Centos: NAT dans Iptables

Table of Contents

Introduction

Présentation d'Iptables

Iptables est un programme qui permet de configurer les tables fournies par le pare-feu du noyau Linux (implémenté en tant que différents modules Netfilter) et les chaînes et règles qu'il stocke. En d'autres termes, iptables est un outil utilisé pour gérer les règles du pare-feu Linux.

Cas d'utilisation

- **Configuration de type routeur/Pour mutualiser l'accés aux adresses publiques**: La plupart des organisations se voient attribuer un nombre limité d'adresses IP routables publiquement par leur FAI. En raison de cette allocation limitée, les administrateurs doivent trouver des moyens de partager l'accès aux services Internet sans donner d'adresses IP publiques à chaque nœud sur le réseau local. L'utilisation d'une adresse IP privée est le moyen courant permettant à tous les nœuds d'un réseau local d'accéder correctement aux services réseau internes et externes. Les routeurs Edge (tels que les pare-feu) peuvent recevoir des transmissions entrantes à partir d'Internet et acheminer les paquets vers le nœud LAN prévu. Les pare-feu / passerelles peuvent également acheminer les demandes sortantes d'un nœud de réseau local vers le service Internet distant.
- **Configuration de type pare feu/Pour restreindre et/ou contrôler l'accès aux serveurs**: Le pare-feu du serveur passerelle peut être utilisé pour restreindre le trafic sortant du sous-réseau.
- **Configuration de type edgerouter**: Un routeur périphérique (edgerouter) est un routeur spécialisé résidant au bord ou à la limite d'un réseau. Ce routeur assure la connectivité de son réseau avec des réseaux externes, un réseau étendu ou Internet. Un routeur de périphérie utilise un protocole de passerelle de frontière externe, qui est largement utilisé sur Internet pour fournir une connectivité avec des réseaux distants. Au lieu de fournir une communication avec un réseau interne, que le routeur central gère déjà, un routeur périphérique peut fournir une communication avec différents réseaux et systèmes autonomes. Dans certains cas, une organisation gère plusieurs réseaux isolés et utilise des routeurs de périphérie pour les relier entre eux au lieu d'utiliser un routeur principal.Il peut aussi arriver de devoir héberger un sous-réseau isolé en dehors du sous-réseau privé ou de l'Internet public, par exemple en cas de secours. Un serveur de passerelle Linux peut être utilisé pour relier deux réseaux ensemble. Une carte réseau se connecte à un réseau externe ou public tandis que l'autre carte réseau se connecte au sous-réseau privé. Le transfert IP et une règle NAT sont ensuite utilisés pour acheminer le trafic du sous-réseau privé vers le réseau externe. Le trafic provenant des serveurs internes semble provenir de l'adresse IP de la passerelle. Le trafic généré de l'extérieur atteindra la passerelle et n'aura aucune visibilité sur le sous-réseau privé.

Description de Iptables

Les tables

Table NAT (Network Address Translation)

Table utilisée pour la translation d'adresse ou la translation de port.

Il a 2 types de chaînes:

- **PREROUTING** qui permet de spécifier « à l'arrivée du pare-feu »
- **POSTROUTING** qui permet de spécifier « à la sortie du pare-feu ».

Table FILTER

C'est la table par défaut lorsque l'on en spécifie pas.

Cette table contient toutes les règles de filtrage, il existe 3 types de chaînes :

- **FORWARD** pour les paquets passant par le pare-feu,
- **INPUT** pour les paquets entrant
- **OUTPUT** pour les paquets sortants.

Table Mangle

C'est la table qui contient les règles pour la modification de paquets.

Elle est peu utilisée et ne sera pas décrite dans cet article.

Les règles sont évaluées dans l'ordre, par défaut la table FILTER est vide et donc accepte tout. Aucune règle de translation d'adresse n'est présente par défaut.

Les Chaînes

une chaîne est une suite de règles, qui sont prises dans l'ordre ; dès qu'une règle s'applique à un paquet, elle est déclenchée, et la suite de la chaîne est ignorée.

Il existe 3 targets (ou cibles) :

1. **SNAT**: Permet de modifier l'adresse source du paquet.
2. **DNAT**: Permet de modifier l'adresse destination du paquet.
3. **MASQUERADE**: Transforme les paquets sortants passant pour donner l'illusion qu'ils sortent de la passerelle elle-même par un port alloué dynamiquement ; lorsque la passerelle recoit une réponse sur ce port, elle utilise une table de correspondance entre le port et les machines du réseau privé qu'elle gère pour lui faire suivre le paquet.

