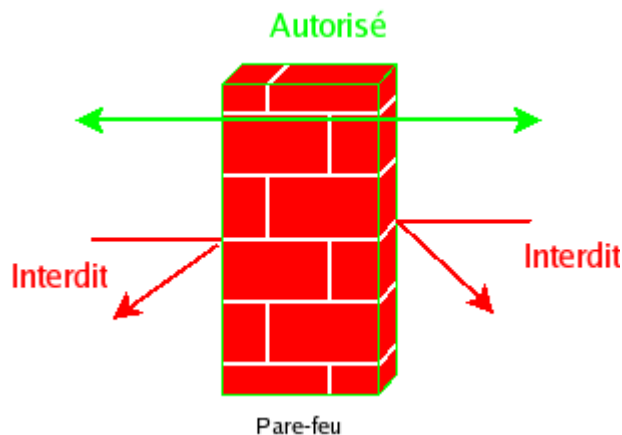


Sommaire

- I.Qu'est-ce qu'un pare-feu ?.....1
- II.Architecture d'iptables.....2
- III.Les différents types de filtrages : les tables.....2
- IV.Fonctionnement de base : les chaînes et les règles.....2
- V.Les extensions.....3
- VI.La commande iptables.....3
 - a)Les commandes d'iptables.....3
 - b)Options courantes pour les règles.....4
 - c)Sélection des paquets par le protocole pour une règle.....4
 - 1Options spécifiques aux protocoles TCP et UDP4
 - 2Options spécifiques au protocole ICMP.....5
 - d)Sélection des paquets par leur état (extension state) pour une règle.....5
 - e)Traitement des paquets sélectionnés par une règle : les cibles.....6
- VII.Sauvegarde des règles du pare-feu.....6
 - a)Première méthode (pour le backup).....6
 - b)Seconde méthode (la meilleure).....7
- VIII.Fichier de configuration du pare-feu.....7
- IX.Log des paquets rejetés7
 - a)Première méthode (un peu encombrante).....8
 - b)Seconde méthode (de loin la meilleure).....8
- X.Bibliographie.....9

I. Qu'est-ce qu'un pare-feu ?

Un pare-feu est un filtre qui protège un système en bloquant les connexions venant de l'extérieur (entrées) ou de l'intérieur (sorties) pour empêcher ou autoriser l'accès à des services Web. Il permet aussi de faire de la translation d'adresse pour servir de routeur.



Iptables (NetFilter) est le pare-feu de Linux depuis la version 2.4 et 2.6 du noyau.

II. Architecture d'iptables

La commande Linux pour gérer le pare-feu est iptables.

L'architecture d'iptables est la suivantes :

- TABLE : utilisé pour classer le type de filtrage (IO, routage, autre)
 - CHAÎNE : pour affiner le type de filtrage (entrées, sorties, passage, ...)
 - REGLE : pour filtrer un type de paquet (port, ip...)
 - POLICE : pour filtrer les paquets qui n'ont pas de règle particulière
 - CIBLE : pour indiquer quoi faire du paquet que correspond à la règle (accepter, refuser, rejeter, modifier...)
 - EXTENSION : pour filtrer très très finement les types de paquets

III. 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

IV. 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 sûre).

V. Les extensions

Les extensions sont des composants enfichables et enfichés plus ou moins statiquement et qui permettent deux choses :

- ajouter des critères de raffinement de correspondance de paquets à une règle : par exemple, par son état (de la connexion dont il provient)
- ajouter des cibles donc des traitements pour les paquets correspondants à une règle

VI. La commande iptables

La commande iptables permet, entre autre, d'ajouter des règles pour les filtrer un type de paquet. Le filtrage se fait par raffinement successif des critères de sélection : IP, protocole, état de la connexion...

a) Les commandes d'iptables

Si la chaîne *chaîne* n'est pas dans la table par défaut (filter), on fera précéder les commandes par *-t table*.

Une règle *règle* se construit comme suit : *sélection_paquet -j cible*

Pour plus d'information sur *sélection_paquet* et *cible* voir les rubriques suivantes.

