

0.1 Firewall et partage de connexion : Iptables

Avant d'aller plus loin, je rappelle que votre distribution est probablement livrée avec un outil permettant de paramétrer un firewall + partage de connexion internet. Si ce n'est pas votre cas, ou si vous désirez savoir comment ça marche, lisez la suite. Je ferai, la part belle à la configuration manuelle, bien qu'il existe des outils graphiques pour cela.

Lorsque vous disposez d'une connexion unique à Internet à partager à plusieurs, vous avez globalement le choix entre 2 stratégies : un proxy ou le masquerading (nat : Network Address Translation). Même sans partager une connexion (à fortiori si vous en partagez une), il peut-être, sécurisant d'avoir un firewall à domicile.

Un proxy est un mandataire, lorsque votre ordinateur serveur fait proxy, cela signifie que les clients ne se connectent pas directement à Internet, mais demande au proxy de télécharger pour eux les pages dont ils ont besoin. Un proxy quelqu'il soit ne couvre qu'une gamme limitée de protocole, généralement http et ftp.

Le masquerading (nat) est une translation d'adresse source, c'est à dire qu'il remplace les adresses source des paquets d'un réseau local, par l'IP de la passerelle. Il conserve cependant des traces des transactions pour acheminer vers chacun le paquet qui lui est destiné. Ainsi, toutes vos machines apparaissent sur Internet comme une seule et même machine.

Sous Linux le masquerading et le firewall se font à partir d'un seul et même couple de logiciels : Netfilter/Iptables. Ces 2 logiciels sont déjà installés ou du moins présents sur vos cdroms. S'ils ne sont pas installés, il suffit généralement d'installer le paquet : iptables. Tapez : iptables - L dans un terminal, il devrait vous afficher les règles courantes.

Netfilter/Iptables forment un couple infernal, qui fournit une solution complète pour faire du firewalling, du nat(partage de connexion) et du mangle. Netfilter est directement intégré au noyau 2.6, tandis que iptables est une commande qui permet de gérer les règles de son firewall.

Notez enfin que le Nat ne se limite pas qu'au masquerading (partage de connexion) mais permet également de faire l'opération inverse : translation d'adresse de destination.

0.1.1 1. Configuration de Netfilter :

0.1.2 1.1 Compilation du noyau

Si vous êtes l'heureux utilisateur d'une Mandriva 10 et plus, Debian (avec noyau 2.6), Slackware 10 et plus avec noyau 2.6 et Fedora core 2 et plus, vous pouvez passer au 1.2. Pour les autres ou pour ceux qui veulent comprendre comment ça marche, assurez-vous que votre noyau est compilé avec les options suivantes :

```
Device Drivers ->
[*] Networking support ->
Networking options ->
<M> Packet socket
[*] Packet socket : mmaped IO
[*] Network packet filtering (replaces ipchains) -->
[*] Network packet filtering debugging
IP : Netfilter Configuration -->
<M> Connection tracking (required for masq/NAT)
<M> FTP protocol support
```

```
<M> IRC protocol support
<M> TFTP protocol support
<M> IP tables support (required for filtering/masq/NAT)
<M> limit match support
<M> TCPMSS target support
<M> Connection state match support
<M> Packet filtering
<M> REJECT target support
<M> ULOG target support
<M> Full NAT
<M> MASQUERADE target support
<M> LOG target support
```

0.1.3 1.2 Chargement des modules

Chargez maintenant les paramètres en module (voir `/lib/modules/votre_noyau/kernel/net/ipv4/netfilter`). Concrètement tapez :

```
# modprobe ip_tables
# modprobe ip_nat_ftp
# modprobe ip_nat_irc
# modprobe iptable_filter
# modprobe iptable_mangle
# modprobe iptable_nat
```

Pour ne plus avoir à le faire manuellement, rajoutez ces lignes à la fin de votre `/etc/rc.d/rc.local`. Une autre solution, consiste bien-sûr à utiliser l'outil de votre distribution pour qu'ils soient chargés à chaque amorçage.

0.1.4 2. Théorie sous-jacente à Iptables :

Je fournis dans le 3/ un script "près à l'emploi" qui n'est pas une panacée mais permet d'avoir un firewall fonctionnel et évolutif. Il est abondamment commenté, pour faciliter, la compréhension, néanmoins un peu de généralité ne vous fera pas de mal, je pense. Iptables manipule 3 tables : la table filter, la table nat et la table Mangle. Une table est formée de chaîne par défaut, auxquelles il faut rajouter celles que vous créez. Pour chaque chaîne, il faut définir une politique par défaut, puis rajouter des règles pour gérer les cas particuliers.

