

**PARE-FEU SOUS LINUX.**  
**<http://nassih.com>**  
**© MOHAMED NASSIH.**

## **Introduction.**

Dans un réseau d'entreprise il est primordial de mettre en place les mécanismes de sécurité nécessaires pour contrôler et limiter l'utilisation des ressources. Le pare-feu est l'une des composantes nécessaires et essentielles d'une politique de sécurité. De nombreux vendeurs offrent des solutions de pare-feu matérielles comme Cisco et Nokia, et logiciels comme Checkpoint. Dans notre cas nous allons étudier la solution Netfilter qu'est une solution complète de pare-feu sous Linux. Nous commençons par l'explication de quelques notions et concepts de base. Puis nous détaillons la solution Netfilter/Iptables,

### **1. Définitions et concepts de base.**

#### **1.1. Politique de sécurité.**

Une politique de sécurité est une approche intégrée qui désigne l'ensemble des règles et des orientations mises en place pour protéger un système d'information contre les risques et les vulnérabilités connus. Les règles et les procédures de sécurité doivent être clairement définies et écrits dans un manuel facilement consultable. Ce manuel devra contenir des informations sur les utilisateurs des systèmes, sur leurs droits d'accès, sur la méthode de sauvegarde, sur la méthode de restauration du système en cas de problème, sur la mise à jour de l'anti-virus, sur la configuration des pare-feu, sur la configuration du système de détection d'intrusion, sur les rapports produits par ces systèmes et sur les personnes ressources qui doivent interpréter ces rapports et réagir en cas de besoin, et en fin sur tout autre information jugée importante en matière de sécurité.

#### **1.2 Pare-feu.**

Un pare-feu est un système conçu pour superviser le flux d'information entre deux ou plusieurs réseaux. Cette supervision se fait en utilisant des règles de filtrage. Une règle est une ou plusieurs instructions auxquelles un paquet va correspondre ou non. Le pare-feu analyse donc

l'en-tête de chaque paquet. Le filtrage pourra se faire à base des adresses IP, des adresses physiques, du protocole de la couche transport ou à base des ports de la couche application. Le filtrage pourra aussi se faire à base de l'état de la connexion en cours, dans ce cas le pare-feu va voir si le paquet appartient à une connexion déjà établie ou non.

Deux stratégies peuvent être s'appliquées, la première est d'empêcher les communications qui ont été explicitement interdites, et la deuxième stratégie est d'autoriser uniquement les communication ayant été explicitement autorisées. Dans notre configuration Nous allons utiliser la deuxième stratégie, puisqu'elle permet de bien maîtriser les accès.

### **1.3. La translation d'adresse.**

Pour résoudre le problème de manque d'adresses Ipv4, le RFC1918 a défini trois intervalles d'adresses privés non routable sur Internet à utiliser dans les réseaux privés internes.

10.0.0.0	-	10.255.255.255
172.16.0.0	-	172.31.255.255
192.168.0.0	-	192.168.255.255

**Figure 1.1 : les intervalles d'adresses privées défini par RFC1918.**

La figure 1.1 montre les trois plages d'adresses définis dans le RFC1918. Pour que ces machines communiquent avec l'Internet - avec un réseau externe-, la translation d'adresse est donc utilisée.

D'une part, la translation d'adresse permet à un réseau local avec plusieurs machines à se connecter à Internet en utilisant seulement une seule adresse publique, dans ce cas on parle de la translation d'adresse source. D'autre part elle permet aux utilisateurs externes d'accéder à des serveurs avec des adresses privées sur le réseau local interne, on parle dans ce cas de la translation d'adresse de destination. La translation d'adresse permet aussi de cacher les machines internes derrière une même adresse publique, il est donc plus difficile pour un pirate externe d'avoir des informations sur l'adressage du réseau interne. La machine -ou la passerelle- qui

s'occupe de la translation d'adresse gère une table de translation d'adresse qui garde la correspondance entre les adresses privées et les adresses publiques.

#### **1.4. Zone démilitarisée.**

Un réseau local d'entreprise est composé en général des stations de travail pour les utilisateurs internes et des serveurs. Ces serveurs peuvent être utilisés par les utilisateurs du réseau local ou bien par des utilisateurs externes. Pour isoler les machines du réseau local, une zone démilitarisée est utilisée. C'est une zone tampon entre le réseau interne et externe. Le but d'une zone démilitarisée est d'éviter les connexions directes vers le réseau interne. Il héberge souvent des serveurs avec les services web les plus standards comme un serveur web ou un serveur de messagerie. En général le trafic du réseau externe est autorisé vers la zone démilitarisée, mais pas vers le réseau local.

