

Sécurité des réseaux

Firewalls

A. Guermouche

1. Firewall?
2. DMZ
3. Proxy
4. Logiciels de filtrage de paquets
5. Ipfwadm
6. Ipchains
7. Iptables
8. Iptables et filtrage
9. Translation d'adresses
 - NAT statique
 - NAT dynamique : Masquerading

Plan

1. Firewall?
2. DMZ
3. Proxy
4. Logiciels de filtrage de paquets
5. Ipfwadm
6. Ipchains
7. Iptables
8. Iptables et filtrage
9. Translation d'adresses
 - NAT statique
 - NAT dynamique : Masquerading

Pourquoi un firewall?

Definition

Programme, ou un matériel, chargé de vous protéger du monde extérieur en contrôlant tout ce qui passe, et surtout tout ce qui ne doit pas passer entre internet et le réseau local.

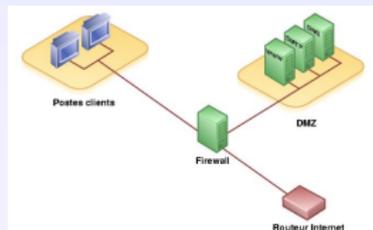
pourquoi un firewall?

Contrôle. Gérer les connexions sortantes a partir du réseau local.

Sécurité. Protéger le réseau interne des intrusions venant de l'extérieur.

Vigilance. Surveiller/tracer le trafic entre le réseau local et internet.

Firewall



Plusieurs types de firewalls :

- ★ Pare-feu au niveau réseau
- ★ Pare-feu au niveau applicatif
- ★ Pare-feu des applications

Différents types de firewalls

Pare-feu niveau réseau. (iptables, paquet filter, ...)

- ★ Firewall fonctionnant à un niveau bas de la pile TCP/IP
- ★ Basé sur le filtrage des paquets
- ★ Possibilité (si mécanisme disponible) de filtrer les paquets suivant l'état de la connexion

Intérêt : Transparence pour les utilisateurs du réseau

Pare-feu au niveau applicatif. (inetd, xinetd, ...)

- ★ Firewall fonctionnant au niveau le plus haut de la pile TCP/IP
- ★ Généralement basé sur des mécanisme de proxy

Intérêt : Possibilité d'interpréter le contenu du trafic

Pare-feu des applications. (/etc/ftpaccess pour ftp, ...)

- ★ Restrictions au niveau des différentes applications

Plan

1. Firewall?
- 2. DMZ**
3. Proxy
4. Logiciels de filtrage de paquets
5. Ipfwadm
6. Ipchains
7. Iptables
8. Iptables et filtrage
9. Translation d'adresses
 - NAT statique
 - NAT dynamique : Masquerading

Definition (DMZ)

Une zone démilitarisée (DMZ) est un sous-réseau se trouvant entre le réseau local et le réseau extérieur.

Propriétés :

- ★ Les connexions à la DMZ sont autorisées de n'importe où.
- ★ Les connexions à partir de la DMZ ne sont autorisées que vers l'extérieur.

Intérêt :

- ★ Rendre des machines accessible à partir de l'extérieur (possibilité de mettre en place des serveurs (DNS, SMTP, ...)).

Plan

1. Firewall?
2. DMZ
- 3. Proxy**
4. Logiciels de filtrage de paquets
5. Ipfwadm
6. Ipchains
7. Iptables
8. Iptables et filtrage
9. Translation d'adresses
 - NAT statique
 - NAT dynamique : Masquerading

Proxy ou mandataire

Définition :

