Administration Réseau

Réseau et sécurité, mise en oeuvre

McInfo4 - Réseaux

Département d'informatique
IUT Bordeaux 1

Mars 2009

Mise en oeuvre

1. Analyse des besoins, politique
2. Choix d'une architecture
3. Mise en place
   - du routage
   - du filtrage
   - des mandataires

Le filtrage

- filtrage sans états (*stateless*),
- suivi de connexions
- filtrage applicatif

Le filtrage sans états

Le filtrage sans états (*stateless*) se base essentiellement sur l'examen des entêtes des paquets IP

- Protocole (ICMP, TCP, UDP, ...)
- Interface de provenance
- Adresses IP de provenance et de destination, Ports
- Flags ACK, SYN, FIN, etc.

Règles de filtrage

Exemple

"Accepter les paquets TCP entrants à destination du port 25 de la machine 147.210.74.32"

Difficultés d'expression

Politique

La machine 147.210.94.200 est

- serveur de courrier entrant
- relais de courrier sortant

Règles de filtrage

- SMTP-ENTRANT-1 : Autoriser tous les paquets TCP entrants, destination IP 147.210.94.200 port 25
- SMTP-ENTRANT-2 : autoriser tous les paquets TCP provenant du port 25 de 147.210.94.200.
- SMTP-SORTANT-1 : autoriser les paquets émis depuis 147.210.94.200 vers port 25 autre machine
- SMTP-SORTANT-2 : autoriser les paquets émis depuis le port 25 d'une machine vers 147.210.94.200 et ayant le bit ACK à 1

Problèmes de sécurité

Remarques

- La dernière règle laisse passer des paquets "bidouillés"
- Ne protège pas des attaques par saturation (DOS, déni de service)
- on préfère utiliser un routeur filtrant **avec suivi de connexion**

Routeur "stateful"

- Accepte certaines nouvelles connexions
- Garde une trace des connexions TCP déjà établies
- Refuse les autres paquets qui n'en font pas partie

Suivi de connexion

Suivi de connexion (routeur *stateful*).

1. accepter les paquets qui font partie d'une connexion déjà établie
2. Accepter les débuts de connexion (ACK=0)
   - vers port 25 du serveur de courrier
   - depuis serveur de courrier vers port 25 autres machines

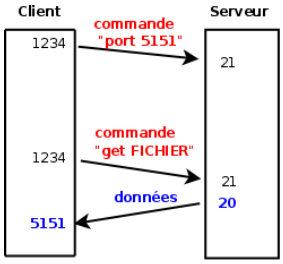
Filtrage applicatif

Pour certains protocoles, le routeur doit "comprendre" le contenu des paquets (et le protocole utilisé) pour ouvrir les connexions nécessaires.
On parle de "filtrage applicatif".

Exemple : FTP mode actif

Fonctionnement

- Le client ouvre une connexion vers le port 21 (commandes) du serveur
- Il lui indique le numéro d'un port à utiliser "PORT 5151"
- Quand le client demande un transfert, le serveur ouvre une connexion TCP depuis son port 20 (données) vers le port 5151 du client



Exemple : filtrage de FTP

Pour laisser passer FTP (mode actif), il faut

- soit autoriser toute connexion venant du port 20 des machines extérieures
- soit examiner les commandes FTP qui passent, pour n'ouvrir les ports qu'à bon escient.

Nécessité de modules de filtrage spécifiques pour les divers protocoles "à problème" (FTP, mais aussi visio-conférence...)

Logiciel de filtrage : iptables

En théorie

Le filtrage est simple

En pratique

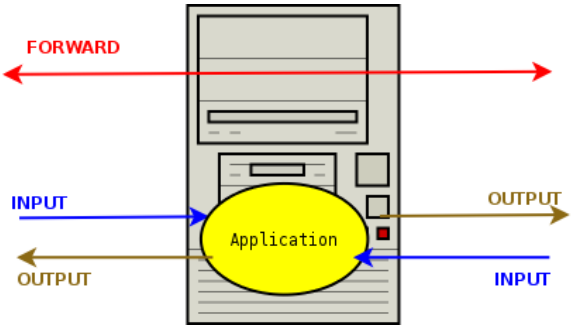
c'est plus complexe

- Beaucoup de protocoles à gérer (ssh, telnet, dns, smtp, pop3, imap, pop3s, imaps, ftp, ntp, http, Xwindow, ldap, ...)
- Des **cas particuliers** : chaque serveur a un rôle différent,
- Droits différents pour des sous-réseaux extérieurs (IUT, U-BORDEAUX, réseau gestion ....)

Organisation en "chaînes" de règles

- Les paquets sont examinés par des règles
- Les règles sont regroupées en chaînes (suites de règles + politique par défaut)
- Quelques chaines prédéfinies :
  - INPUT paquets destinés au routeur
  - OUTPUT : paquets émis par le routeur
  - FORWARD : paquets transitant par le routeur (adresses IP source et destination distinctes de celles du routeur).

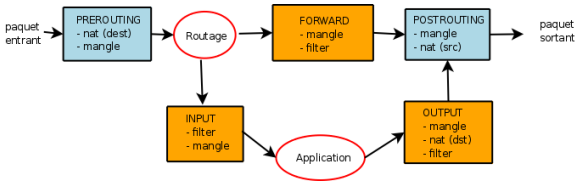
Les chaînes INPUT, OUTPUT, FORWARD



Organisation en "tables"

- Les tables regroupent des chaînes
- Les chaînes INPUT, OUTPUT et FORWARD font partie de la table par défaut *filter*
- table nat pour la translation d'adresse. Chaînes
  - PREROUTING
  - POSTROUTING
- autre table : mangle pour la modification des paquets (notamment qualité de service)

Organisation des tables



On dispose de commandes pour

- Ajouter des règles avec des actions (ACCEPT, DROP ...)
- Définir des politiques par défaut
- Créer de nouvelles chaînes pour faciliter l'organisation

Poste de travail qui refuse de servir de relais

```
iptables -A  INPUT  -j ACCEPT
iptables -A OUTPUT  -j ACCEPT
iptables -A FORWARD -j DROP
```

Exemple 2

Poste de travail connecté à internet.

- Ouvre des connexions TCP vers l'extérieur,
- mais n'en accepte pas.

```
iptables -A INPUT -p tcp -m state  \
                 --state ESTABLISHED,RELATED \
                 -j ACCEPT

iptables -A INPUT -j DROP

iptables -A OUTPUT-j ACCEPT
```

Ajout de chaînes

Pour simplifier l'organisation,

- on peut définir de nouvelles chaînes

```
iptables -N entrant
iptables -N sortant

iptables -A FORWARD -i eth0 -j entrant
iptables -A FORWARD -i eth1 -j sortant
```

- et les utiliser

```
iptables -A entrant ...
```

Intérêt des chaînes

Exemple : DMZ avec routeur à 3 pattes

- 3 interfaces eth0, eth1 eth2 pour les 3 réseaux intérieur, extérieur et dmz :
- Organisation naturelle en 6 chaînes de filtrage, selon provenance et destination.
- Attention à l'organisation : les chaînes sont destinés à être maintenus.

Mandataires/ Proxies

Certains services fonctionnent naturellement sans besoin de mandataires spécifiques, exemple :

- transmission de courrier par SMTP (principe de "Store and forward")
- serveurs de noms

D'autres se "chaînent" entre eux facilement

- exemple : proxy/cache http
- parfois : protocole spécifique de communication entre le client et le mandataire (ftp, telnet, visioconférence...)