

# *Administration réseau*

Architecture réseau  
et  
Sécurité



# *Pourquoi la sécurité ?*

- Maladroits, pirates, plaisantins et autres malveillants
- Protéger ce qu'on a à protéger
- Continuer à fonctionner
- Responsabilité morale et légale



# *La sécurité n'est pas (seulement) un problème technique*

- Les mesures de sécurité amènent des contraintes
- Gêne trop grande pour les utilisateurs => stratégies de contournement
- Nécessité
  - de sensibiliser les utilisateurs
  - d'établir une *politique de sécurité*



# *Les éléments d'une politique*

- Ce qui est obligatoire
  - Changer de mot de passe, le garder secret
- Ce qui est interdit
  - Ce qui est illégal (évidemment !)
  - WEB pendant les heures de travail, ....



# *Les éléments d'une politique (suite)*

- Ce qui est autorisé
- Ce qui est conseillé
- ...



# *Facteurs de succès d'une politique de sécurité*

- Nécessité d'une politique *officielle*, correspondant à la *réalité*
- Soutien de la hiérarchie
  - Avec des moyens correspondant aux besoins
- Adhésion des utilisateurs



# *Sécurité réseau*

Les éléments techniques  
d'une architecture réseau



# *Éléments techniques*

- Réseaux interne / externe
  - Adresses privées / publiques
  - Routeurs, passerelles, filtrage...
  - Serveurs, Postes-clients, « Bastions »,...
  - Proxy (mandataire, relais)
  - Tunnels, réseaux privés virtuels (VPN)
  - ...
- 
-



# *Approche fonctionnelle*

- Un réseau, avec des machines à intérieur, relié à l'extérieur
- Permettre l'accès
  - De **l'intérieur**, à certains services extérieurs
  - De **l'extérieur**, à certains services hébergés sur des serveurs internes
- Protection des machines internes



## *Approche fonctionnelle (suite)*

- **Bastions** : les serveurs « exposés » qui doivent être surveillés particulièrement
- En général, on limite l'accès aux services extérieurs (P2P...)
- Rappel : responsabilité des employeurs



# *Isolation du réseau interne*

- Se fait par
  - Un routeur filtrant (boîtier spécialisé avec 2 interfaces réseau),
  - Un « hôte à double réseau » (ordinateur avec 2 interfaces réseau)