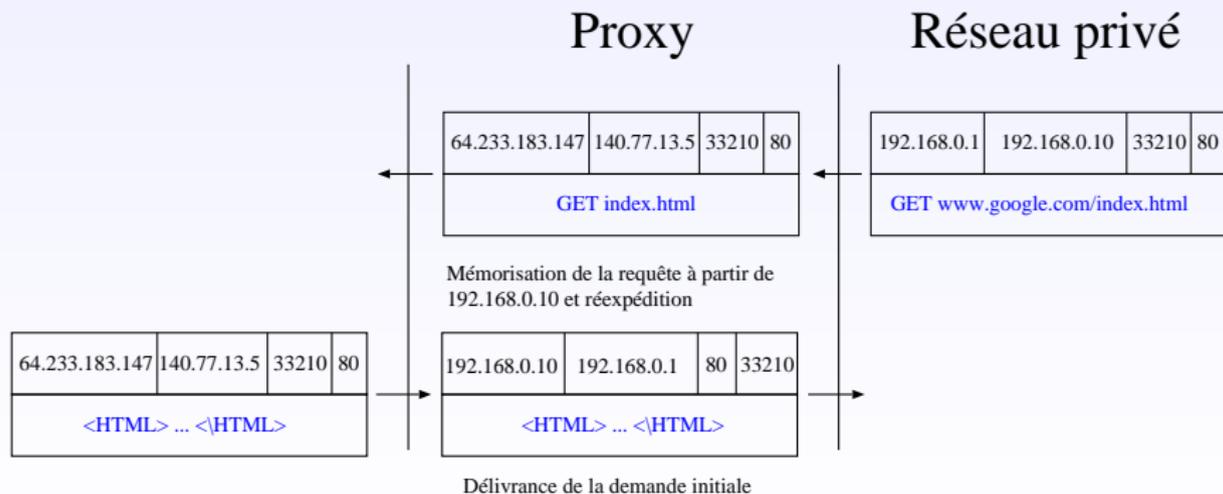
- ★ Un proxy est un intermédiaire dans une connexion entre le client et le serveur
- ★ Le client s'adresse toujours au proxy
- ★ Le proxy est spécifique à une application donnée (HTTP, FTP, ...)

→ Possibilité de modification des informations échangées entre le client et le serveur.

Proxy ou mandataire

Définition :

- ★ Un proxy est un intermédiaire dans une connexion entre le client et le serveur
- ★ Le client s'adresse toujours au proxy
- ★ Le proxy est spécifique à une application donnée (HTTP, FTP, ...)



Plan

1. Firewall?
2. DMZ
3. Proxy
4. Logiciels de filtrage de paquets
5. Ipfwadm
6. Ipchains
7. Iptables
8. Iptables et filtrage
9. Translation d'adresses
 - NAT statique
 - NAT dynamique : Masquerading

Logiciels de filtrage de paquets

- ★ Fonctionnalités de “firewall” filtrant directement implémentée dans le noyau Linux.
- ★ Filtrage de niveau 3 ou 4.
- ★ 3 types de firewall filtrants :
 - ipfwadm.** Jusqu'à la version 2.1.102 du noyau linux
 - ipchains.** Entre les versions 2.2.0 et 2.4 du noyau linux
 - iptables.** À partir des noyaux 2.4

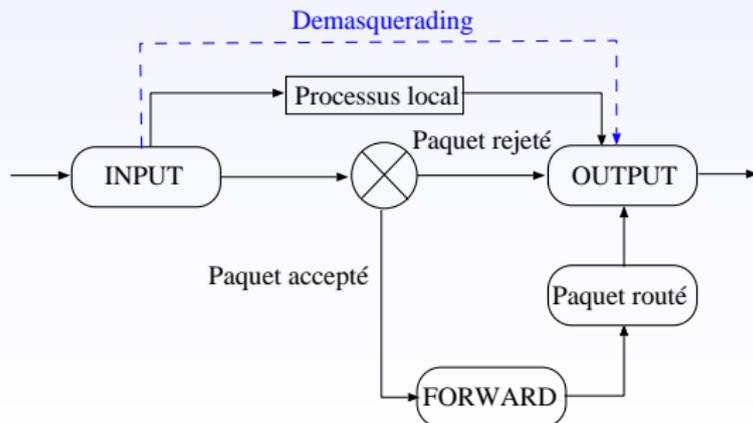
Plan

1. Firewall?
2. DMZ
3. Proxy
4. Logiciels de filtrage de paquets
- 5. Ipfwadm**
6. Ipchains
7. Iptables
8. Iptables et filtrage
9. Translation d'adresses
 - NAT statique
 - NAT dynamique : Masquerading

Ipfwadm

- ★ Firewall permettant la gestion des paquets TCP, UDP et ICMP.
- ★ 3 types de règles :
 - INPUT.** sont appliquées lors de l'arrivée d'un paquet.
 - FORWARD.** sont appliquées lorsque la destination du paquet n'est pas le routeur.
 - OUTPUT.** sont appliquées dès qu'un paquet doit sortir du routeur.

Fonctionnement :



Ipfwadm

- ★ Firewall permettant la gestion des paquets TCP, UDP et ICMP.
- ★ 3 types de règles :
 - INPUT.** sont appliquées lors de l'arrivée d'un paquet.
 - FORWARD.** sont appliquées lorsque la destination du paquet n'est pas le routeur.
 - OUTPUT.** sont appliquées dès qu'un paquet doit sortir du routeur.

