Licence 3 Informatique - SR2

Systèmes et Réseaux II

Chapitre 5

Filtrage

Éric Leclercq
Département IEM / uB
Eric.Leclercq@u-bourgogne.fr
Bureau R8 Aile H

8 février 2024

Plan du chapitre

Bibliographie

Ouvrages complémentaires sur le sujet :

- Linux iptables Pocket Reference, Firewalls, NAT and Accounting, Gregor N. Purdy, O'Reilly, August 2004
- Linux Security Cookbookn Daniel J. Barrett, Richard E. Silverman, Robert G. Byrnes, O'Reilly, June 2003
- Linux Server Security, Second Edition, Michael D. Bauer, O'Reilly, January 2005

Principes de la translation d'adresses

Permettre aux machines de réseaux privés de dialoguer sur Internet

- NAT = Network Address Translation
- Utilise généralement un routeur d'entrée (border router)
- Intercepte les packets, les réécrits en modifiant adresse source, port source
- Permet donc un multiplexage des connections via des correspondances de port
- Fonctionne au moyen une table de correspondance entre des couples (adresses source, port) internes et externes réécrits (adresse destination, port)
- Sous linux NAT et ip-masquerading sont synonymes
- Ce n'est pas un mécanisme qui assure la sécurité !

Netfilter / iptables

Un filtre de paquets est un **module du noyau** qui inspecte l'en-tête des paquets qui transitent par les cartes réseau et décide du sort du paquet entier

- Il peut décider de détruire le paquet (faire comme si il n'avait jamais été reçu), accepter le paquet (le laisser passer), ou quelque chose de plus complexe (par exemple le loguer, compter le nombre de paquets par seconde) module ou directement dans le code du noyau (Netfilter).
- Sous Linux, le module de filtrage (netfilter) est très complet, il propose un véritable mécanisme de firewall avec états. Le principe de base reste simple : regarder les en-têtes et décider du sort du paquet.
- iptables travail au niveau 3 de la couche OSI, pour avoir un filtre de niveau 2 (Ethernet) utiliser ebtables (Ethernet Bridge Table)
- Par abus de langage nous confondrons iptables avec Netfilter.

iptables

- La commande `iptables` insère et retire des règles de la table de filtrage du noyau
- Les règles seront perdues au reboot
- `iptables` utilise la notion de table, de liste, de chaîne ou règles complexes.
- les tables groupent les règles par fonctions (nat, filtrage etc.)

```
## Chaine qui bloque les connections sauf celles qui viennent
## de l'intérieur
 iptables -N block
 iptables -A block -m state --state ESTABLISHED,RELATED -j ACCEPT
 iptables -A block -m state --state NEW -i ! ppp0 -j ACCEPT
 iptables -A block -j DROP

## Lancer la chaîne block à partir des chaînes INPUT et FORWARD.
 iptables -A INPUT -j block
 iptables -A FORWARD -j block
```