Les policy disponibles sont :

1. **policy ACCEPT**: Permet d'accepter un paquet grâce à la règle vérifiée.
2. **policy DENY**:
3. **policy DROP** : Rejet d'un paquet sans message d'erreur si la règle est vérifiée (“non ! j'en veux pas mais je dis rien à l'expediteur”).
4. **policy REJECT**: Rejet avec un retour de paquet d'erreur à l'expediteur si la règle est verifiée (“un paquet recommandé de La Poste refusé par son destinataire”).
5. **policy LOG**: Affiche le résultat vers la sortie standard. Peut être associé a un préfixe pour une meilleure lisibilité (un préfixe pour les paquets rejetes,un autre pour les paquets forwardés etc...).

```
-j LOG --log-prefix '[IPTABLES DROP] : '
```

Les commandes

iptables n'est pas livré avec une interface graphique par défaut ; les commandes et les règles sont généralement passées en ligne de commande. Le mieux est d'écrire des scripts (à rajouter dans /etc/rc.d/init.d) qui permettent d'appliquer toutes les règles d'un seul coup, dès le démarrage de GNU/Linux.

Commandes principales

-A –append	Ajoute la règle à la fin de la chaîne spécifiée	`iptables -A INPUT ...`
-D –delete	Permet de supprimer une chaîne. On peut l'utiliser de 2 manières, soit en spécifiant le numéro de la chaîne a supprimer, soit en spécifiant la règle à retirer.	`iptables -D INPUT --dport 80 -j DROP` `iptables -D INPUT 1`
-R –replace	Permet de remplacer la chaîne spécifiée.	`iptables -R INPUT 1 -s 192.168.0.1 -j DROP`
-I –insert	Permet d'ajouter une chaîne dans un endroit spécifié de la chaîne.	`iptables -I INPUT 1 --dport 80 -j ACCEPT` [1)]
-L –list	Permet d'afficher les règles.	`iptables -L # Affiche toutes les règles des chaînes de FILTER` `iptables -L INPUT # Affiche toutes les règles de INPUT (FILTER)`
-F –flush	Permet de vider toutes les règles d'une chaîne.	`iptables -F INPUT`
-N –new-chain	Permet de créer une nouvelle chaîne.	`iptables -N LOG_DROP`
-X –delete-chain	Permet d'effacer une chaîne.	`iptables -X LOG_DROP`
-P –policy	Permet de spécifier au noyau la politique par défaut d'une chaîne DENY, ACCEPT, REJECT, DROP ...	`iptables -P INPUT DROP`

Commandes pour matcher

Le “!” peut être utilisé pour certaines commandes afin de spécifier le contraire (on peut le traduire par “sauf”). Par exemple une commande qui doit refuser tout trafic TCP sauf ce qui provient de l'adresse IP 10.42.42.42 sera traduite par la commande suivante :

```
iptables -A INPUT -p tcp --source ! 10.42.42.42 -j DROP
```

Les adresses IP peuvent optionnellement être spécifiées avec le masque associé sous la forme [adresse ip]/[masque].

-j (jump)	Défini l'action à prendre si un paquet répond aux critères de cette règle: ACCEPT, LOG, DROP...	`iptables -A INPUT -p icmp -j DROP`
-p –protocol	Spécifie un protocole : tcp, udp, icmp, all (tous)	`iptables -A INPUT -p icmp -j DROP`
-s –source	Spécifie une adresse source à matcher	`iptables -A INPUT -p tcp -s 192.168.42.42 -j ACCEPT`
-d –destination	Spécifie une adresse destination	`iptables -A FORWARD -p tcp -d 10.1.0.1 -j ACCEPT`
-i –in-interface	Spécifie une interface d'entrée	`iptables -A INPUT -p icmp -i eth0 -j DROP`
-o –out-interface	Spécifie une interface de sortie	`iptables -A OUTPUT -p icmp -o eth0 -j DROP`
-f –fragment	Paquet fragmenté	`iptables -A INPUT -p icmp -f -j DROP`