Fonctionnement :

- 1: lorsqu'un paquet entre, il traverse les règles de type INPUT
- 2: **Si** il est accepté **Alors**
- 3: **Si** il est destiné à une autre machine **Alors**
- 4: il est routé vers les règles FORWARD
- 5: **Sinon**
- 6: il est rejeté
- 7: le paquet est finalement émis

Dans tous les cas, le paquet traverse les règles OUTPUT

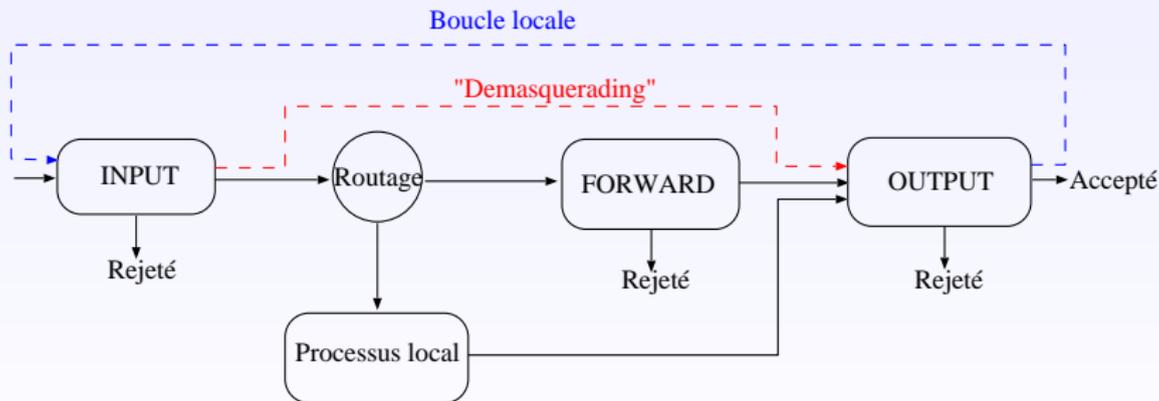
Plan

1. Firewall?
2. DMZ
3. Proxy
4. Logiciels de filtrage de paquets
5. Ipfwadm
- 6. Ipchains**
7. Iptables
8. Iptables et filtrage
9. Translation d'adresses
 - NAT statique
 - NAT dynamique : Masquerading

Ipchains

- ★ Module du noyau Linux réalisant le filtrage de paquets.
- ★ Inspiré du pare-feu BSD (tout comme ipfwadm)

Fonctionnement :



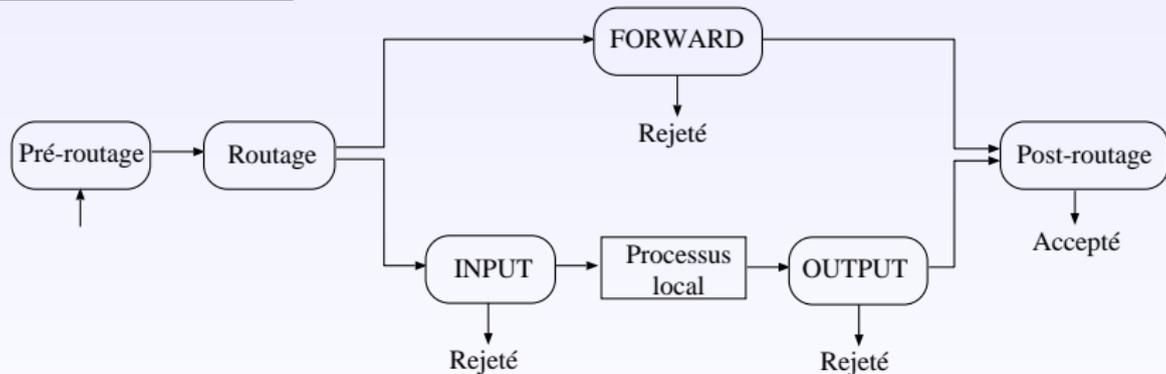
Plan

1. Firewall?
2. DMZ
3. Proxy
4. Logiciels de filtrage de paquets
5. Ipfwadm
6. Ipchains
- 7. Iptables**
8. Iptables et filtrage
9. Translation d'adresses
 - NAT statique
 - NAT dynamique : Masquerading

Iptables (1/2)

- ★ Module du noyau Linux réalisant le filtrage de paquets (noyaux ≥ 2.4).
- ★ Améliorations en matière de filtrage et de translation d'adresses par rapport à Ipchains.

Fonctionnement :



Iptables (1/2)

- ★ Module du noyau Linux réalisant le filtrage de paquets (noyaux ≥ 2.4).
- ★ Améliorations en matière de filtrage et de translation d'adresses par rapport à Ipchains.