Les listes principales

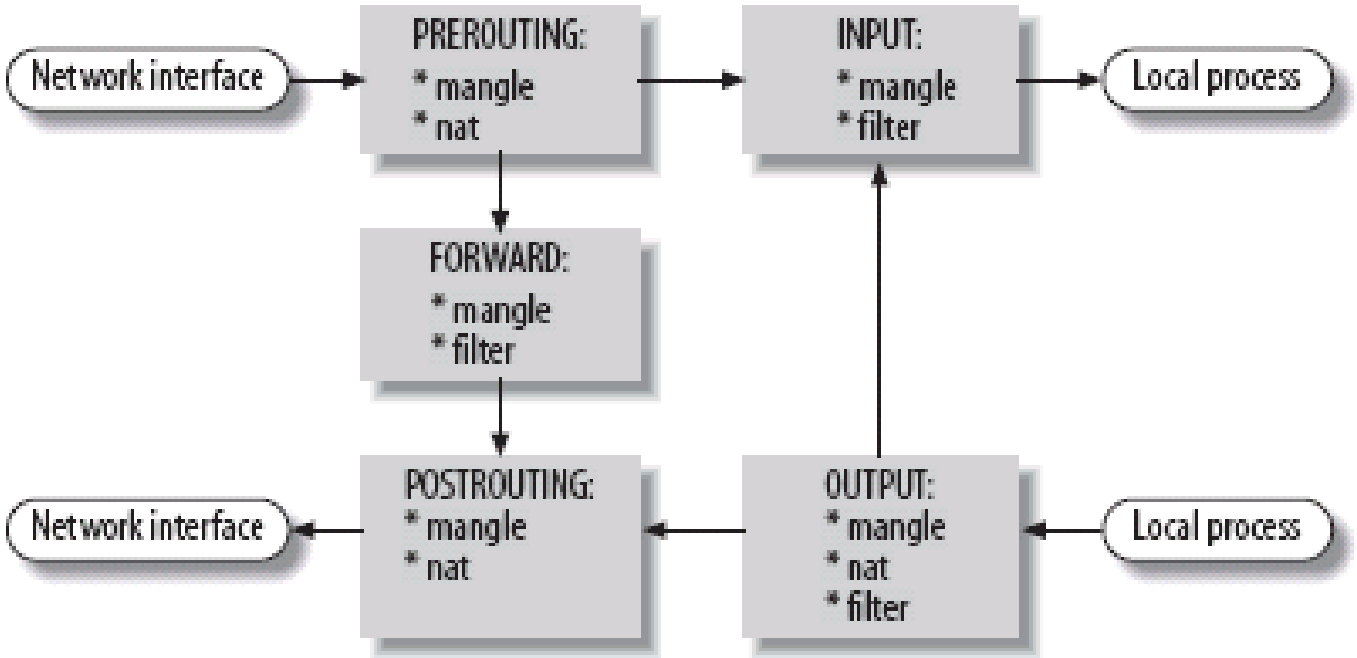
- Le noyau contient par défaut cinq listes (`INPUT`, `OUTPUT`, `FORWARD`, `PREROUTING`, `POSTROUTING`) appelée hook points
- Les listes sont corrélées aux tables mais pas liées
- Les éléments des listes sont appelés chaînes ou règles
- On peut considérer qu'une chaîne est une série de règles élémentaires (atomiques).
- Chaque règle élémentaire est une condition que doit satisfaire le paquet, si la règle ne convient pas, la chaîne suivante est examinée, finalement, si il ne reste plus de chaîne, le noyau regarde la chaîne par défaut pour décider de l'action à exécuter

Les listes principales

L'enchaînement des trois listes est le suivant :

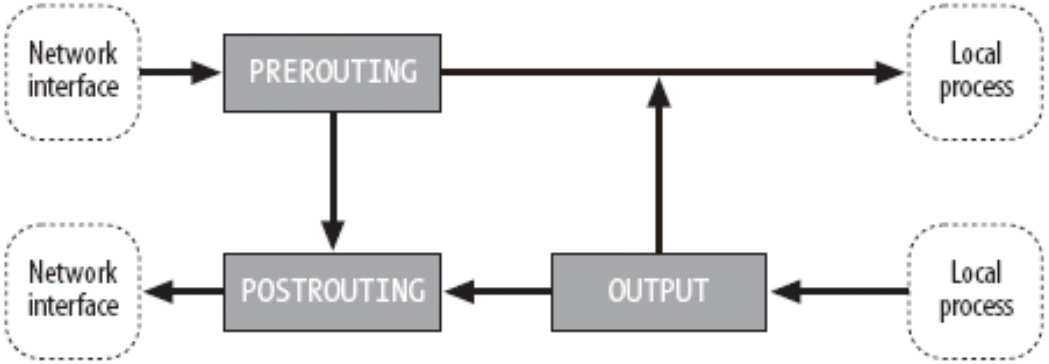
1. Quand un paquet arrive le noyau regarde en premier la destination de ce paquet (prise en charge). Si le paquet est destiné à la machine il est transmis la chaîne `INPUT`. Si il la passe, les processus qui attendent le paquet le recevront.
2. Si le noyau n'a pas de forwarding activé, ou qu'il ne sait pas comment forwarder le paquet, le paquet est tué.
3. Si le forwarding est autorisé et que le paquet est destiné à une autre interface réseau, le paquet va directement à la chaine `FORWARD`. Si il est accepté par une des chaînes, il sera envoyé.
4. Finalement, un programme qui tourne sur la machine peut envoyer des paquets. Ces paquets passeront par la chaîne `OUTPUT` immédiatement : si elle active `ACCEPT`, alors le paquet continue vers l'interface à laquelle il est destiné.