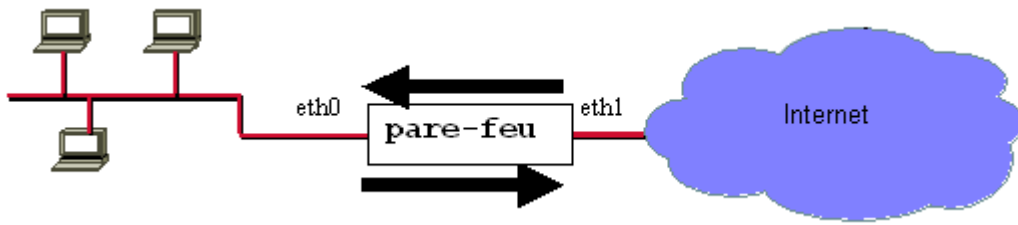
#### **1.5. Netfilter.**

Netfilter est une solution complète de pare-feu tournant sous Linux. Netfilter permet de faire du filtrage à états, de la translation de port et d'adresse et du filtrage des paquets. IpTables est la commande qui permet la configuration du Netfilter, c'est une interface très fiable et se dispose de très nombreuses options qui permettent de faire du filtrage très fin. Netfilter/IpTables est intégré sur de nombreuses distributions Linux récentes basées sur un kernel 2.4.x, et pourra être téléchargé à partir du site de Netfilter suivant, <http://www.netfilter.org/>.

## **2. Pare-feu sous Linux.**

### **2.1. Réacheminement IP (IPForwarding).**

Le pare-feu doit être la seule interface entre le réseau local et Internet, et donc il doit avoir au moins deux interfaces réseau (cartes réseau). Pour permettre la transition des paquets entre ces deux réseaux, le réacheminement IP doit être activé. C'est une technique qui consiste à retransmettre les paquets qui entrent sur une interface vers une autre.



**Figure 2.1 : le réacheminement IP entre eth0 et eth1.**

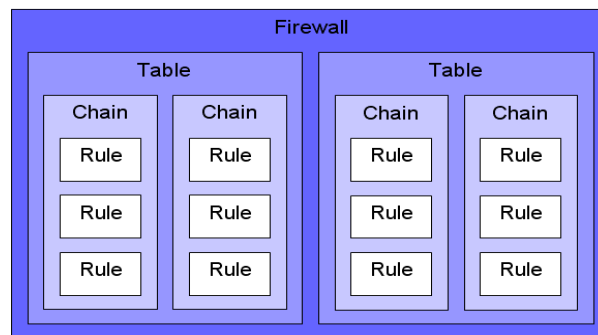
Pour activer le réacheminement IP sur une machine Linux, on pourra utiliser la commande de la figure 2.2.

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

**Figure 2.2 : activer le réacheminement IP.**

## 2.2. L'architecture de Netfilter.

Le Netfilter de la version 2.4 est organisé en 3 tables, chaque table est composée de plusieurs chaînes, et chaque chaîne pourra contenir une ou plusieurs règles.



**Figure 2.3 Architecture d'un Firewall sous Linux.**

Une règle définit l'action et le type de paquets sur lesquels l'action va être appliquée, une chaîne est une liste ordonnée des règles. Une table représente une séparation des règles selon le type d'opération. La figure 2.4 montre un schéma de l'architecture Netfilter.

Netfilter définit trois tables, Filter, NAT et Mangle :

**1. Filter** : c'est la table par défaut, utilisée pour le filtrage des paquets et elle est composée des chaînes suivantes :

- INPUT : cette chaîne s'applique aux paquets destinés à la machine locale.
- OUTPUT : cette chaîne s'applique aux paquets générés par un processus interne et qui sortent de la machine.
- FORWARD : cette chaîne s'applique aux paquets qui transitent via la machine locale, ces paquets sont reçus par une interface réseau et expédiés par une autre.

**2. NAT** : table utilisée pour la translation d'adresse. Les chaînes de cette table sont les suivantes :

- PREROUTING : cette chaîne modifie les paquets reçus via une interface réseau lorsqu'ils arrivent (avant l'opération de routage).
- POSTROUTING : cette chaîne modifie les paquets avant qu'ils ne soient expédiés par une interface de réseau (après l'opération de routage).