- Voyons d'abord les tables qui nous sont proposées :
 - **La table Filter :**
 - **INPUT** : c'est la chaîne par laquelle passent tous les paquets entrant par une interface.
 - **FORWARD** : c'est la chaîne par laquelle transitent les paquets qui traversent la machine d'une interface à une autre.
 - **OUTPUT** : c'est la chaîne par laquelle passent les paquets qui sortent par une interface.
 - **La table Nat :**

- **PREROUTING** : chaîne qui permet de faire de la translation d'adresse de destination. C'est ce qui permet par exemple de faire croire à vos clients qu'il y'a un serveur ftp sur le port 21 de votre passerelle alors qu'il est hébergé en réalité sur un autre Pc écoutant sur le port 2021.
- **POSTROUTING** : C'est grâce cette chaîne que vous pourrez faire du masquerading(partage de connexion) et faire croire à tous sur Internet que votre réseau n'a qu'une unique IP, celle de la passerelle.
- **OUTPUT** : Celle-ci va permettre de modifier la destination de paquets générés localement (par la passerelle elle-même).
- **la table Mangle** : qui permet de marquer et/ou modifier des paquets à la volée. Ceci sert par exemple à optimiser des transactions par FTP.
- Ensuite pour chaque règle, énoncée il est possible d'appliquer une politique :
 - **ACCEPT** on laisse passer le paquet.
 - **DROP** on ignore le paquet.
 - **REJECT** on rejette le paquet et on envoie un message d'erreur. Elle n'est utilisable que dans les chaînes INPUT, FORWARD et OUTPUT.
 - **QUEUE** on envoie le paquet à un programme utilisateur capable de communiquer avec NetFilter
 - **RETURN** pour sortir de la chaîne immédiatement, ou appliquer la règle de la politique par défaut pour les chaînes prédéfinies
 - **LOG** on enregistre une notification du paquet dans syslog
 - **MASQUERADE** pour effectuer une translation d'adresse sur ce paquet, dans le but de réaliser un partage de connexion à Internet. Cette politique n'est accessible que dans la chaîne POSTROUTING de la table nat.
 - **SNAT** pour modifier l'adresse source du paquet.
 - **DNAT** pour modifier l'adresse du destinataire du paquet.
- Voici également quelques unes des options que l'on peut passer à Iptables :
 - **-N** création d'une nouvelle chaîne
 - **-X** suppression d'une chaîne vide
 - **-P** changement de politique par défaut
 - **-L** liste des chaînes courantes
 - **-F** Elimination de toutes les règles d'une chaîne
 - **-Z** remise à zéro des compteurs
 - **-A** ajoute une règle à la fin d'une chaîne
 - **-I** insère une nouvelle règle à une position donnée
 - **-R** remplace une règle donnée dans une chaîne donnée
 - **-D** efface une règle.
- Enfin, les commandes pour matcher :
 - **-p** on spécifie le protocole : icmp, udp, tcp ou all
 - **-s** on spécifie la source à matcher, généralement une adresse ou une classe d'adresse
 - **-d** on spécifie la destination, généralement une adresse ou une classe d'adresse
 - **-i** on spécifie l'interface d'entrée : eth0, ppp0 ...
 - **-o** on spécifie l'interface de sortie : eth0, ppp0 ...
 - **-t** on spécifie la table à laquelle on fait référence : filter, nat, mangle
 - **-sport** on spécifie le port source, il peut s'agir du numéro de port (21, 22 ...) ou du protocole (ftp, ssh ...). Pour la correspondance voir /etc/services
 - **-dport** on spécifie le port de destination, il peut s'agir du numéro de port (21, 22

- ...) ou du protocole (ftp, ssh). Pour la correspondance voir /etc/services
- **-state** on spécifie l'état, **ESTABLISHED** (connexion déjà établie), **NEW** (nouvelle connexion), **INVALID** (connexion inconnue), **RELATED** (Nouvelle connexion mais liée à une existante).

Tout ceci n'est évidemment pas exhaustif, mais vous permettra, de construire un firewall, rapidement.

0.1.5 3. Pratique de Iptables :

On peut assigner les règles à iptables à la volée, mais il est bien plus pratique de tout regrouper dans un fichier qui sera lu au démarrage de la machine par exemple.