Ordre des listes dans les tables

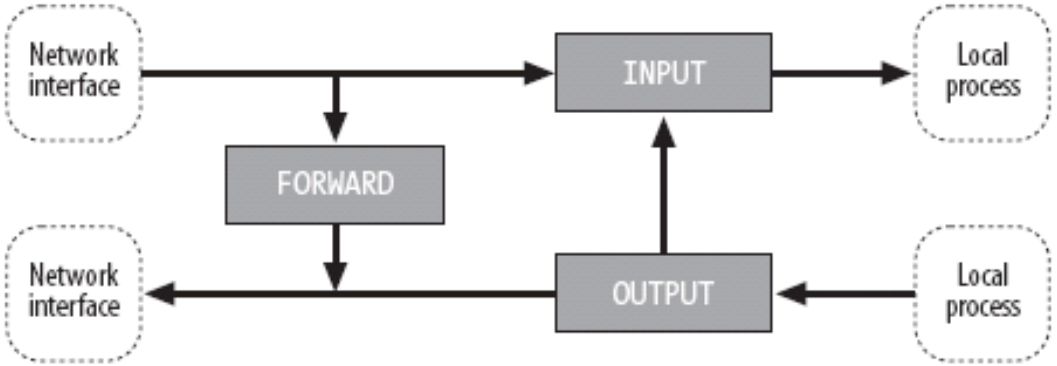


Ordre des listes dans les tables

Pour la table nat :



Pour la table filter :



iptables : les chaînes de base

Listes	Table	Description
PREROUTING	nat, mangle	Par cette chaîne passeront les paquets entrant dans la machine avant routage
INPUT	filter	Cette chaîne traitera les paquets entrants avant qu'ils ne soient passées aux couches supérieures (les applications).
FORWARD	filter	Ce sont les paquets uniquement transmis par la machine sans que les applications n'en aient connaissance.
OUTPUT	filter, nat, mangle	Cette chaîne sera appelée pour des paquets envoyés par des programmes présents sur la machine.
POSTROUTING	nat	Les paquets prêts à être envoyés (soit transmis, soit générés) seront pris en charge par cette chaîne.

Action	Signification
ACCEPT	Les paquets envoyés vers cette cible seront tout simplement acceptés et pourront poursuivre leur cheminement au travers des couches réseaux.
DROP	Cette cible permet de jeter des paquets qui seront donc ignorés.
REJECT	Permet d'envoyer une réponse à l'émetteur pour lui signaler que son paquet a été refusé.
LOG	Demande au noyau d'enregistrer des informations sur le paquet courant. Cela se fera généralement dans le fichier /var/log/messages (selon la configuration du programme syslogd).
MASQUERADE	Cible valable uniquement dans la chaîne POSTROUTING de la table nat. Elle change l'adresse IP de l'émetteur par celle courante de la machine pour l'interface spécifiée. Cela permet de masquer des machines et de faire par exemple du partage de connexion.
SNAT	Egalement valable pour la chaîne POSTROUTING de la table nat seulement. Elle modifie aussi la valeur de l'adresse IP de l'émetteur en la remplaçant par la valeur fixe spécifiée.
DNAT	Valable uniquement pour les chaînes PREROUTING et OUTPUT de la table nat. Elle modifie la valeur de l'adresse IP du destinataire en la remplaçant par la valeur fixe spécifiée.
RETURN	Utile dans les chaînes utilisateurs. Cette cible permet de revenir à la chaîne appelante. Si RETURN est utilisé dans une des chaînes de base précédente, cela est équivalent à l'utilisation de sa cible par défaut.

Applications

Les applications de Netfilter sont multiples :

- Packet filtering
- Accounting : pour auditer le volume de traffic par protocole (applicatif par exemple)
- Connection tracking : tres utile pour ftp
- Packet mangling
- Network address translation (NAT) : type de mangling qui agit sur source et destination. Existe sous deux formes : SNAT et DNAT
- Masquerading : forme spécifique de SNAT
- Port Forwarding
- Load Balancing

Translation d'adresses et protocoles CS inversés

- Tous les protocoles ne supportent pas la translation d'adresse
- Les protocoles pour lesquels le serveur est dans le réseau privé sont mal adaptés à la translation d'adresse :
  - affichage X11
  - FTP
- Il faut alors un mécanisme supplémentaire capable de suivre une session ou établir un relai (`ssh -X`, `ip_contrack_ftp`)
- Il est préférable de ne pas utiliser X11 sur internet, lui préférer un protocole moins gourmand en bande passante plus sûre (VNC, ssh)

Où en sommes nous ?

