Julien Iguchi-Cartigny

Associate Professor, XLIM, University of Limoges, France

Firewall

Buts

- Comprendre le fonctionnement d'un firewall

Préparation

Nous réutilisons le *lab* déployé durant le précédent TP, mais nous nous limiterons aux actions suivantes avant de commencer le présent TP:

- démarrer le TP
- activer le mécanisme DHCP sur pc1 et pc2 pour obtenir une adresse IP
- activier le NAT sur *gateway*

Introduction

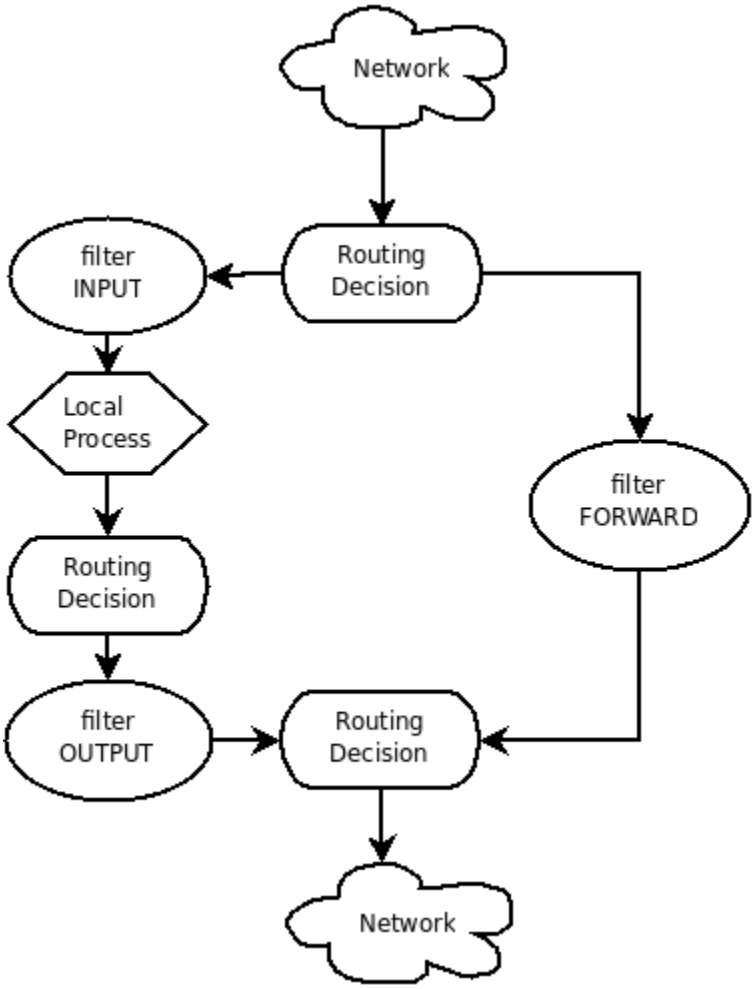
La commande activant le mécanisme de translation d'adresse permet d'agir sur IPTABLES, un mécanisme permettant de rejeter ou modifier les paquets arrivant ou sortant de la machine en fonction de certains critères.

> Nous avons eu l 'exemple dans le précédent TP en activant le mécanisme NAT (*i.e.* la table `nat` de iptables): celui ci va modifier chaque paquet sortant de notre réseau (en changeant l'adresse IP de l'émetteur et le numéro de port source) afin de pouvoir reconnaître à quel destinataire doit être transmis les réponses.

Dans cette partie, nous nous intéresserons uniquement à la table *filter* en charge de filtrer (accepter / refuser) les paquets entrants et sortants. Pour voir le contenu de cette table, la commande est la suivante:

```
gateway:~# iptables -L -v --line-number
```

On peut voir pour l'instant qu'il n'y a aucune règle ajoutée, la configuration par défaut (*policy*) de chaque chaîne est de ne rien faire (ACCEPT) sur tous les paquets de chaque chaîne (INPUT, OUTPUT, FORWARD).



(schéma réduit à partir du schéma de traversé des tables iptables)

Une chaîne s'applique aux paquets entrants (INPUT), sortants (OUTPUT) ou en transit (FORWARD) dans le cas d'un transfert de paquet d'une interface à une autre (pour un routeur par exemple).