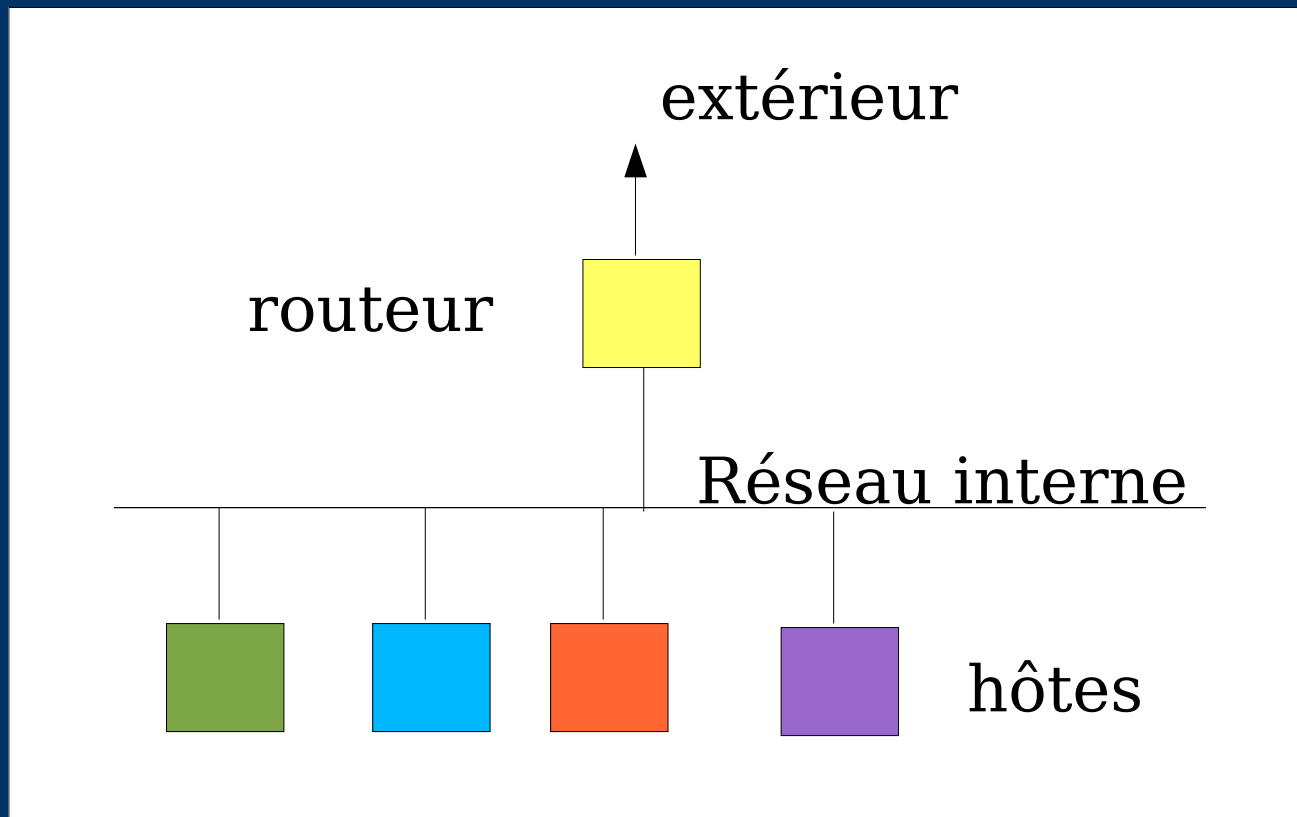


# *Isolation du réseau interne*

- La différence est ténue
  - Un hôte à double réseau peut se comporter en routeur filtrant + possibilités d'y héberger des services (par exemple mandataires)
  - Un routeur filtrant est souvent un ordinateur « fermé »



# *Le routeur sépare et relie*



# *Exemple d'architecture*

- Routeur non filtrant (passerelle ou simple switch !)
- On donne des adresses IP *privées* aux machines du réseau
  - Exemple : 10.1.1.1, 10.1.1.2

## Exemple d'architecture (suite)

- Les serveurs ont aussi une adresse IP publique
- Moyen: utilisation d'alias pour les cartes réseau

```
ifconfig eth0 10.1.1.4
```

```
ifconfig eth0:0 147.210.94.200
```

# *Communication entre clients et extérieur ?*

- Les clients ne peuvent pas dialoguer directement avec l'extérieur
  - Passage par des **mandataires** internes
  - Ok pour certains services
    - smtp, nntp, web
    - ftp
  - Plus compliqué ou impossible pour d'autres (sessions telnet)
- 
-



## « *masquerading* » (NAT)

- Idée : le client fait passer ses communications par le routeur
  - Le routeur « déguise » les paquets pour faire croire qu'il en est l'émetteur
  - Le serveur distant répond au routeur
  - Le routeur fait suivre les réponses au client
  - C'est un exemple de NAT (Network Address Translation)
- 
-

## *Exemple :*

- le poste 10.1.2.3 démarre une session telnet (TCP, port 23) en direction de 220.6.7.8
  - Le routeur remplace l'adresse d'origine (10.1.2.3) par sa propre adresse, et fait suivre à l'extérieur
  - Le site extérieur répond au routeur
  - Le routeur remplace l'adresse de destination (la sienne) par celle du demandeur 10.1.2.3
  - Le demandeur a obtenu sa réponse !
- 
-