Fonctionnement :

À l'arrivée d'un paquet (après décision de routage) :

- 1: **Si** le paquet est destiné à l'hôte local **Alors**
- 2: il traverse la chaîne INPUT.
- 3: **Si** il n'est pas rejeté **Alors**
- 4: il est transmis au processus impliqué.
- 5: **Sinon**
- 6: **Si** le paquet est destiné à un hôte d'un autre réseau **Alors**
- 7: il traverse la chaîne FORWARD
- 8: **Si** il n'est pas rejeté **Alors**
- 9: il poursuit alors sa route

Iptables (1/2)

- ★ Module du noyau Linux réalisant le filtrage de paquets (noyaux ≥ 2.4).
- ★ Améliorations en matière de filtrage et de translation d'adresses par rapport à Ipchains.

Fonctionnement :

À l'arrivée d'un paquet (après décision de routage) :

- 1: **Si** le paquet est destiné à l'hôte local **Alors**
- 2: il traverse la chaîne INPUT.
- 3: **Si** il n'est pas rejeté **Alors**
- 4: il est transmis au processus impliqué.
- 5: **Sinon**
- 6: **Si** le paquet est destiné à un hôte d'un autre réseau **Alors**
- 7: il traverse la chaîne FORWARD
- 8: **Si** il n'est pas rejeté **Alors**
- 9: il poursuit alors sa route

Tous les paquets émis par des processus locaux au routeur traversent la chaîne OUTPUT.

Iptables (2/2)

Fonctionnalités :

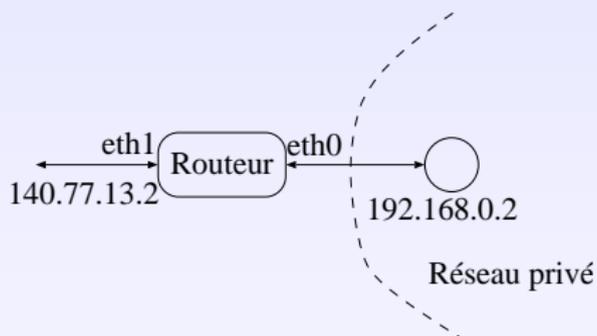
- ★ Filtrage de paquets
- ★ NAT
- ★ Marquage de paquets

Architectures : Trois tables de chaînes (FILTER, NAT et MANGLE).

FILTER (filtrage des paquets)		NAT (translation d'adresses)	
INPUT	paquet entrant sur le routeur	PREROUTING	NAT de destination
OUTPUT	paquet émis par le routeur	POSTROUTING	NAT de source
FORWARD	paquet traversant le routeur	OUTPUT	NAT sur les paquets émis localement

La table **MANGLE** sert au marquage des paquets

Fonctionnalités NAT d'Iptables (1/2)



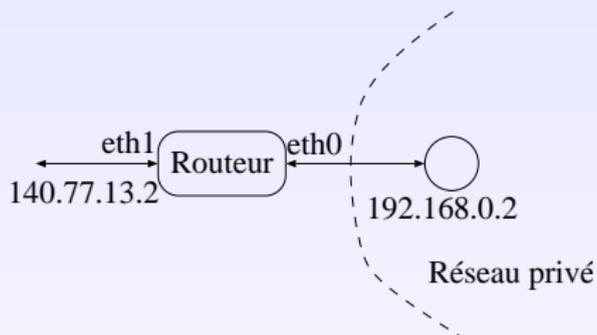
Modification de la destination du paquet avant le routage (paquet reçu de l'extérieur).

```
iptables -t nat -A PREROUTING -d 140.77.13.2 -i eth1 -j DNAT
-to-destination 192.168.0.2
```

Modification de la source du paquet après le routage (paquet émis à partir du réseau privé).

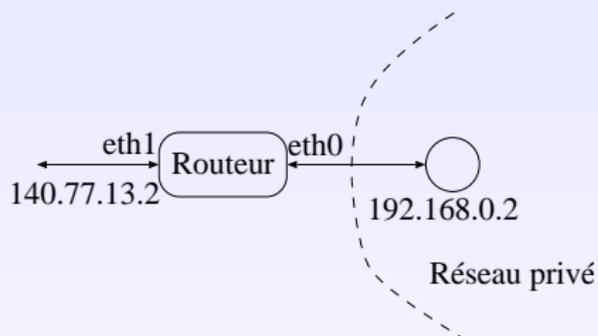
```
iptables -t nat -A POSTROUTING -s 192.168.0.2 -o eth1 -j
SNAT -to-source 140.77.13.2
```

Fonctionnalités NAT d'Iptables (1/2)



Exercice : Comment faire pour que le routeur puisse envoyer un paquet à l'adresse 140.77.13.2?