-sport –source-port	Spécifie le port source ou une plage de ports, fonctionne aussi en udp, -m multiport permet de spécifier plusieurs ports à matcher.	`iptables -A INPUT -p tcp --sport 80 -j ACCEPT`
		`iptables -A INPUT -p udp --sport 80 -j DROP`
		`iptables -A OUTPUT -p tcp -m multiport --sport 3128,21,1000 -j DROP`
		`iptables -A OUTPUT -p tcp --sport 1024:2042 -j ACCEPT`

-dport –destination-port	Spécifie le port destination ou une plage de ports, fonctionne aussi en udp, -m multiport permet de spécifier plusieurs ports a matcher.	`iptables -A INPUT -p tcp --dport 110 -j DROP`
		`iptables -A INPUT -p udp --dport 110 -j DROP`
		`iptables -A INPUT -p tcp -m multiport --dport 110,4242,119 -j DROP`
		`iptables -A INPUT -p tcp --sport 4925:4633 -j ACCEPT`

-tcp-flags	Spécifier un flag tcp à matcher : SYN ACK FIN RST URG PSH ALL NONE	`iptables -A INPUT -p tcp --dport 42 --tcp-flags SYN,ACK -j ACCEPT`
-icmp-type	Spécifie un type de paquet icmp à matcher	`iptables -A INPUT -p icmp --icmp-type 8 -j DROP`

-mac-source	Spécifier l'adresse MAC à matcher	`iptables -A INPUT --mac-source 42.42.AA.42.42.AA -j DROP`
		n cas d'erreur, il peut-être nécessaire de spécifier le module mac :
		`iptables -A INPUT -m mac --mac-source 42.42.AA.42.42.AA -j DROP`

-state	Permet de spécifier l'état du paquet à matcher parmi les états suivants : * **ESTABLISHED**: paquet associé à une connexion déjà établie * **NEW**: paquet demandant une nouvelle connexion * **INVALID** : paquet associé à une connexion inconnue * **RELATED** : Nouvelle connexion mais liée, idéal pour les connexions FTP	`iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT` `iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT`

Spécifiques pour le NAT

-to-destination	Utilisé en target pour le DNAT, permet de spécifier l'adresse de destination de la translation, on peut également spécifier un port s'il est différent du port source.	`iptables -t nat -A PREROUTING -d 42.12.42.12 -p tcp --dport 110 -j DNAT --to-destination 192.168.1.2:6110` `iptables -t nat -A PREROUTING -d ! 42.12.42.12 -p tcp --dport 80 -j DNAT --to-destination 192.168.2.1:3128`
-to-source	Utilisé pour en target pour le SNAT, permet de spécifier l'adresse source de la translation.	

Spécifiques pour les LOGS

-log-level	Level, niveau de log
-log-prefix	Permet de spécifier un préfixe pour les logs.

Exemples de configuration

Pré requis

Activation du forward IP

Par défaut, la stratégie IPv4 dans les noyaux Red Hat Enterprise Linux désactive la prise en charge du transfert IP, ce qui empêche les serveurs exécutant Red Hat Enterprise Linux de fonctionner en tant que routeurs dédiés. Pour activer le transfert IP, exécuter la commande suivante:

```
$ sysctl -w net.ipv4.ip_forward = 1
```

Si cette commande est exécutée via l'invite du shell, le paramètre n'est pas mémorisé après un redémarrage. On peut définir définitivement le transfert en remplaçant 0 par 1 sur la ligne net.ipv4.ip_forward dans le fichier /etc/sysctl.conf:

```
$ sed -i 's/net.ipv4.ip_forward = 0/net.ipv4.ip_forward = 1/'
/etc/sysctl.conf
```

Exécuter la commande suivante pour activer la modification dans le fichier sysctl.conf:

```
$ sysctl -p /etc/sysctl.conf
```

Pour partager une connexion, il faut que le forwarding soit activé dans le noyau, puis il faut autoriser iptable à faire le forwarding :

```
$ iptables -F FORWARD
$ iptables -A FORWARD -j ACCEPT
```

Activation des modules du noyau

Vérifier les noms des modules avec la commande suivante

```
$ ls -l /lib/modules/$(uname -r)/kernel/net/ipv4/netfilter/
```

- Dans le noyau 2.6.19 ou antérieur, les noms des modules sont ip_
- Dans les noyaux postérierurs, les noms des modules sont nf_