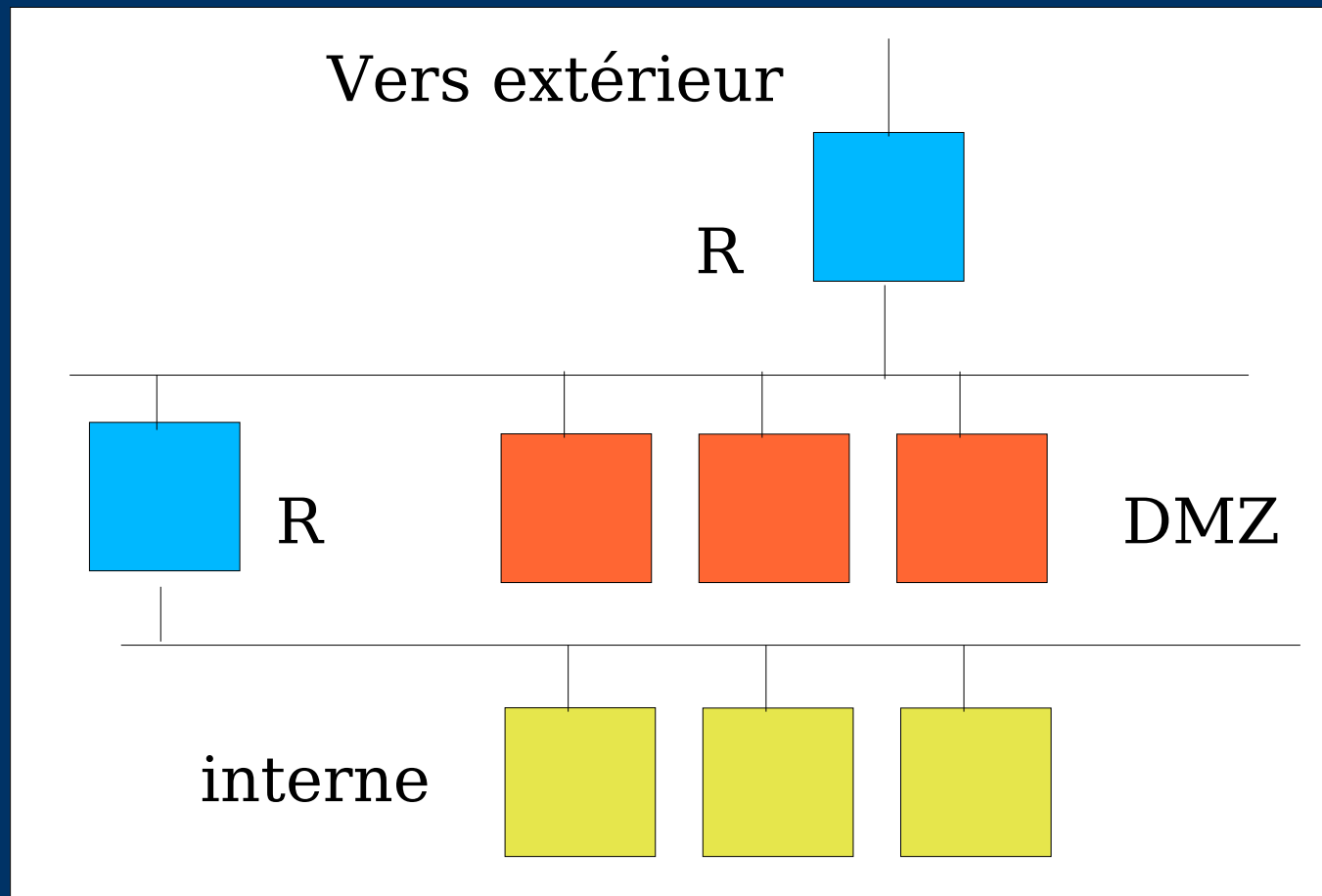
# *Routage*

- Poste client ou serveurs
    - Route locale vers autres clients
    - Route locale vers serveurs internes
    - Route par défaut : routeur de sortie
  - Routeur
    - Route locale vers clients et machines internes
    - Route locale vers réseau extérieur
    - Route par défaut : passerelle de sortie
- 
-

