effectuent sur les paquets sortants !! (POSTROUTING)
Pour vérifier :

```
[root@passerelle ~]# iptables -L -t nat
Chain OUTPUT (policy ACCEPT)
target          prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target          prot opt source                destination
MASQUERADE     all  --  anywhere              anywhere

Chain PREROUTING (policy ACCEPT)
target          prot opt source                destination
```

A partir de ce moment, le client doit pouvoir se connecter sur internet.

Nous voulons maintenant pouvoir utiliser à partir d'internet, un serveur web qui se trouve sur le client sur le port 80, il faut donc réaliser ce que l'on appelle du "forwarding de port", c'est à dire que toutes les connexions entrantes sur le firewall vers le port 80, seront redirigées et translatées vers le client sur le port 80, pour cela on utilisera la commande suivante :

```
[root@passerelle ~]# iptables -t nat -A PREROUTING -d 197.24.19.1
-p tcp --dport 80 -j DNAT --to-destination 192.168.0.10:800
```

En effet, il faut effectuer la transformation en amont du firewall (PREROUTING), à destination du firewall (-d 197.24.19.1) vers le client sur le port 80 (--to-destination 192.168.0.10:800) et effectuer une modification de l'adresse IP (-DNAT : Destination NAT).

a) EXEMPLE

Soit la configuration suivante :

- client avec l'IP $\{\text{IP_CLIENT}\}$ sur un réseau $\{\text{ADR_RES_INT}\}$
- passerelle avec les deux IPs $\{\text{IP_PASS_INT}\}$ et $\{\text{IP_PASS_EXT}\}$.

Par exemple :

- $\{\text{IP_CLIENT}\}$: 192.168.12.34
- $\{\text{ADR_RES_INT}\}$: 192.168.0.0/24
- $\{\text{IP_PASS_INT}\}$: 192.168.12.10
- $\{\text{IP_PASS_EXT}\}$: 172.20.12.34
- $\{\text{MASQUE_INT}\}$: 255.255.255.0
- $\{\text{MASQUE_EXT}\}$: 255.255.0.0
- $\{\text{PASS_INT_INTERF}\}$ = eth0
- $\{\text{PASS_EXT_INTERF}\}$ = eth1

b) Configuration de l'interface du client

Sur le client $\{\text{IP_CLIENT}\}$:

Dans le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0` :

```
DEVICE=interface_client
BOOTPROTO=none
```

```
HWADDR=xx:xx:xx:xx:xx:xx
ONBOOT=yes
TYPE=Ethernet
IPADDR=${IP_CLIENT}
NETMASK=${MASQUE_INT}
GATEWAY=${IP_PASS_INT}
```

c) Configuration de l'interface de la passerelle

Sur la passerelle `${IP_PASS_EXT}/${IP_PASS_INT}` :

Dans le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0` :

```
DEVICE=${PASS_EXT_INTERF}
BOOTPROTO=none
HWADDR=xx:xx:xx:xx:xx:xx
ONBOOT=yes
TYPE=Ethernet
IPADDR=${IP_PASS_EXT}
NETMASK=${MASQUE_EXT}
```

Dans le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0:0` :

```
DEVICE=${PASS_INT_INTERF}
BOOTPROTO=none
HWADDR=xx:xx:xx:xx:xx:xx
ONBOOT=yes
TYPE=Ethernet
IPADDR=${IP_PASS_INT}
NETMASK=${MASQUE_INT}
```

d) Configuration iptables de la passerelle pour les connexions sortantes

Il faut configurer le firewall de manière "propre", c'est à dire que l'on interdit TOUT par défaut, et l'on n'autorise que ce qui est vraiment nécessaire, aussi bien dans la chaîne FORWARD, que dans la translation d'adresse.

```
#activer l'ip forwarding
echo "1" > /proc/sys/net/ipv4/ip_forward

# tout interdire par défaut
iptables -P FORWARD DROP

#on autorise les clients à se connecter au web par la passerelle
iptables -A FORWARD -i ${PASS_INT_INTERF} -o ${PASS_EXT_INTERF} -s
${ADR_RES_INT} --protocol tcp --destination-port 80 -j ACCEPT

#on autorise les clients à la résolution de noms par la passerelle
iptables -A FORWARD -i ${PASS_INT_INTERF} -o ${PASS_EXT_INTERF} -s
${ADR_RES_INT} -p udp --dport 53 -j ACCEPT
iptables -A FORWARD -i ${PASS_EXT_INTERF} -o ${PASS_INT_INTERF} -d
${ADR_RES_INT} -p udp --sport 53 -j ACCEPT
```

```

#on autorise les clients à se connecter en ssh par la passerelle
iptables -A FORWARD -i _${PASS_INT_INTERF}_ -o _${PASS_EXT_INTERF}_ -s
_${ADR_RES_INT}_ --protocol tcp --destination-port 22 -j ACCEPT

#on autorise les connexions établies
iptables -A FORWARD -i _${PASS_EXT_INTERF}_ -o _${PASS_INT_INTERF}_ -d
_${ADR_RES_INT}_ -m state --state ESTABLISHED,RELATED -j ACCEPT

#active le NAT dynamique entre le réseau local _${ADR_RES_INT}_ et l'extérieur
_${ADR_RES_EXT}_
iptables -t nat -A POSTROUTING -o _${PASS_EXT_INTERF}_ -j MASQUERADE

```

e) Configuration iptables de la passerelle pour les connexions entrantes

Une connexion sur le port 23 du firewall sera en fait redirigée sur le port 22 (ssh) du client.

```

#on active le DNAT de firewall:23 sur client:22
iptables -t nat -A PREROUTING -i _${PASS_EXT_INTERF}_ -d _${IP_PASS_EXT}_ -p tcp
-dport 23 -j DNAT --to-destination _${IP_CLIENT}:22

#on autorise l'extérieur à se connecter sur le port 23 du client
iptables -A FORWARD -i _${PASS_EXT_INTERF}_ -o _${PASS_INT_INTERF}_ -d
_${IP_PASS_INT}_ --protocol tcp --destination-port 22 -j ACCEPT

```

f) Amélioration de la chaîne POSTROUTING

Normalement, la cible MASQUERADE est utilisée uniquement si l'une des adresses IP du firewall est attribué dynamiquement, sinon on utilisera plutôt la cible SNAT de la table NAT, en POSTROUTING, qui permet de spécifier directement l'adresse IP à utiliser en sortie, et donc de "consommer" moins de ressources. Apportez les modifications nécessaires pour utiliser maintenant ce mécanisme à la place du MASQUERADE.

```

#active le NAT dynamique entre le réseau local et l'extérieur
iptables -t nat -A POSTROUTING -o _${PASS_EXT_INTERF}_ -j SNAT --to-source
_${IP_PASS_EXT}_

```

I. Bibliographie

[netfilter/iptables project homepage - The netfilter.org project](#)