1. Translation d'adresses
2. Filtrage : les principes
3. Les éléments de Netfilter
4. **Détails de la commande iptables**

Options d'iptables

Les principales options d'iptables sont :

- Créer une nouvelle chaine (-N) ou plutôt une liste de chaînes
- Effacer une chaîne (-X)
- Changer la règle par défaut pour une chaîne (-P)
- Lister les règles (-L) généralement utilisée sous la forme `iptables -L -v` pour visualiser aussi les compteurs
- Retirer les règles d'une chaîne (-F).
- Mettre à zéro les compteurs de bits et de paquets d'une chaîne (-Z)

iptables

Il y a plusieurs manières de manipuler une règle dans une liste de chaînes :

- Ajouter une nouvelle règle à une liste (-A) ;
- Insérer une nouvelle règle à une position dans une liste (-I) ;
- Remplacer une règle à une position dans une liste (-R) ;
- Supprimer une chaîne à une position dans une liste (-D) ;
- Supprimer la première règle qui convient dans une chaine (-D).

Les spécifications des règles de filtrage peuvent concerner :

- la source, la destination
- un protocole
- une interface
- ou des fragments de paquet

iptables : source et destination

Les adresses IP source (`-s`, `--source` ou `--src`) et destination (`-d`, `--destination` ou `--dst`) peuvent être specifiées de 4 façons :

1. le nom complet, comme `localhost` ou `panda.ville-dijon.fr`;
2. l'adresse IP comme `10.11.12.1`;
3. un groupe d'IPs, comme `192.168.207.0/24` ou `192.168.207.0/255.255.255.0`. Elles specifient toutes deux les adresses de `192.168.207.0` à `192.168.207.255` inclus. Les nombres après le `/` indiquent quelle partie des adresses IP a de la signification. `/32` ou `/255.255.255.255` est le défaut (correspond à toutes les adresses IP).
4. pour spécifier toutes les adresses IP `/0` peut être utilisé, comme dans la règle : `iptables -A INPUT -s 0/0 -j DROP`

iptables : source et destination

- Beaucoup d'options comme `-s` (ou `--source`) et `-d` (`--destination`) peuvent avoir leurs arguments précédés de `!` (négation) pour correspondre aux adresses différentes égales à celles données.
- Le `!` permet donc la négation d'une règle.
- iptables est extensible, ce qui veut dire que le noyau et le programme iptables peuvent être étendus pour avoir de nouvelles capacités.
- Les extensions au noyau sont normalement situées dans le répertoire des modules du kernel comme `/lib/modules/2.X.Y/net`. Elles sont chargées à la demande.

iptables (filtrage sur protocole et port)

- Les options `-i` (ou `--in-interface`) et `-o` (ou `--out-interface`) spécifient le nom d'une interface à laquelle le paquet doit correspondre.
- Les paquets qui traversent la chaîne `INPUT` n'ont pas encore d'interface de sortie donc, une règle utilisant `-o` dans cette chaîne n'est pas valide.
- Les paquets traversant la chaîne `OUTPUT` n'ont pas d'interface d'entrée, donc toute règle utilisant `-i` dans cette chaîne n'est pas valide. Seuls les paquets traversant la chaîne `FORWARD` ont une interface d'entrée et de sortie.

iptables (filtrage sur protocole et port)

Les extensions TCP sont automatiquement chargées si `-p tcp` est spécifié.

- `--tcp-flags` : suivi d'un `!` optionnel, ensuite 2 chaînes de caractères permettent de filtrer suivant des drapeaux TCP spécifiques. La première chaîne de drapeaux est le masque : une liste de drapeaux à examiner. La deuxième chaîne de drapeaux dit lequel doit être présent. Par exemple :

  ```
  iptables -A INPUT -p tcp --tcp-flags ALL SYN,ACK -j DENY
  ```

- `--source-port` : suivi d'un `!` optionnel optionnel, ensuite soit un port TCP seul, ou un bloc de ports. Les ports peuvent être des noms de ports, listés dans `/etc/services`, ou des nombres. `--sport` est synonyme de `--source-port`.
- `--destination-port` et `--dport` ils spécifient la destination plutôt que la source qui convient.