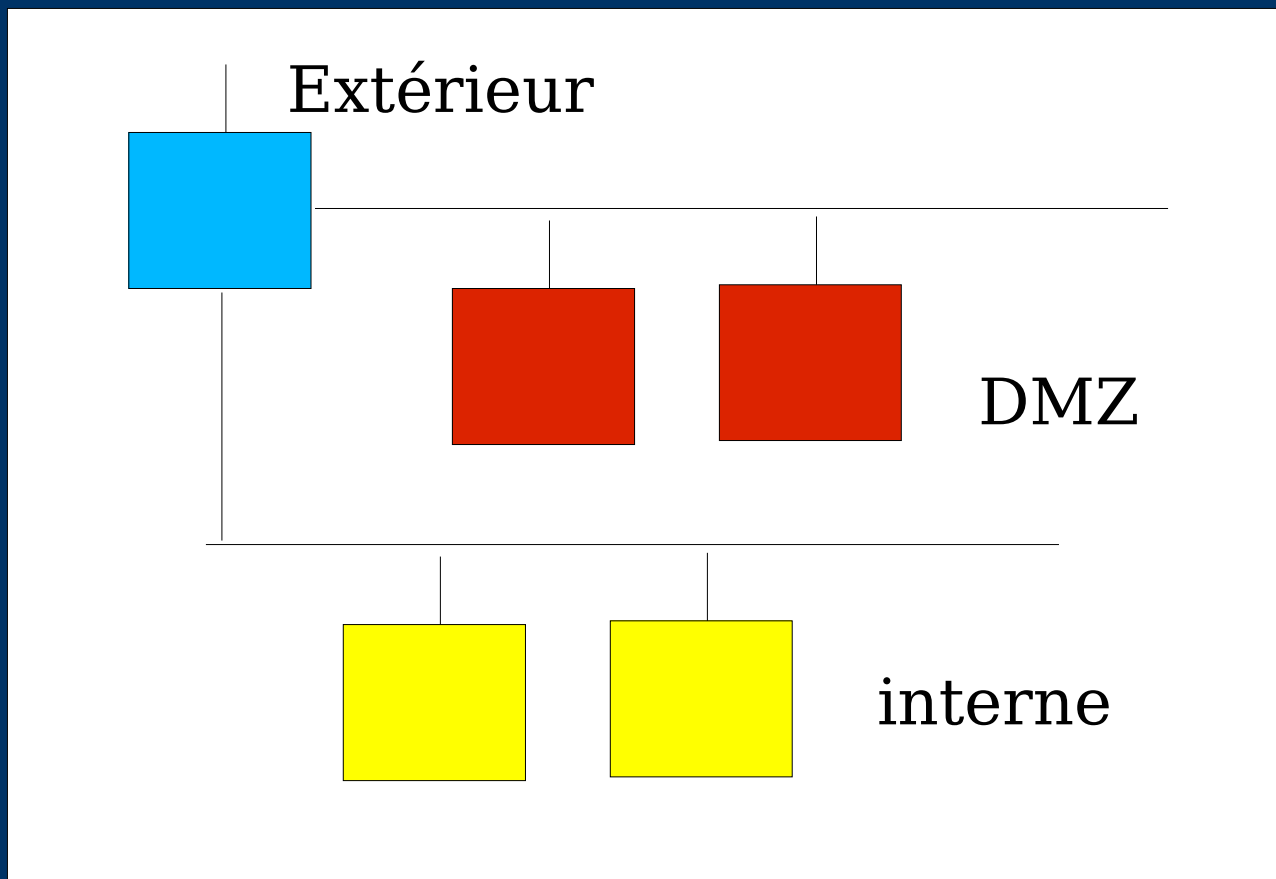
# Architecture à « DMZ »