Fonctionnalités NAT d'Iptables (1/2)



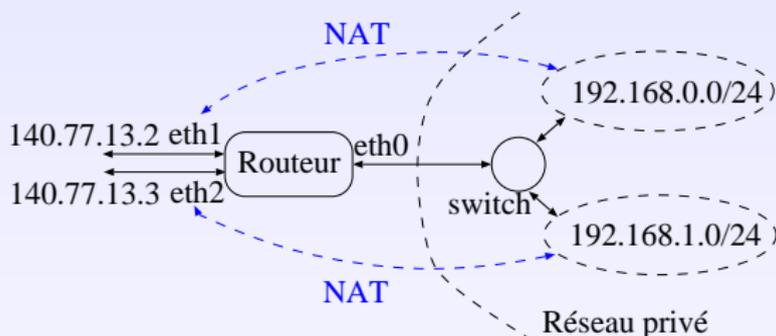
Exercice : Comment faire pour que le routeur puisse envoyer un paquet à l'adresse 140.77.13.2?

Réponse :

Il faut modifier la destination du paquet émis localement avant le routage.

```
iptables -t nat -A OUTPUT -d 140.77.13.2 -j DNAT  
-to-destination 192.168.0.2
```

Fonctionnalités NAT d'Iptables (2/2)



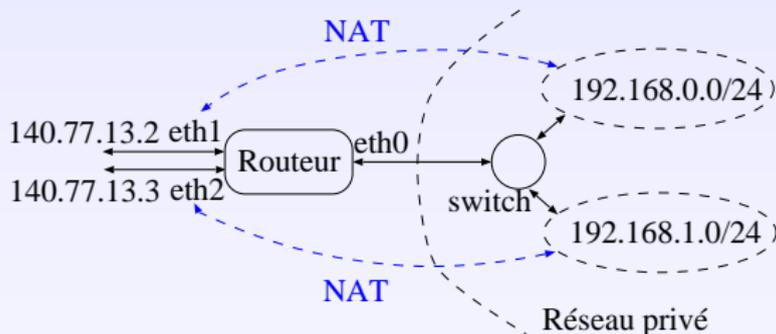
Association entre toutes les adresses privées du sous-réseau 192.168.0.0/24 avec l'interface eth1.

```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.0/24 -j MASQUERADE
```

Association entre toutes les adresses privées du sous-réseau 192.168.1.0/24 avec l'interface eth2.

```
iptables -t nat -A POSTROUTING -o eth2 -s 192.168.1.0/24 -j MASQUERADE
```

Transfert de ports



Transférer les connexions sur le port 80 de l'adresse 140.77.13.2 sur la machine ayant l'adresse privée 192.168.0.200 sur le port 8080 :

```
iptables -t nat -A PREROUTING -p tcp -i eth0 -d 140.77.13.2
-dport 80 -sport 1024:65535 -j DNAT -to 192.168.0.200:8080
```

Plan

1. Firewall?
2. DMZ
3. Proxy
4. Logiciels de filtrage de paquets
5. Ipfwadm
6. Ipchains
7. Iptables
- 8. Iptables et filtrage**
9. Translation d'adresses
 - NAT statique
 - NAT dynamique : Masquerading

Iptables et filtrage(1/2)

- ★ Filtrage des paquets IP, TCP, UDP ou ICMP
- ★ Spécification de règle pour le rejet ou l'acceptation de paquet
- ★ Utilisation de la table FILTER et des chaînes INPUT, OUTPUT et FORWARD
- ★ Règles traitées de manière séquentielle : Le paquet sort dès qu'il rencontre une règle qui peut lui être appliquée

Exemples :

- ★ Accepter tous les paquets en provenance de n'importe où et destinés à l'adresse du routeur 192.168.1.1.

```
iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1 -p TCP
-j ACCEPT
```

- ★ Accepter de router les paquets entrant sur eth0 tels que :

@source	@dest	P-source	P-dest
0/0	192.168.1.58	1024-65535	80

```
iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58 -o
eth1 -p TCP -sport 1024:65535 -dport 80 -j ACCEPT
```

Iptables et filtrage(2/2)

- ★ Accepter un paquet ICMP “echo-request” (ping) par seconde

```
iptables -A INPUT -p icmp -icmp-type echo-request -m
limit -limit 1/s -i eth0 -j ACCEPT
```