iptables (un premier script naïf)

```
echo 1 > /proc/sys/net/ipv4/ip_forward
/sbin/iptables -P INPUT ACCEPT
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -t nat -P PREROUTING ACCEPT
/sbin/iptables -t nat -P POSTROUTING ACCEPT
/sbin/iptables -t nat -P OUTPUT ACCEPT
/sbin/iptables -F
/sbin/iptables -t nat -F
/sbin/iptables -A INPUT -i lo -j ACCEPT
/sbin/iptables -A OUTPUT -o lo -j ACCEPT
/sbin/iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
/sbin/iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
/sbin/iptables -t nat -A POSTROUTING -s 10.21.19.0/24 -
j MASQUERADE
#redirection du port 80 pour proxy squid transparent
/sbin/iptables -t nat -A PREROUTING -i eth1 -p tcp --
dport 80 \
-j REDIRECT --to-port 3128
```

iptables (squelette de script)

```
echo 1 > /proc/sys/net/ipv4/ip_forward
# PARTIE A MODIFIER
public=ppp0
prive=eth0
reseauprive="10.11.12.0/24"
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
iptables -F
iptables -t nat -F
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A OUTPUT -o $public -p tcp --dport http -j ACCEPT
iptables -A INPUT -i $public  -p tcp --sport http -j ACCEPT
iptables -A OUTPUT -o $public -p tcp --dport https -j ACCEPT
iptables -A INPUT -i $public  -p tcp --sport https -j ACCEPT
```

Exemples pratiques

- Pour fowarder les connexion entrantes vers un serveur web dans le domaine privé : `iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j DNAT --to-destination 192.168.1.3 :8080`
- Dans une architecture avec un proxy HTTP (comme Squid) configuré pour fonctionner de manière transparente sur le filtre et écoutant sur le port 8888, la règle suivant permet de rediriger la trafic sortant vers le proxy : `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8888`
- Distribué la charge sur plusieurs machines : utiliser les extensions `nth match` et les actions DNAT

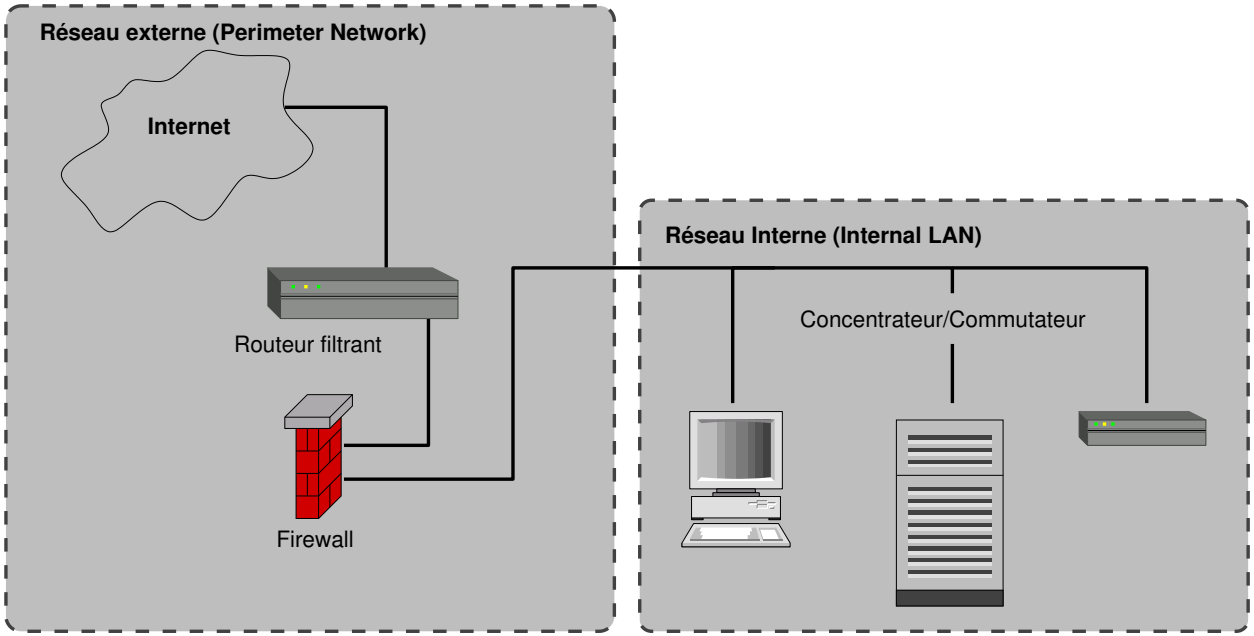
Outils d'analyse

- Ethereal Network protocol analyzer http://www.ethereal.com/
- Nessus Remote security scanner http://www.nessus.org/intro.html
- nmap Network mapper http://www.insecure.org/nmap/
- ntop Network traffic http://ntop.ethereal.com/ntop.html
- tcpdump pour la capture des paquets http://www-nrg.ee.lbl.gov/