Les commandes essentiels d'iptables sont (et utilisation classique) :

- **-A** ou **--append** : Ajout d'une règle à la fin de la chaîne
[root]# iptables -A chaîne règle
- **-D** ou **--delete** : Suppression d'une règle par son numéro de 1 à *n*
[root]# iptables -D chaîne numéro
- **-R** ou **--replace** : Remplacement d'une règle avec son numéro de 1 à *n*
[root]# iptables -R chaîne numéro_règle_replacée règle
- **-I** ou **--insert** : Insertion d'une règle à l'emplacement numéro_place (1 à *n*)
[root]# iptables -I chaîne numéro_place règle
- **-L** ou **--list** : Lister les règles. Pour avoir les interfaces, ajouter l'option **-v**.
[root]# iptables -L
- **-F** ou **--flush** : Vider une chaîne : supprimer toutes les règles de la chaîne
[root]# iptables -F chaîne
- **-Z** ou **--zero** : Mettre à zéro une chaîne : mettre à zéro les compteurs de paquets d'une chaîne
[root]# iptables -F chaîne
- **-N** ou **--new-chain** : Créer une nouvelle chaîne personnalisée. Cela peut servir à classer les règles suivant le protocole ou le type (serveur, passerelle, limiteur, icmp, tcp...)
[root]# iptables -N chaîne_perso
- **-X** ou **--delete-chain** Effacer une chaîne créée avec **-N**
[root]# iptables -X chaîne_perso
- **-P** : **--policy** : Permet de spécifier une politique par défaut pour les paquets qui ne correspondent à aucune règle
[root]# iptables -P chaîne -j cible

Au début de chaque script de pare-feu, il faut vider (-F) et remettre les compteurs à zéro (-Z) de

toutes les chaînes que l'on utilise. Ceci est nécessaire pour ne pas garder la configuration antérieure.

b) Options courantes pour les règles

Options courtes	Options longues	Description
-t <i>table</i>	--table	Indique la table (filter, nat ou mangle) dans laquelle se trouve la chaîne à configurer (par défaut : filter)
-s [!] <i>IP[/masque]</i>	--source	Indique que la règle est valable uniquement pour les paquets ayant (ou ! n'ayant pas) <i>IP</i> (et éventuellement <i>masque</i>) comme adresse IP (et masque) d'émetteur.
-d [!] <i>IP[/masque]</i>	--destination	Indique que la règle est valable uniquement pour les paquets ayant (ou ! n'ayant pas) <i>IP</i> (et éventuellement <i>masque</i>) comme adresse IP (et masque) de destinataire.
-j <i>cible</i>	--jump	Indique la cible <i>cible</i> comme action pour les paquets correspondants à cette règle. Il peut s'agir de ACCEPT, DROP, REJECT ou LOG.
-i [!] <i>interface</i>	--in-interface	Indique que la règle est valable uniquement pour les paquets arrivant (ou ! n'arrivant pas) par l'interface <i>interface</i> (lo, eth0, eth0:0, eth1...). (valide uniquement pour INPUT, FORWARD, PREROUTING)
-o [!] <i>interface</i>	--out-interface	Indique que la règle est valable uniquement pour les paquets devant sortir (ou ! ne devant pas sortir) par l'interface <i>interface</i> (lo, eth0, eth0:0, eth1...). (valide uniquement pour FORWARD, OUTPUT, POSTROUTING)
-p [!] <i>protocole</i>	--protocol	Indique que la règle est valable uniquement pour les paquets ayant (ou pas) pour protocole, le protocole <i>protocol</i> . (tcp, udp ou icmp)

c) Sélection des paquets par le protocole pour une règle

Le choix du protocole de la règle se fait avec l'option -p. On utilisera -p tcp pour TCP, -p udp pour UDP et -p icmp pour ICMP.

1 Options spécifiques aux protocoles TCP et UDP

Si l'on précise l'une des options suivantes -p tcp ou -p udp, on peut la faire suivre par les options suivantes :

Options courtes	Options longues	Description
-sport [!] [port[:portfin]]	--source-port	Indique que la règle est valable uniquement pour les paquets ayant (ou ! n'ayant pas) un port d'émission <i>port</i> ou un port d'émission compris entre <i>port</i> et <i>portfin</i> .
-dport [!] [port[:portfin]]	--destination-port	Indique que la règle est valable uniquement pour les paquets ayant (ou ! n'ayant pas) un port de destination <i>port</i> ou un port de destination compris entre <i>port</i> et <i>portfin</i> .