Vérifier la présence des modules nécessaires (comme nf_nat_ftp, nf_conntrack_ftp) avec la commande lsmod pour afficher le contenu du fichier /proc/modules.

```
$ lsmod | grep nf
```

```
  nf_nat_ftp              3443  0
  nf_conntrack_ftp       11953  1 nf_nat_ftp
  nf_nat                 22676  3 nf_nat_ftp,ipt_MASQUERADE,iptable_nat
  nf_conntrack_ipv4       9154  4 iptable_nat,nf_nat
  nf_defrag_ipv4          1483  1 nf_conntrack_ipv4
  nf_conntrack           79206  7
nf_nat_ftp,nf_conntrack_ftp,ipt_MASQUERADE,iptable_nat,nf_nat,nf_conntrack_i
pv4,xt_state
```

Si les modules ne sont pas chargés utiliser la commande modprobe pour les activer

CAS 1 : Configuration de type routeur

Descriprion de la configuration

Le serveur dispose de 2 cartes réseau configurées comme ci-dessous:

- **eth0**: externe du réseau de perspective d'eth1, configuré avec une adresse IP statique 10.10.1.20,
- **eth1**: réseau interne (LAN), configuré avec une adresse IP statique 10.8.8.2.

Le réseau 10.10.1.0/24 est ensuite naturellement NATé via le routeur de la maison pour avoir accès à Internet.

On souhaite obtenir la configuration suivante

- Les serveurs sur le réseau privé 10.8.8.0/24 doivent pouvoir envoyer une requête ping aux adresses IP externes,
- Les serveurs sur le réseau privé 10.8.8.0/24 doivent pouvoir accéder aux serveurs Web publics via HTTP / HTTPS,
- Tout autre trafic sortant du réseau privé 10.8.8.0/24 vers le public devrait être refusé,
- Les serveurs sur le réseau public 10.10.1.0/24 doivent pouvoir accéder aux serveurs Web sur le réseau privé 10.8.8.0/24 via HTTP / HTTPS,
- Les serveurs du réseau public 10.10.1.0/24 doivent pouvoir se connecter au réseau privé via SSH,
- Tout autre trafic entrant provenant du réseau public 10.10.1.0/24 vers le réseau privé 10.8.8.0/24 doit être enregistré et refusé.

La table de routage ressemble à ceci:

```
$ ip ro
```

```
  10.10.1.0/24 dev eth0  proto kernel  scope link  src 10.10.1.20
  10.8.8.0/24 dev eth1  proto kernel  scope link  src 10.8.8.2
  default via 10.10.1.1 dev eth0
```

La passerelle par défaut est sur un interface eth0 “publique”.

Configuration de la chaîne Forward (table de filtre)

Pour que les machines du réseau interne (eth1) puissent envoyer une requête ping aux serveurs publics, il faut autoriser tout nouveau trafic ICMP, provenant de eth1, à quitter via eth0:

```
$ iptables -A FORWARD -i eth1 -o eth0 -p icmp -m state --state NEW -j ACCEPT
```

Pour que les serveurs du réseau interne (eth1) puissent établir des connexions aux serveurs Web publics sur les ports HTTP et HTTPS:

```
$ iptables -A FORWARD -s 10.8.8.0/24 -i eth1 -o eth0 -p tcp -m multiport --
dports 80,443 -m state --state NEW -j ACCEPT
```

La définition de l'adresse réseau source n'est pas obligatoire ici.

Pour pouvoir contacter les serveurs web privés depuis le réseau public (eth0) , :

```
$ iptables -A FORWARD -d 10.8.8.0/24 -i eth0 -o eth1 -p tcp -m multiport --
dports 80,443 -m state --state NEW -j ACCEPT
```

Pour pouvoir se connecter au réseau privé (eth1) via SSH:

```
$ iptables -A FORWARD -d 10.8.8.0/24 -i eth0 -o eth1 -p tcp -m multiport --
dports 22 -m state --state NEW -j ACCEPT
```

Pour autoriser toutes les connexions déjà établies ou connexes, dans les deux sens:

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

Enfin, pour logger tout le reste à des fins de dépannage avant de le jetter:

```
$ iptables -A FORWARD -p tcp -j LOG --log-prefix "iptables_forward "
$ iptables -A FORWARD -p tcp -j DROP
```