- ★ Accepter 5 segments TCP ayant le bit SYN positionné par seconde (permet d'éviter de se faire inonder)

```
iptables -A INPUT -p tcp -syn -m limit -limit 5/s -i
eth0 -j ACCEPT
```

- ★ Accepter de router les paquets entrants sur eth0 tels que :

@source	@dest	P-source	P-dest
0/0	192.168.1.58	1024-65535	80 ou 443

```
iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58 -o
eth1 -p TCP -sport 1024:65535 -m multiport -dport 80,443
-j ACCEPT
```

Iptables et suivi des connexions

- ★ Suivi des connexions disponible (*conntrack*)
- ★ Quatre états possibles pour une connexion :
 - NEW** . Nouvelle connexion établie
 - ESTABLISHED** . La connexion analysée est déjà établie
 - RELATED** . La connexion est en relation avec une connexion déjà établie (ftp-data par exemple)
 - INVALID** . Le paquet reçu n'appartient à aucune des trois catégories précédentes.

Exemples :

- ★ Autoriser tous les paquets émis par le routeur concernant des connexions déjà établies.

```
iptables -A OUTPUT -o eth0 -m state --state  
ESTABLISHED,RELATED -j ACCEPT
```

Iptables et suivi des connexions

- ★ Suivi des connexions disponible (*conntrack*)
- ★ Quatre états possibles pour une connexion :
 - NEW** . Nouvelle connexion établie
 - ESTABLISHED** . La connexion analysée est déjà établie
 - RELATED** . La connexion est en relation avec une connexion déjà établie (ftp-data par exemple)
 - INVALID** . Le paquet reçu n'appartient à aucune des trois catégories précédentes.

Exemples :

- ★ Autoriser le routeur à relayer tous les paquets reçus concernant de nouvelles connexions sur le port 22.

```
iptables -A FORWARD -p tcp -i eth0 -dport 22 -sport  
1024:65535 -m state -state NEW -j ACCEPT
```

Outils de diagnostic

Traces iptables. Possibilité de tracer certaines actions iptables.

exemple :

1. Tracer toutes les actions iptables :

```
iptables -A OUTPUT -j LOG
```

```
iptables -A INPUT -j LOG
```

```
iptables -A FORWARD -j LOG
```

2. Rajouter une règle pour tracer les paquets rejetés

```
iptables -N LOG_DROP
```

```
iptables -A LOG_DROP -j LOG -log-prefix
```

```
' [IPTABLES DROP] : '
```

```
iptables -A LOG_DROP -j DROP
```

nmap, nessus, ... Logiciels permettant de diagnostiquer l'état d'un firewall (trouver les ports ouverts, détecter les services utilisant les ports, ...)

Plan

1. Firewall?
2. DMZ
3. Proxy
4. Logiciels de filtrage de paquets
5. Ipfwadm
6. Ipchains
7. Iptables
8. Iptables et filtrage
9. Translation d'adresses
 - NAT statique
 - NAT dynamique : Masquerading

Pourquoi avoir des adresses privées?

- ★ Gérer la pénurie d'adresses au sein d'un réseau
- ★ Masquer l'intérieur du réseau par rapport à l'extérieur (le réseau peut être vu comme une seule et même machine)
- ★ Améliorer la sécurité pour le réseau interne
- ★ Assouplir la gestion des adresses du réseau interne
- ★ Faciliter la modification de l'architecture du réseau interne

→ Mécanisme de translation d'adresses (NAT - Network Address Translation)

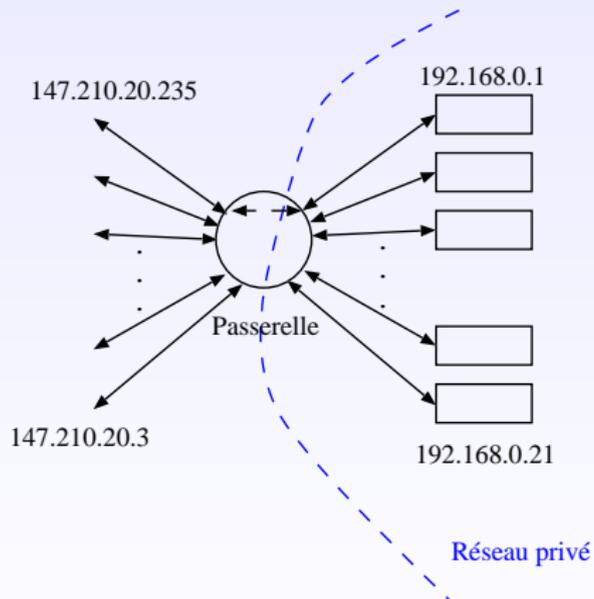
Deux types de NAT :

statique. association entre n adresses publiques et n adresses privées.

dynamique. association entre 1 adresse publique et n adresses privées.