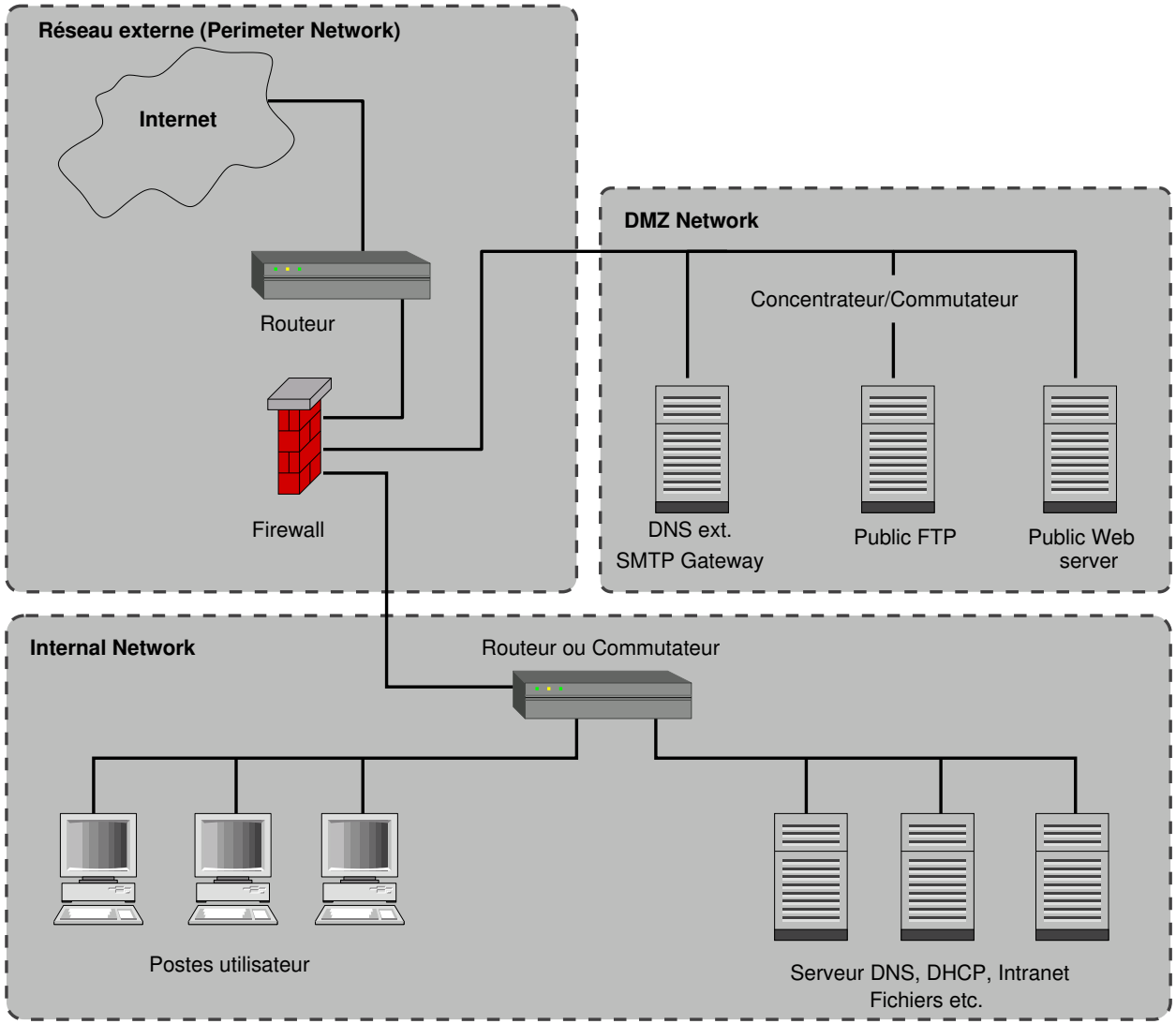
Où en sommes nous ?