2 Options spécifiques au protocole ICMP

Si l'on précise l'une des options suivantes -p icmp, on peut la faire suivre par les options suivantes :

Options	Description
--icmp-type [!] <i>nom_du_type_icmp</i>	Indique que la règle est valable uniquement pour les paquets ayant (ou ! n'ayant pas) un type icmp <i>nom_du_type_icmp</i> .

Les types ICMP sont :

- any : tous les messages ICMP
- echo-reply : réponse d'un ping
- destination-unreachable : impossible de trouver un chemin vers le serveur
- network-unreachable : impossible de trouver un chemin vers le réseau du serveur
- host-unreachable : impossible de trouver un chemin vers le serveur
- port-unreachable :: impossible de se connecter au port voulu
- echo-request : demande de ping
- time-exceeded (ttl-exceeded) : la durée de vie de la trame est atteinte
- icmp-net-prohibited : le réseau source est exclu
- icmp-host-prohibited : la machine source est exclue

d) Sélection des paquets par leur état (extension state) pour une règle

Le module state permet de sélectionner les paquets par leur état pour les connexions TCP (à l'aide des drapeaux SYN et ACK entre autre). Pour l'activer pour une règle, il faut utiliser l'option -m state (valable uniquement pour TCP : protocole à pseudo-connexion)

Option	Description
--state <i>états</i>	Indique que la règle est valable uniquement pour les paquets ayant l'un des états <i>états</i> (liste séparée par une virgule). Les états sont les suivants : NEW (nouvelle connexion), ESTABLISHED (connexion établie), RELATED (nouvelle connexion en rapport avec une autre déjà établie) ou INVALID (aucun des autres cas).

Remarque : dans le cas du protocole TCP, on peut aussi utiliser le flags `-syn` à la place de `-m state --state NEW` pour détecter les nouvelles connexions.

e) Traitement des paquets sélectionnés par une règle : les cibles

Une fois les paquets sélectionnés, il faut indiquer à iptables quoi en faire.

Cible	Option	Description
-j LOG		Cette chaine permet de logger les paquets. Si elle se trouve en fin de chaîne et la police de la chaîne à DROP, elle loggue tous les paquets qui n'ont pas correspondu à une règle de la chaîne (les paquets rejetés). Les règles ayant cette cible passe le paquet à la règle suivante (contrairement aux autres cibles). Par défaut les logs sont stockés dans kernel (on ne peut pas le changer) au niveau warning (on peut le changer). On lit les logs iptables avec <code>dmesg</code> (ou autres).
	<code>--log-level <i>niveau</i></code>	Indique le niveau auquel se font les logs dans la source kernel (par défaut warning). Peut être utile de le changer pour filtrer les logs iptables. <i>Niveau</i> est à choisir parmi : <i>debug, info, notice, warning, err, crit, alert, emerg</i>
	<code>--log-prefix <i>prefixe</i></code>	Indique le préfixe utilisé pour les inscriptions dans les logs syslog. Utile pour distinguer les messages iptables des messages du noyau. (maximum 29 caractères)
-j REJECT		Identique de DROP : supprime le paquet mais renvoie un paquet ICMP d'erreur.
	<code>--reject-with <i>type</i></code>	Indique le type d'erreur renvoyé en cas de rejet. <i>Type</i> est à choisir parmi : <i>icmp-net-unreachable, icmp-host-reachable, icmp-port-unreachable, icmp-proto-unreachable, icmp-net-prohibited</i> ou <i>icmp-host-prohibited</i> .
-j ACCEPT		Indique que le paquet est autorisé à être transmis à sa destination
-j DROP		Indique que le paquet est supprimé sans notifier l'émetteur

VII. Sauvegarde des règles du pare-feu

Maintenant il faudrait pouvoir sauvegarder tout ça !!!

a) Première méthode (pour le backup)

L'enregistrement des règles se fait avec la commande `iptables-save` et restaurer les règles enregistrées avec `iptables-restore`. Ces deux programmes affichent les règles sur stdout. Il faut donc rediriger le tout vers un fichier. Ensuite il faut se faire un script de démarrage qui charge les règles au boot de la machine.

b) Seconde méthode (la meilleure)

Toutefois il y a plus simple. Il existe un fichier de configuration `/etc/sysconfig/iptables` dans lequel on peut sauvegarder la configuration du firewall. Le format est le même que celui utilisé par `iptables-save`.