NAT statique

Association entre **une** adresse publique et **une** adresse privée.



NAT statique

Association entre **une** adresse publique et **une** adresse privée.

Intérêt :

- ★ Uniformité de l'adressage dans la partie privée du réseau (modification de la correspondance **@publique/@privée** facile)
- ★ Sécurité accrue (tous les flux passent par la passerelle NAT)

Inconvénient :

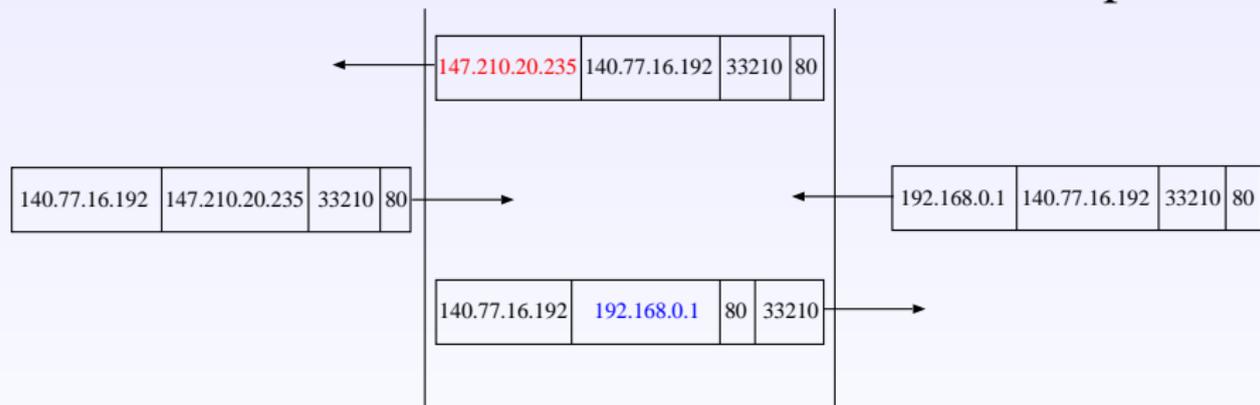
- ★ Problème de pénurie d'adresses IP publiques non-résolu

NAT statique : Principe

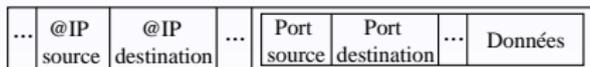
Pour chaque paquet sortant (resp. entrant), la passerelle modifie l'adresse source (resp. destination).

Passerelle

Réseau privé



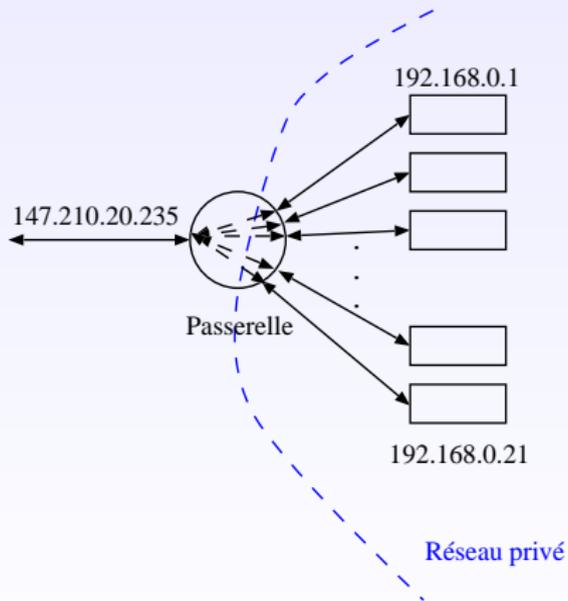
Paquet IP



Paquet TCP

NAT dynamique : Masquerading

Association entre m adresses publiques et n adresses privées
($m < n$).



NAT dynamique : Masquerading

Association entre m adresses publiques et n adresses privées
($m < n$).