Voici la configuration finale d'iptables pour la chaîne de transfert de table de filtres:

```
$ iptables -S FORWARD
-P FORWARD ACCEPT
-A FORWARD -i eth1 -o eth0 -p icmp -m state --state NEW -j ACCEPT
-A FORWARD -s 10.8.8.0/24 -i eth1 -o eth0 -p tcp -m multiport --dports
80,443 -m state --state NEW -j ACCEPT
-A FORWARD -d 10.8.8.0/24 -i eth0 -o eth1 -p tcp -m multiport --dports
80,443 -m state --state NEW -j ACCEPT
-A FORWARD -d 10.8.8.0/24 -i eth0 -o eth1 -p tcp -m multiport --dports 22 -m
state --state NEW -j ACCEPT
-A FORWARD -p tcp -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -p tcp -j LOG --log-prefix "iptables_forward "
-A FORWARD -p tcp -j DROP
```

Configuration de le Chaîne Prerouting

La chaîne de pré-acheminement (Prerouting) aide à traduire l'adresse IP de destination des paquets en quelque chose qui correspond au routage sur le serveur local. Ceci est utilisé pour le NAT de destination (DNAT).

Dans l'exemple suivant, on souhaite transférer tout le trafic SSH sur le port local 22 du port 22043 à 10.8.8.43:

```
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22043 -m comment --
```

```
comment "DNAT SSH for .43" -j DNAT --to-destination 10.8.8.43:22
```

Lorsqu'on initialise une connexion SSH depuis le réseau extérieur (eth0) vers 10.10.1.20:22043, on accédera à la machine 10.8.8.43 sur le port 22.

Plus de règles DNAT, similaires à celles ci-dessus, peuvent être ajoutées, par exemple, pour atteindre un serveur web HTTP sur le port 8080 qui est hébergé sur 10.8.8.44:80:

```
$ ptables -t nat -A PREROUTING -i eth0 -p tcp --dport 8080 -j DNAT --to-
destination 10.8.8.44:80
```

Configuration de chaîne Postrouting

Chaîne de post-routage (Postrouting) aide à traduire l'adresse IP source des paquets à quelque chose qui pourrait correspondre au routage sur le serveur de destination. Ceci est utilisé pour le NAT source (SNAT).

Pour permettre aux nœuds privés (eth1) avec des adresses IP privées de communiquer avec des réseaux publics externes, configurer le pare-feu pour le masquage IP, qui masque les requêtes des nœuds LAN avec l'adresse IP du périphérique externe du pare-feu (dans ce cas , eth0):

```
$ iptables -t nat -A POSTROUTING -o eth0 -p icmp -j MASQUERADE
$ iptables -t nat -A POSTROUTING -o eth0 -p tcp -m multiport --dports 80,443
-j MASQUERADE
```

Voici la dernière configuration d'iptables pour la table NAT (toutes les chaînes):

```
$ iptables -t nat -S
-P PREROUTING ACCEPT
-P POSTROUTING ACCEPT
-P OUTPUT ACCEPT
-A PREROUTING -i eth0 -p tcp -m tcp --dport 22043 -m comment --comment "DNAT
SSH for .43" -j DNAT --to-destination 10.8.8.43:22
-A POSTROUTING -o eth0 -p icmp -j MASQUERADE
-A POSTROUTING -o eth0 -p tcp -m multiport --dports 80,443 -j MASQUERADE
```

Ne pas oublier de sauvegarder les règles iptables:

```
$ service iptables save
```

CAS 2 : Configuration de type pare feu

Politique par défaut

Pour fixer les politiques par défaut (cad: ce qui se passe quand aucune règle ne correspond - ne matche pas), ici, on refuse tout

```
$ iptables -P INPUT DROP
$ iptables -P OUTPUT DROP
$ iptables -P FORWARD DROP
```

Pour logguer tout ce qu'on jette :

```
$ iptables -N LOG_DROP
$ iptables -A LOG_DROP -j LOG --log-prefix '[IPTABLES DROP] : '
$ iptables -A LOG_DROP -j DROP
```

Les trois dernières règles doivent être :

```
$ iptables -A FORWARD -j LOG_DROP
$ iptables -A INPUT -j LOG_DROP
$ iptables -A OUTPUT -j LOG_DROP
```

Définition du trafic autorisé

Recommencer en remplaçant -j DROP, par -j ACCEPT et a la place de -j LOG*DROP il faut mettre -j LOG*ACCEPT

Pour accepter tout ce qui se passe sur l'interface lo (sinon ce n'est pas la peine d'activer le réseau !) :

```
$ iptables -A INPUT -i lo -j ACCEPT
$ iptables -A OUTPUT -o lo -j ACCEPT
```