**3. Mangle** : table utilisée pour la modification de types spécifiques de paquets. Les chaînes de cette table sont les suivantes :

- INPUT : modifie les paquets réseau destinés à la machine local.
- OUTPUT : modifie les paquets réseau générés de l'interne et destinés à l'extérieur.
- FORWARD : modifie les paquets réseau qui transitent via la machine.
- PREROUTING : cette chaîne modifie les paquets avant le routage.
- POSTROUTING : cette chaîne modifie les paquets après le routage.

Le système vérifie chaque paquet reçu et lorsqu'un paquet correspond à une règle précise, il se voit assigner une action ou cible. Il pourra donc être accepté «ACCEPT», abandonné sans envoyé un message à la machine d'origine «DROP» ou bien abandonné par rejet «REJECT», dans ce dernier cas un message d'erreur est envoyé à la machine d'origine.

### 2.3. La commande iptables.

Iptables est une commande qui sert à ajouter, modifier, supprimer et afficher les règles Netfilter. Elle se lance en ligne de commande par le super-utilisateur « root », et attend de nombreux paramètres et options.

```
iptables [-t <nom-de-table>]
<commande>
<nom-de-chaîne>
<paramètre-1> <option-1>
```

**Figure 2.4 : la commande iptables.**

*<nom-table>* : permet de spécifier une table autre que la table par défaut Filter. *<commande>* : représente une action, telle que l'ajout ou la suppression d'une règle. *<nom-de-chaîne>* : représente la règle sur laquelle l'action sera accomplie. On pourra aussi ajouter des paires de paramètres et options servant à définir l'action à entreprendre lorsqu'un paquet correspond aux critères de la règle. (Voir La figure 2.5).

```
-L      :   liste les chaînes courantes.
-P      :   définit la stratégie par défaut
-A      :   ajouter une règle.
-t      :   spécifier la chaîne.
-p      :   pour spécifier le protocole.
-s      :   spécifier une ou plusieurs adresses sources.
-d      :   spécifier une ou plusieurs adresses destination.
-i      :   spécifier l'interface d'entrée.
-o      :   spécifier l'interface de sortie.
--sport :   spécifier le port source.
--dport :   spécifier le port destination.
--state  :   spécifier l'état du paquet.
          ESTABLISHED   :   pour une connexion déjà établie.
          NEW           :   pour une nouvelle connexion.
          INVALID       :   pour une connexion invalide.
          RELATED       :   pour une nouvelle connexion mais qui est reliée à
                           une connexion déjà établie.
```

**Figure 2.5 les options de la commande iptables.**

## 2.4. Filtrage des paquets.

Netfilter a la possibilité de choisir laisser certains paquets pénétrés dans le système et bloquer les autres. La table Filter est utilisée pour créer les règles de filtrage. Cette table contient les chaînes standards INPUT, OUTPUT et FORWARD.

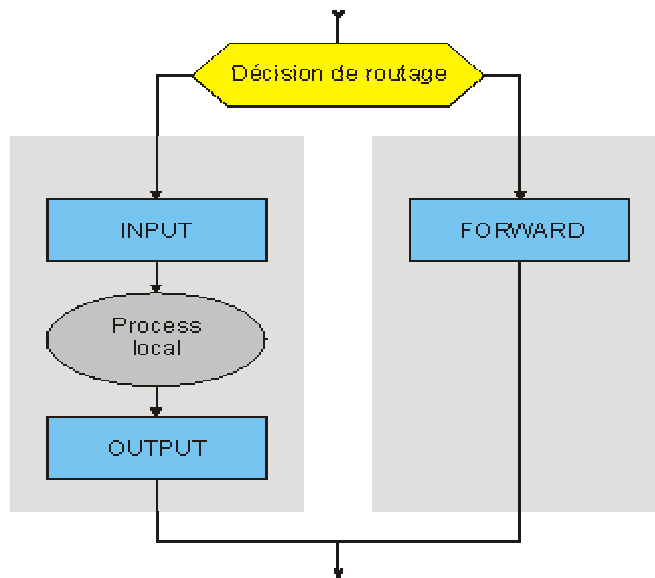


Figure 2.6 : filtrage de paquets.