Intérêt :

- ★ Plusieurs machines utilisent la même adresse IP publique pour sortir du réseau privé
- ★ Sécurité accrue (tous les flux passent par la passerelle NAT)

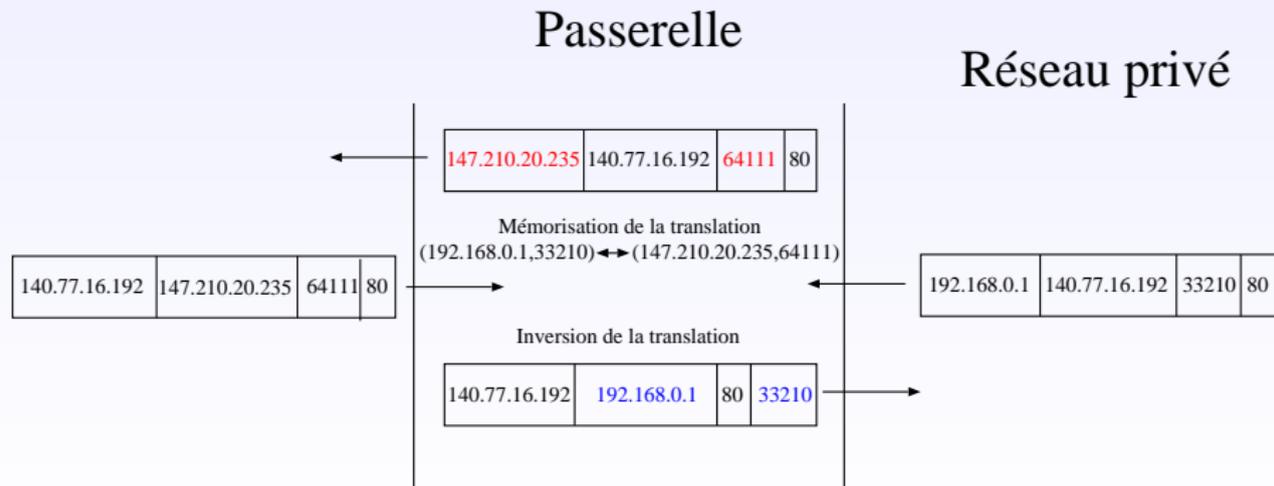
Inconvénient :

- ★ Les machines du réseau interne ne sont pas accessibles de l'extérieur (impossibilité d'initier une connexion de l'extérieur)

NAT dynamique : Principe (1/2)

L'association de n adresses privées à 1 adresse publique nécessite, au niveau de la passerelle, de :

- ★ modifier l'adresse source (resp. destination) des paquets sortant (resp. entrants)
- ★ changer le **numéro de port source** pour les flux sortant



NAT dynamique : Principe (2/2)

Comment est ce que le routeur différencie les paquets qui lui sont destinés de ceux qu'il doit relayer?

À chaque nouvelle connexion :

- 1: Modifier l'adresse source et le port source :
(@source_privée,port_source)→(@publique,port_source')
- 2: Sauvegarder l'association dans la table NAT

Pour chaque paquet entrant :

- 3: Chercher une association correspondant au couple (@destination, port_destination)
- 4: **Si** \exists une association dans la table NAT **Alors**
- 5: Modifier l'adresse de destination et le port de destination
- 6: Relayer le paquet
- 7: **Sinon**
- 8: /* Erreur de routage */

NAT dynamique : Principe (2/2)

Comment est ce que le routeur différencie les paquets qui lui sont destinés de ceux qu'il doit relayer?

À chaque nouvelle connexion :

- 1: Modifier l'adresse source et le port source :
(@source_privée,port_source)→(@publique,port_source')
- 2: Sauvegarder l'association dans la table NAT

Pour chaque paquet entrant :

- 3: Chercher une association correspondant au couple (@destination, port_destination)
- 4: **Si** \exists une association dans la table NAT **Alors**
- 5: Modifier l'adresse de destination et le port de destination
- 6: Relayer le paquet
- 7: **Sinon**
- 8: /* Erreur de routage */

Le routeur gère toutes les associations

⇒ Unicité de l'association (donc du port source après translation)

Problèmes liés à NAT dynamique

Comment faire de la translation d'adresse sur des protocoles qui ne sont pas basés sur TCP ou UDP (pas de numéro de port)?

- ★ Nécessité d'implémenter une méthode spécifique au protocole (identifiant ICMP pour ICMP par exemple).
- ★ Dans le cas des protocoles dont les paquets contiennent des données relatives aux adresses IP, il est nécessaire de mettre en place des "proxy" (FTP en mode actif par exemple).

Comment rendre joignables des machines du réseau local?

- ★ Nécessité de faire de la redirection de port (port forwarding/mapping).

Principe. Toutes les connexions entrantes sur un port donné sont redirigée vers une machine du réseau privé sur un port (qui peut être le même ou non).