Pour accepter tout ce qui se passe sur le réseau local 192.168.1.0 :

```
$ iptables -A INPUT -s 192.168.1.0/24 -j ACCEPT
$ iptables -A OUTPUT -d 192.168.1.0/24 -j ACCEPT
$ iptables -A FORWARD -s 192.168.1.0/24 -j ACCEPT
```

Pour accepter les résolutions de nom (ie: le dns) :

```
$ iptables -A INPUT -i eth0 --protocol udp --source-port 53 -j ACCEPT
$ iptables -A OUTPUT -o eth0 --protocol udp --destination-port 53 -j ACCEPT
$ iptables -A INPUT -i eth0 --protocol tcp --source-port 53 -j ACCEPT
$ iptables -A OUTPUT -o eth0 --protocol tcp --destination-port 53 -j ACCEPT
```

Pour accepter le traffic web (on veut surfer!) :

```
$ iptables -A INPUT -i eth0 --protocol tcp --source-port 80 -m state --state
ESTABLISHED -j LOG_ACCEPT
$ iptables -A OUTPUT -o eth0 --protocol tcp --destination-port 80 -m state -
-state NEW,ESTABLISHED -j LOG_ACCEPT
```

La première ligne pour accepter ce qui entre sur notre interface eth0 sur le port 80 (le port http) si c'est une connexion déjà établie, la seconde pour accepter ce qui sort sur eth0 sur le port 80 si c'est une nouvelle connexion ou si c'est une connexion déjà établie.

Autorisation des services

De manière générale, le numéros de port se trouvent dans /etc/services.

- Pour autoriser le ssh il faut préciser le port 22;
- pour autoriser l'irc (Internet Relay Chat), le port 6667 (ou celui utilisé pour se connecter au serveur);
- pour le smtp (envoi d'emails), le port 25;
- pour le pop3 (réception d'emails), le port 110;
- pour le imap (réception d'emails), les ports 143 et 220 (imap3);
- pour le cvs, le port 2401;pour le https, le port 443.

Configuration des clients

Sur chaque machine devant être cachée par le pare-feu (ou devant partager la connexion avec la machine qui est connectée à internet), il faut ajouter une route par defaut :

```
$ route add default gw 192.168.1.1
```

Si la machine connectée à internet a comme ip : 192.168.1.1. Par exemple, il suffit avec une Red Hat ou Mageia d'éditer :

```
$ vi /etc/sysconfig/network
```

et d'ajouter dedans :

```
GATEWAY=192.168.1.1
```

puis de redémarer le réseau :

```
$ service network restart
```

CAS 3 : Configuration de type edgerouter

Description de la configuration

Le pare-feu dispose d'un port eth0 paramétré dans le réseau 172.31.0.0/24.

Un serveur HTTP dédié est connecté sur un réseau isolé à l'adresse 10.0.4.2 (dans un sous réseau en dehors de la plage 172.31.0.0/24 du réseau local).

Pour définir une règle d'acheminement des requêtes HTTP entrantes vers l'adresse 10.0.4.2 NAT appelle une table PREROUTING pour transférer les paquets à leur destination correcte

Forward et NAT Rules

La stratégie FORWARD permet de contrôler où les paquets peuvent être routés dans un réseau local. Par exemple, pour autoriser le transfert pour l'intégralité du réseau local, les règles suivantes peuvent être définies:

```
iptables -A FORWARD -i eth0 -j ACCEPT
iptables -A FORWARD -o eth0 -j ACCEPT
```

Avec cette règle la passerelle achemine les paquets d'un noeud LAN vers son noeud cible, en passant tous les paquets via son périphérique eth0.