1. Translation d'adresses
2. Filtrage : les principes
3. Les éléments de Netfilter
4. Détails de la commande iptables
5. **Architectures de filtrage type**

Architecture simple firewall

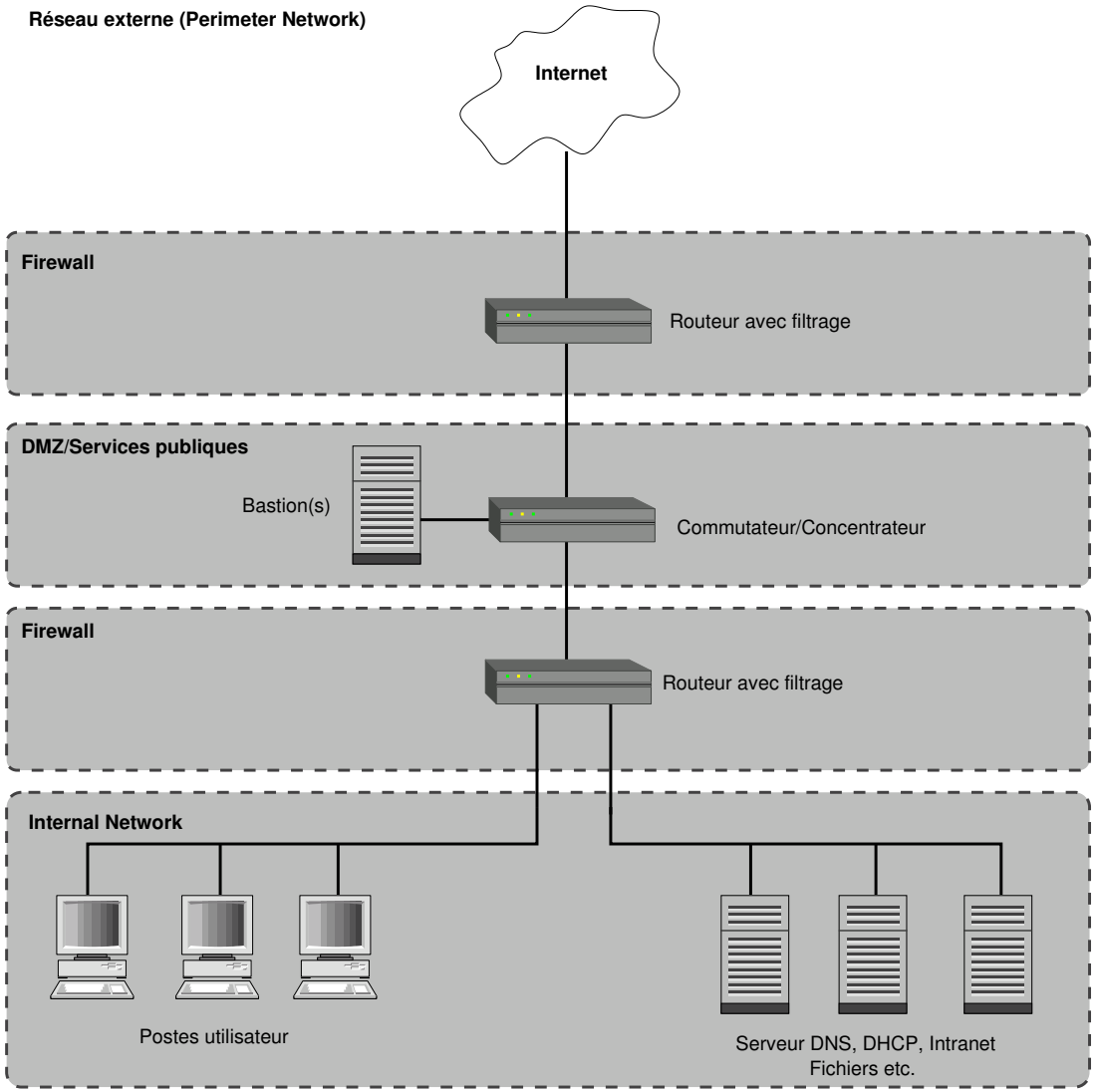


Architecture 3 zones



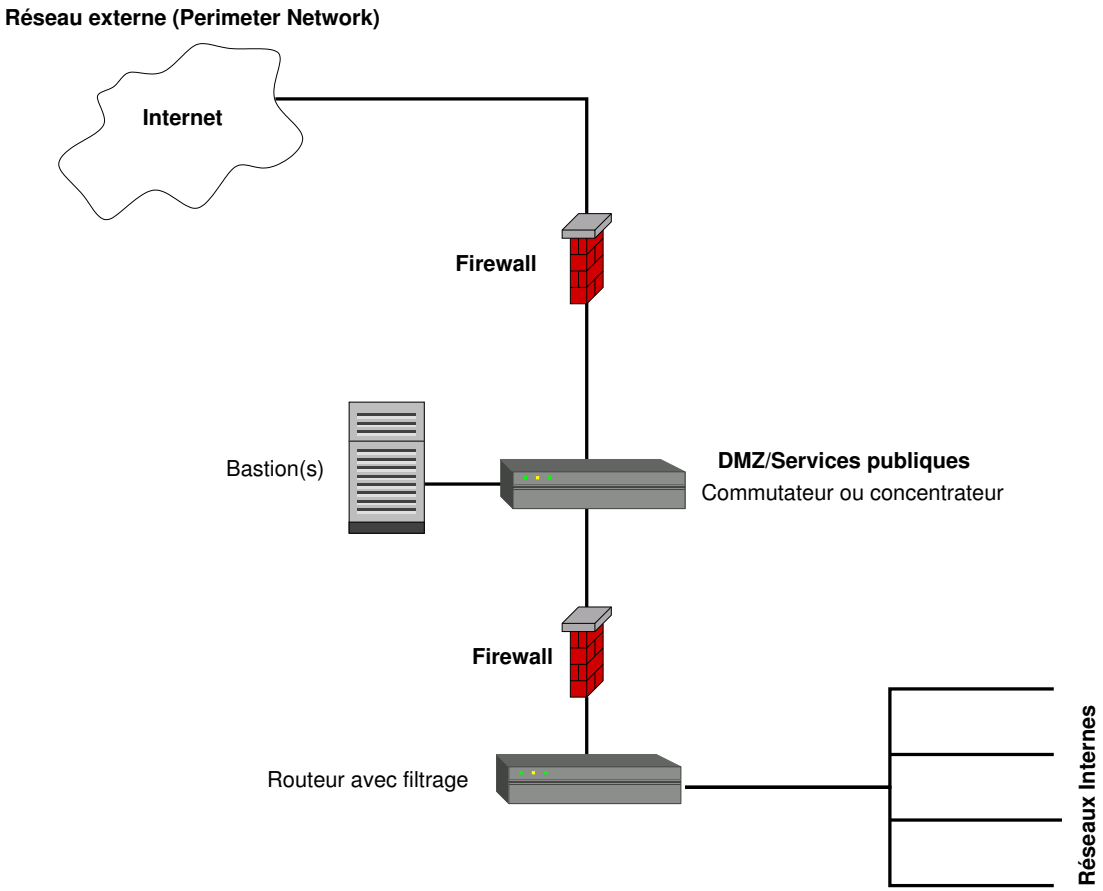
Architecture type Réseau écran

Traduction anglaise de Weak Screened-Subnet Architecture.



Architecture type Réseau écran forte

Traduction anglaise de Strong Screened-Subnet Architecture.



Où en sommes nous ?

L'outil `fail2ban`

Un serveur connecté en permanence constitue une cible pour les attaques :

- l'utilisation d'un pare-feu réduit les risques mais ce n'est pas suffisant
- il est nécessaire de contrôler les accès protégés par mot de passe
- pour éviter des attaques par force brute (test des mots de passe faibles)

`fail2ban` permet de surveiller l'activité des logs (journaux) de certains services, tel que SSH ou Apache

Principe : lorsqu'un grand nombre d'authentifications échoue `fail2ban` va générer une règle iptable, cette règle aura pour but d'interdire pendant une durée déterminée les connexions depuis l'adresse IP susceptible d'être un attaquant.

fail2ban : mise en place

```
apt-get install fail2ban
cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
vi /etc/fail2ban/jail.conf
```

Exemple de fichier de configuration :

```
ignoreip = 127.0.0.1/8          #adresses IP ignorées par les actions de fail2ban
bantime  = 600                  #temps de bannissement en secondes
maxretry = 5
banaction = iptables-multiport
protocol = tcp
enabled = false
...
[sshd]
enabled  = true
port     = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
...
[recidive]
enabled = true
logpath  = /var/log/fail2ban.log
banaction = %(banaction_allports)s
bantime  = 1w
findtime = 1d
```

fail2ban : administration

```
# Administrer le service
systemctl status|start|stop|restart fail2ban.service

# Etat du service
fail2ban-client status sshd
systemctl status fail2ban

# Bannir ou débannir
fail2ban-client set sshd unbanip 11.12.13.14
fail2ban-client set sshd banip 11.11.11.11

# Monitorer
awk '($(NF-1) = /\[sshd\] Ban/){print $NF}' \
/var/log/fail2ban.log  | sort | uniq -c | sort -n
```