- Découpage du réseau interne en 2 zones isolées
  - Serveurs accessibles de l'extérieur, situés en « zone démilitarisée »
  - Postes clients inaccessibles de l'extérieur
- Variantes principales
  - Utilisation de deux routeurs
  - Utilisation d'un routeur « à 3 pattes »

# Architecture DMZ à deux routeurs



# *DMZ, routeur à 3 pattes*



# *Avantages-Inconvénients*

- 3 pattes : plus économique, 1 seule machine à gérer
- 2 routeurs :
  - Configuration plus simple
  - Meilleure sécurité (2 étapes successives pour « braquer » le réseau interne)



# *Le filtrage*

- Basé essentiellement sur l'examen des entêtes des paquets IP
    - Protocole (ICMP, TCP, UDP, ...)
    - Interface de provenance
    - Adresse de provenance et de destination, Ports
    - Flags « ACK », « SYN », « FIN », etc.
- 
-



# *Règles de filtrage*

- Ecriture de règles de filtrage du type
  - « accepter les paquets TCP entrants à destination du port 25 de la machine 147.210.74.32 »

# *Filtrage « sans états »*

- La politique est difficile à exprimer par des règles de ce type.
- Exemple : Décision d'utiliser la machine 147.210.94.200 comme
  - serveur de courrier entrant
  - relais de courrier sortant

# *Règles pour filtrage « stateless »*

- 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.

# Règles pour filtrage « stateless »

- 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**

# Remarque

- La dernière règle n'empêche pas le passage de paquets « bidouillés » en direction du serveur de courrier
- Laisse passer les tentatives d'attaques par saturation (DOS, déni de service)



## *Remarque (suite)*

- Solution : routeur filtrant avec *suivi de connexion (TCP)*
  - Le routeur mémorise les connexions TCP établies
  - Refuse les paquets qui n'en font pas partie, ou ne sont pas le début d'une nouvelle connexion autorisée

## *Routeur filtrant « stateful »*

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



# *Examen du contenu*

- Pour certains protocoles, le routeur doit « comprendre » le contenu des paquets (et le protocole utilisé) pour ouvrir les connexions nécessaires.
- Exemple : FTP mode actif





## *Exemple : ftp mode actif*

- Fonctionnement en mode actif
  - 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

# FTP et filtrage

- 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 visioconférence... )
- 
-

# Logiciel de filtrage : iptables

- Le filtrage est simple *en théorie*
  - En pratique, grand nombre de règles
    - Beaucoup de protocoles à gérer (ssh, telnet, dns, smtp, pop3, imap, pop3s, imaps, ftp, ntp, http, proxy-http, Xwindow, ldap, ...)
    - Cas particuliers : chaque serveur a un rôle différent,
    - Droits différents pour des sous-réseaux extérieurs (IUT, U-BORDEAUX, réseau gestion ....)
- 
-

## *Les « tables »*

- Les paquets IP sont regardés par des listes des règles appelées tables
  - Tables 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).
- 
-

# *Iptables*

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



# Exemple

- 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. Il 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
```

---

---

## *tables utilisateur*

- Pour simplifier l'organisation, on peut définir de nouvelles tables (groupes de règles).

Exemple :

```
iptables -N entrant
```

```
iptables -N sortant
```

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

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



## *(intérêt des tables)*

- Exemple : DMZ avec routeur à 3 pattes
  - 3 interfaces eth0, eth1 eth2 pour les 3 réseaux intérieur, extérieur et dmz :
  - Organisation naturelle en 6 groupes de règles de filtrage, selon provenance et destination.
  - Attention à l'organisation : les ensembles de règles sont destinés à être maintenus.
  - Besoin de « traçabilité » des décisions !
- 
-

# *Conclusion*

- La sécurité réseau a des aspects techniques, mais
- La clé du succès est dans
  - La définition d'une politique acceptée (compromis entre besoins des utilisateurs et moyens qu'il est possible de mettre en oeuvre)
  - L'organisation