Pour faire cela on utilise la commande :

```
[root@server ~]# service iptables save
```

ou encore

```
[root@server ~]# /etc/init.d/iptables save
```

Maintenant nos règles seront actives au démarrage.

VIII. Fichier de configuration du pare-feu

Il existe également des options pour le service `iptables` qui se trouvent dans le fichier `/etc/sysconfig/iptables-config`.

Les options sont les suivantes :

Option	Description
<code>IPTABLES_MODULES="liste_module s_sep_espace"</code>	Charge des modules <code>iptables</code> additionnels (principalement pour le NAT (ftp, irc...)) Ces modules sont inscrits dans <code>/etc/modprobe.conf</code> .
<code>IPTABLES_MODULES_UNLOAD="yes no"</code>	Indique s'il faut décharger les modules <code>iptables</code> à l'arrêt du service. Devrait toujours être <code>yes</code> .
<code>IPTABLES_SAVE_ON_STOP="yes no"</code>	Indique s'il faut enregistrer les règles du pare-feu à son arrêt.
<code>IPTABLES_SAVE_ON_RESTART="yes no"</code>	Indique s'il faut enregistrer les règles du pare-feu à son redémarrage.
<code>IPTABLES_SAVE_COUNTER="yes no"</code>	Indique s'il faut enregistrer les compteurs du pare-feu.
<code>IPTABLES_STATUS_NUMERIC="yes no"</code>	Indique si l'on veut un affichage numérique des IPS et ports pour un <code>iptables -L</code>

IX. Log des paquets rejetés

Il peut être intéressant aussi de loguer les tentatives qui sont refusées.
Pour cela, il y a deux solutions.

a) Première méthode (un peu encombrante)

On peut utiliser la cible LOG

```
iptables -A chaine -j LOG --log-level info
```

Dans /etc/syslog.conf :

```
kern.=info -/var/log/iptables
```

Les options que l'on peut ajouter après le -j LOG, sont les suivantes :

Options	Descriptions
--log-level <i>niveau</i>	Indique le niveau auquel les logs sont faits dans syslog dans la catégorie noyau. Par défaut, il s'agit du niveau warn.
--log-prefix <i>préfixe</i>	Indique un préfixe à ajouter à chaque ligne de log pour faciliter le parsing du fichier, par exemple, avec grep. Maximum 29 caractères

ATTENTION : la cible LOG met toujours ses logs dans syslog dans la catégorie KERN (noyau). En outre, par défaut, elle log au niveau warn. Ceci a pour effet de polluer très massivement les logs et les consoles.

ATTENTION : même avec la configuration précédente de syslog, il se peut que le noyau émette des logs au niveau info et donc dans les logs de syslog. Cette méthode est donc à utiliser UNIQUEMENT POUR DES TESTS.

b) Seconde méthode (de loin la meilleure)

On peut aussi utiliser ULOG :

- Installer ULOG (yum install ulogd)
- démarrer ulogd (service ulogd start et chkconfig ulogd on)
- configurer ulogd :

```
[LOGEMU]
```

```
#indique le fichier dans lequel on met les logs d'iptables
```

```
file="/var/log/ulogd/ulogd.syslogemu"
```

```
sync=1
```

- remplacer `iptables -A chaine -j LOG -log-level info` par `iptables -A chaine -j ULOG`

Note : ULOG est à réserver pour une journalisation intensive et continue et pas pour des tests. Il permet par exemple de stocker les logs dans une base MySQL.

X. Bibliographie

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

[freshmeat.net: Project details for **ulogd**](#)