Tous les paquets qui entrent et sortent sont traités par la table Filter. La figure 2.6 représente la structure interne de cette table. Lorsqu'un paquet entre sur une des interfaces de la machine, une décision de routage est prise, si le paquet est destiné à un processus local, alors il traverse la chaîne INPUT, si le paquet est destiné à un autre réseau alors il traverse la chaîne FORWARD. Un paquet généré par un processus local et qui a comme destination le réseau, traverse la chaîne OUTPUT. La figure 2.7 représente un exemple d'une règle de filtrage.

```
Iptables -A FORWARD -p tcp -d 205.230.20.90 -dport 80 -j ACCEPT
```

Commande      Chaîne      Critères de filtrage      Cible

Figure 2.7 : Exemple d'une règle.

## 2.5. Translation d'adresse.

### 2.5.1. Translation d'adresse source (Masquerading).

Elle consiste à remplacer l'adresse source des paquets qui sortent du réseau local par l'adresse IP publique de la passerelle. Ce type de translation se fait après le routage.

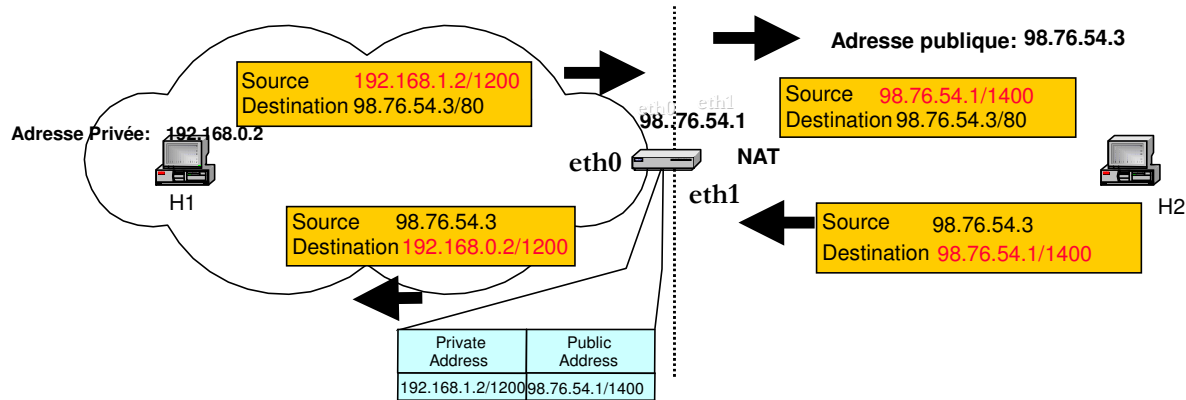


Figure 2.8 translation d'adresse source.

Comme le montre la figure 2.8, la machine passerelle (Adresse publique 98.76.54.1) va effectuer le remplacement de l'adresse IP source de la machine H1 par son adresse publique (98.76.54.1) puis envoyer le paquet vers Internet. Elle va mettre à jour la table de translation d'adresse, d'après cette figure l'adresse privée 192.168.0.2, port 1200 correspond à l'adresse publique : 98.76.54.1, port 1400. Quand la réponse lui parviendra, elle va cette fois-ci modifier l'adresse de destination (98.76.54.1/1400) par celle de H1 (192.168.0.2/1200). Pour activer la translation de l'adresse source, on utilise la commande de la figure 2.9.

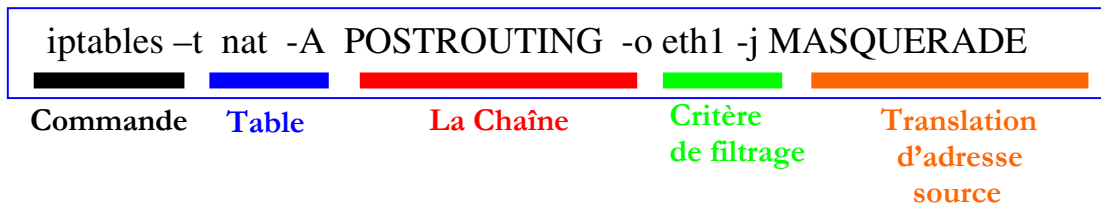
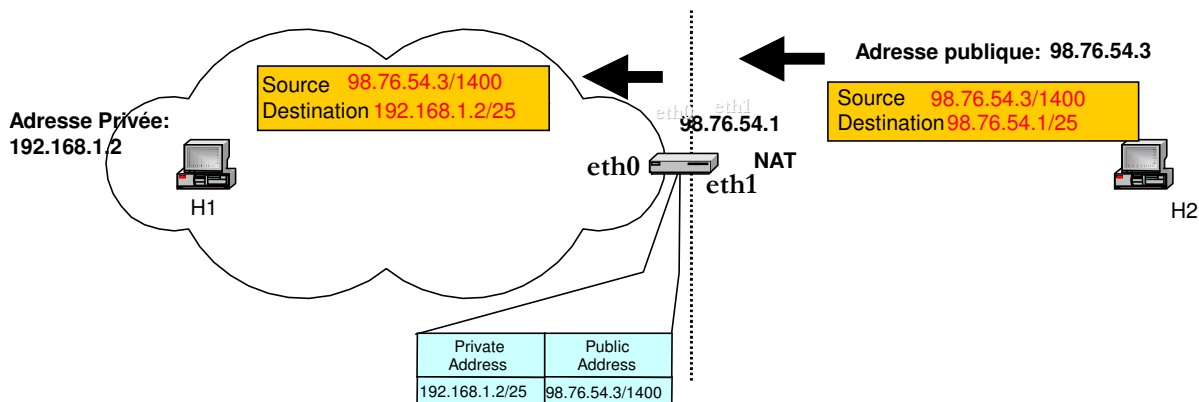