Pour ajouter une règle, il faut 3 choses:

- Une chaîne où l'appliquer:
  - *INPUT* permet d'appliquer un filtrage à tous les paquets entrants
  - *OUTPUT* ...à tous les paquets sortants
  - *FORWARD* ...à tous les paquets transitants d'une interface vers une autre

- une (ou plusieurs) conditions pour que la règle s'applique (port ? procotole ?)
  - *-s* l'adresse source, *-d* l'adresse destination
  - *-p* le protocole (paramètres possibles: *tcp*, *udp*, *icmp*)

    - Pour *tcp* et *udp* il existe des paramètres pour: *--sport* le port source, *--dport* le port destination
    - *-i* interface entrante, *-o* interface de sortie

- l'action à effectuer:
    - ACCEPT: laisser passer le paquet
    - REJECT: refuser le paquet en renvoyant un message d'erreur
    - DROP: refuser le paquet en l'oubliant

Exemples

- Rejet (*-j REJECT*) de tous les paquets entrants (*-A INPUT*) ayant comme source l'adresse IP 192.168.0.3:

  gateway:~# iptables -A INPUT -s 192.168.0.3 -j REJECT

- Accepte (*-j ACCEPT*) tous les paquets sortants (*-A OUTPUT*) de type UDP (*-p udp*) ayant comme port source 53 (*--sport 53*) et sortant par l'interafce eth0 (*-o eth0*)

  gateway:~# iptables -A OUTPUT -p udp --sport 53 -o eth0 -j ACCEPT

Questions

La meilleure politique de sécurité est de bloquer tous les ports et de les ouvrir un par un. C'est ce que nous allons faire sur *gateway* et *server*. Nous allons choisir le blocage de tout paquet par défaut (*i.e.* s'il n'y a aucune règle qui répond au paquet) :

```
gateway:~# iptables -P INPUT DROP
gateway:~# iptables -P OUTPUT DROP
gateway:~# iptables -P FORWARD DROP

server:~# iptables -P INPUT DROP
server:~# iptables -P OUTPUT DROP
server:~# iptables -P FORWARD DROP
```

Activer les règles suivantes sur *server* et *gateway* :
* toutes les machines du réseau local peuvent envoyer une demande DHCP à *server* (et recevoir la réponse)
* toutes les machines du réseau local peuvent envoyer une requête DNS à *server* (et recevoir la réponse)
* *server* peut envoyer une requête DNS à *dns* (et recevoir la réponse)

* toutes les machines du réseau local peuvent envoyer un paquet ICMP sur les machines sur Internet (et recevoir la réponse), mais PAS pinger *gateway* ni *server*
* toutes les machines du réseau local peuvent envoyer une requête HTTP à *httpd* (et recevoir la réponse)

Attention de ne pas ouvrir trop largement votre *firewall* et ainsi laisser passer des paquets non autorisés ! Essayer d'avoir les règles les plus complètes possibles: "*je bloque / accepte les paquet du protocole XXX de/vers le(s) port(s) YYY de la machine WWW sur l'interface ZZZ*"

Mode Statefull

Il y a plusieurs limitations aux blocage que nous venont de mettre en place. Le plus important est que autoriser les paquets se fait de manière bidirectionelle, il est donc impossible d'empêcher une machine B de pinger A si le firewall autorise A a pinger B.

Plus généralement, nous voudrions autoriser les flots et non les paquets. Une idée est d'allors d'autoriser tous les paquets dont les communications ont déjà commencés, et d'autoriser ou pas les débuts de communication.

Penser à remettre à zéro votre firewall

Voici les commandes pour autoriser tous paquets donc la communication est déjà en place à traverser le firewall:

```
gateway:~# iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
gateway:~# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
gateway:~# iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Le paramètre `state` indique l'état de l'échange: demande de connexion (NEW) ou établie (ESTABLISHED et/ou RELATED).

Après, il nous suffit de mettre les règles pour autoriser un début de connexion. Pour autoriser une ouverture de connexion vers le port 53 d'une machine distante:

```
gateway:~# iptables -A INPUT -m state --state NEW -p udp --dport 53 -i eth0 -j ACCEPT
```

Réécrire toutes les règles des questions précédentes en suivant la même

logique

Annexe: commandes iptables

N'hésiter pas à compléter si vous pensez que des commandes manquent à la compréhension de ce TP

Voir l'ensemble des règles (et le numéro de chacune):

```
pc:~# iptables -L -v --line-number
```

Vider toute la table de filtrage:

```
pc:~# iptables -F
```

Effacer la règle numéro 3 de la chaîne INPUT:

```
pc:~# iptables -D INPUT 3
```

Annexe: références

Référence sur IPTables:

- le MAN de iptables

Si vous voulez aller plus loin:

- le tutorial iptables (complet et exhaustif)