La politique la plus conseillée, est la suivante : tout interdire sans exception, du moins tout ce qui rentre. Puis accepter au cas par cas certaines choses indispensables. Je vais supposer dans la suite que mon PC sur lequel je monte le firewall a pour adresse locale 192.168.0.1 et donc fait partie du réseau 192.168.0.x.

Mon réseau est formé d'une passerelle, sur laquelle tournent mon firewall, ainsi que mes serveurs apache, proftp, ssh, bind, donkey, samba ... C'est donc lui qui est connecté à Internet et partage sa connexion par nat avec les clients Windows ou Linux.

Mes clients (Windows ou Linux), accèdent donc à Internet de manière transparente, avec des clients : mail, web, ftp, donkey ... tout en étant protégés par le firewall. Est-ce utile de préciser que les firewalls ne protègent pas des virus ? Le script est consultable ICI¹. Le firewall base sa protection sur les interfaces réseau, les adresses sources et destination des paquets ainsi que les protocoles. Il y'a moyen de faire encore plus fins, consultez la documentation officielle. Pour lancer le script, tapez :

```
cd /où_est_le_script/
./firewall start
```

Il accepte, un certains nombre d'arguments, start pour démarrer, stop pour l'arrêter, restart pour le redémarrer et status pour voir les règles en cours.

0.1.6 4. Lancement du firewall au démarrage :

Pour lancer le firewall à l'amorçage de la machine, il vous suffit de le mettre dans /usr/bin/ et de le rendre exécutable, en root, par :

```
chmod 744 /usr/bin/firewall
```

Finissez en rajoutant les lignes qui suivent dans votre /etc/rc.d/rc.local (après les lignes de chargement de module, bien-sûr) :

```
if [ -x /usr/bin/firewall ]; then
/usr/bin/firewall start
fi
```

Au prochain démarrage, le firewall sera lancé.

0.1.7 5. Configurer les clients pour le NAT :

Ce qui suit concerne les personnes, qui font du masquerading (partage de connexion) en plus du firewall. Si votre serveur, fait proxy ou ne fait pas de masquerading, passez au 6/

¹<http://file.truostonme.net/data/firewall>

- **Clients Linux** : Il vous suffit de taper la ligne suivante, en root dans un terminal :
`/sbin/route add -net default gw IP_de_La_Passerelle netmask 0.0.0.0 metric 1` Ensuite vous devez, modifier votre `/etc/resolv.conf` pour qu'il ressemble à ceci :

```
nameserver DNS_1_de_votre_FAI
nameserver DNS_2_de_votre_FAI
```

Pour wanadoo, les DNS 1 et 2 sont : 193.252.19.3 et 193.252.19.4

- **Clients Windows** : La configuration est en tout point similaire à celle que vous auriez eue, à faire avec un nat géré par sygate, par exemple. Concrètement, rendez-vous dans le "**voisinage réseau**" (ou "**connections réseaux et accès à distance**"). Là choisissez votre carte réseau et allez dans "**propriétés**". Double-cliquez sur "**Protocole Internet (TCP/IP)**". Là comme "passerelle par défaut" mettez l'IP de la passerelle. Comme "**DNS préféré**" mettez le DNS 1 de votre FAI, comme DNS auxiliaire le DNS 2 de votre FAI. Selon votre version de Windows, il se peut que vous ayez à valider par "ajouté" à chaque onglet. Une fois satisfait validez et c'est bon.

0.1.8 6. Liens et conclusion :

Quelques liens vers de la documentation sur Iptables :

- Site Officiel ²
- HOWTO NAT avec iptables ³
- HOWTO pour netfilter ⁴
- Pour faire mon firewall, j'ai lu entre autre cet Article ⁵
- Interface graphique pour la création de scripts iptables ⁶

quel que soit, le niveau de votre firewall, un firewall n'est pas une fin en soi. C'est juste un maillon (fort) dans une politique de défense contre des personnes malveillantes. Un peu de bon sens permet souvent de limiter pas mal de casse.

²<http://www.netfilter.org/>

³<http://www.linux-france.org/prj/inetdoc/guides/NAT-HOWTO/>

⁴<http://www.linux-france.org/prj/inetdoc/guides/netfilter-hacking-HOWTO/>

⁵<http://christian.caleca.free.fr/netfilter/index.htm>

⁶<http://www.fwbuilder.org>