Figure 2.9 : translation d'adresse source (masquerading).

La commande de la figure 2.9 implique la translation d'adresse source à tous les paquets qui sortent via l'interface eth1, qu'est l'interface connecté à Internet.

### 2.5.2. Translation d'adresse de destination (Portforwarding).

Consiste à remplacer l'adresse de destination des paquets qui rentrent dans la passerelle par l'adresse d'une machine qui existe sur le réseau local privé. La translation d'adresse de destination va permettre d'héberger un serveur web ou bien un serveur de messagerie dans le réseau local avec une adresse privée. Lorsque les paquets arrivent à la passerelle de l'Internet, cette dernière va changer l'adresse de destination (qui était son adresse) à l'adresse privée du serveur sur le réseau local (voir figure 2.10).



**Figure 2.10 : translation d'adresse de destination (Portforwarding).**

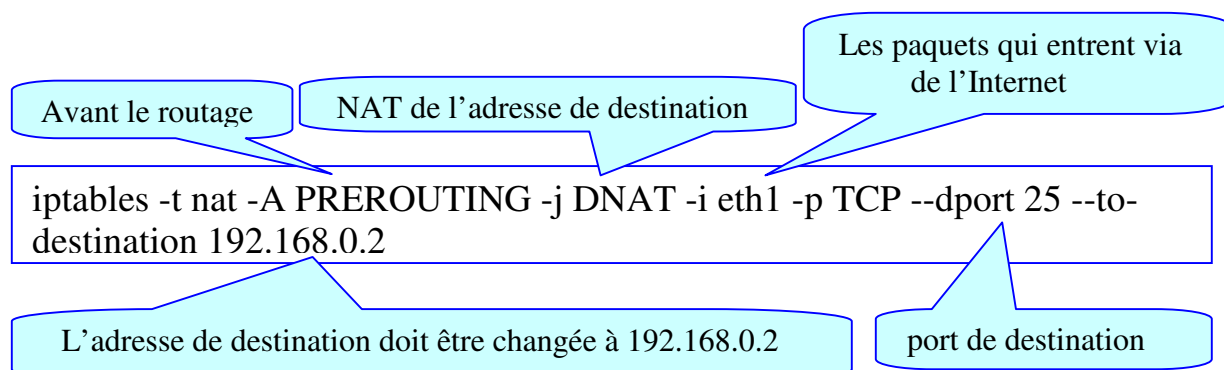
Dans la figure 2.10, H1 est un serveur de messagerie (192.168.0.2). Quand les paquets arrivent avec un numéro de port 25, la passerelle change l'adresse de destination de ces paquets à l'adresse du serveur de messagerie (192.168.0.2). Pour activer la translation d'adresse de destination il faut premièrement permettre le transit (Forwarding) des paquets qui arrivent sur le port 25 en utilisant la commande de la figure 2.11.

```
iptables -A FORWARD -i eth1 -p tcp --dport 25 -j ACCEPT
```

Accepter le transit (Forward) des paquets qui entrent via l'interface eth1 (Internet) et qu'ont comme port de destination 25.

**Figure 2.11 : activation du transit des paquets vers un serveur de messagerie.**

Deuxièmement, on active la translation d'adresse de destination en utilisant la commande de la figure 2.12.



**Figure 2.12 : activation de la translation d'adresse de destination.**

## BIBLIOGRAPHIE.

- 1- Andrew Lockhart, Network Security Hacks, April 2004
- 2- [www.lea-linux.org](http://www.lea-linux.org)