Masquerade

L'acceptation des paquets transférés via le périphérique IP interne du pare-feu permet aux nœuds LAN de communiquer entre eux; Cependant, ils ne sont toujours pas autorisés à communiquer en externe sur Internet. Pour permettre aux nœuds LAN avec des adresses IP privées de communiquer avec des réseaux publics externes, configurer le pare-feu pour le masquage IP, qui masque les requêtes des nœuds LAN avec l'adresse IP du périphérique externe du pare-feu (dans ce cas, eth0):

```
$ iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

La règle utilise la table de correspondance de paquets NAT (-t nat) et spécifie la chaîne POSTROUTING intégrée pour NAT (-A POSTROUTING) sur le périphérique de réseau externe du pare-feu (-o eth0). POSTROUTING permet de modifier les paquets lorsqu'ils quittent le périphérique externe du pare-feu. La cible -j MASQUERADE est spécifiée pour masquer l'adresse IP privée d'un noeud avec l'adresse IP externe du pare-feu / passerelle.

DMZ et iptables

Les règles iptables peuvent être configurées pour acheminer le trafic vers certaines machines, telles qu'un serveur HTTP ou FTP dédié, dans une zone démilitarisée (DMZ) - un sous-réseau local spécial dédié à la fourniture de services sur un opérateur public tel qu'Internet

Pour rendre le serveur interne disponible en externe, il faut utiliser la cible -j DNAT de la chaîne PREROUTING dans NAT pour spécifier une adresse IP de destination et un port où les paquets entrants demandant une connexion à votre service interne peuvent être transmis. Par exemple, pour transférer les demandes HTTP entrantes vers le serveur Apache HTTP Server dédié à l'adresse 10.0.4.2, exécuter la commande suivante:

```
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --
destination 10.0.4.2:80
```

Cette règle spécifie que la table NAT utilise la chaîne PREROUTING intégrée pour transférer les demandes HTTP entrantes exclusivement vers l'adresse IP de destination répertoriée 10.0.4.2.

Si la politique par défaut de DROP est définie, il faut ajouter une règle dans la chaîne FORWARD pour autoriser le transfert des demandes HTTP entrantes afin que le routage NAT de destination puisse être possible. Pour ce faire, exécuter la commande suivante:

```
iptables -A FORWARD -i eth0 -p tcp --dport 80 -d 10.0.4.2 -j ACCEPT
```

Cette règle permet de transférer les demandes HTTP entrantes du pare-feu à la destination prévue du serveur Apache HTTP Server derrière le pare-feu.

CAS spécifique pour ftp

L'une des principales raisons pour l'utilisation du mode FTP passif est qu'il permet de passer plus facilement à travers les pare-feux. Cependant, certains utilisateurs peuvent ne pas savoir comment activer le mode FTP passif ou avoir des clients incompatibles.

Pour permettre aux requètes FTP de passer correctement les pare-feu, iptables devant être capable de marquer et suivre ces connexions, il faut activer le module du noyau nf_conntrack_ftp (c'est lui qui suit - track en anglais - les connexions ftp).

Pour nater (en utilisant le masquerading par exemple) les connexions ftp, il faut activer le module du noyau nf_nat_ftp.

Pour les noyaux 2.6.19 ou antérieurs, les noms des modules sont **ip_nat_ftp** et **ip_conntrack_ftp** et ils sont normalement chargés à partir de `/lib/modules/<version du noyau>/kernel/net/ipv4/netfilter/`.

Activation des modules

```
$ modprobe nf_conntrack_ftp
$ modprobe nf_nat_ftp
```

Ajouter dans /etc/sysconfig/iptables-config les modules:

```
IPTABLES_MODULES="nf_conntrack_ftp nf_nat_ftp"
```

Paramétrage d'iptables

Autoriser le tafic sur le port 21

Autoriser le tafic NEW ESTABLISHED etRELATED sur le port 21

```
$ iptables -A INPUT -p tcp --dport 21 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
$ iptables -A OUTPUT -p tcp --sport 21 -m state --state ESTABLISHED -j
ACCEPT
```

Autoriser le trafic pour les données en mode actif

Autoriser le trafic ESTABLISHED et RELATED sur le port 20 (on a besoin de nf_conntrack_ftp)

```
$ iptables -A INPUT -p tcp --dport 20 -m state --state ESTABLISHED,RELATED -
j ACCEPT
$ iptables -A OUTPUT -p tcp --sport 20 -m state --state ESTABLISHED -j
ACCEPT
```

Autoriser la connexion pour les données en mode passif

Autoriser le trafic ESTABLISHED et RELATED sur les port 1024 à 65535 (on a besoin de nf_conntrack_ftp)

```
$ iptables -A INPUT -p tcp --dport 1024:65535 -m state --state
ESTABLISHED,RELATED -j ACCEPT
$ iptables -A OUTPUT -p tcp --sport 1024:65535 -m state --state ESTABLISHED
-j ACCEPT
```

[1)]
Si aucun chiffre n'est spécifié à la suite (#iptables -I INPUT –dport 80 -j ACCEPT), la règle est ajoutée au début de la chaîne spécifiée