# Firewall : Pare-feu

# Pare-feux (‘Firewalls’)

## *Définition*:

***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***.

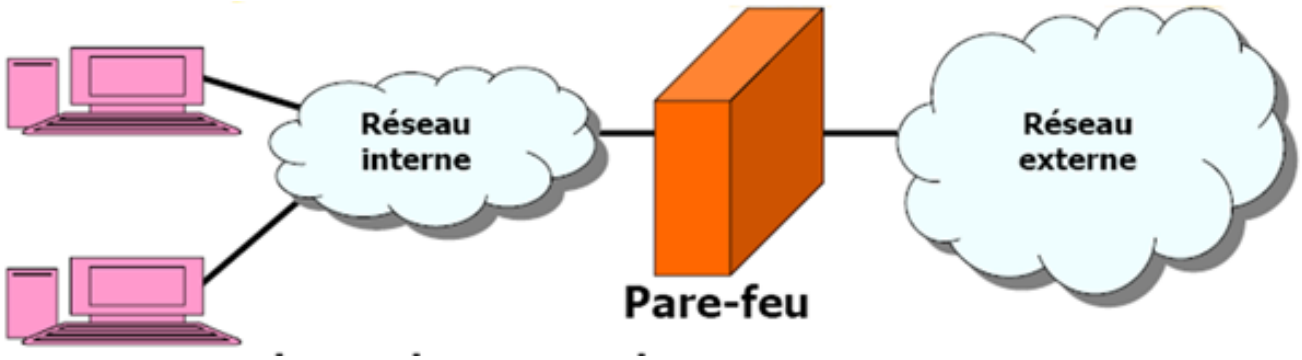
- Ensemble de composant applicant une politique de contrôle d’accès entre deux réseaux

- Le pare-feu est un système permettant de filtrer les paquets de données échangés avec le réseau, il s'agit ainsi d'une passerelle filtrante comportant au minimum les interfaces réseaux suivantes :
  - une interface pour le réseau à protéger (réseau interne) ;
  - une interface pour le réseau externe.

# Fonctions d'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.

# Pare-feux: Architecture de base




**1. Un domaine à protéger: un réseau 'interne':**

- Un réseau d'entreprise/personnel que l'on veut protéger vis à vis d'un réseau 'externe' d'ou des intrus sont susceptibles de conduire des attaques.

**2. Un pare-feu:**

- Unpoint de passage obligatoire entre le réseau à protéger (interne) et un réseau non sécuritaire (externe).
- C'est un ensemble de différents composants matériels et logiciels qui contrôlent le trafic intérieur/extérieur selon une politique de sécurité.

# Fonctionnement d'un système pare-feu

- Un système pare-feu contient un ensemble de règles prédéfinies permettant :
  - D'autoriser la connexion (***allow***) ;
  - De bloquer la connexion (***deny***) ;
  - De rejeter la demande de connexion sans avertir l'émetteur (***drop***).
  - De rejeter la demande de connexion avec avertissement l'émetteur (**reject**).

# Différents types de firewalls

- Plusieurs types de firewalls :
  - Pare-feu des applications
  - Pare-feu au niveau réseau
  - Pare-feu au niveau applicatif

# Différents types de firewalls: Pare-feu des applications

- Pare-feu des applications.
  - (/etc/mail/access pour le mail)
  - (/etc/ftpaccess pour ftp, . . . )
- Restrictions au niveau des différentes applications

# Différents types de firewalls: Pare-feu niveau réseau

- 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

# Différents types de firewalls: Pare-feu niveau applicatif

- 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

# Filtre applicatif (proxy ou mandataire)

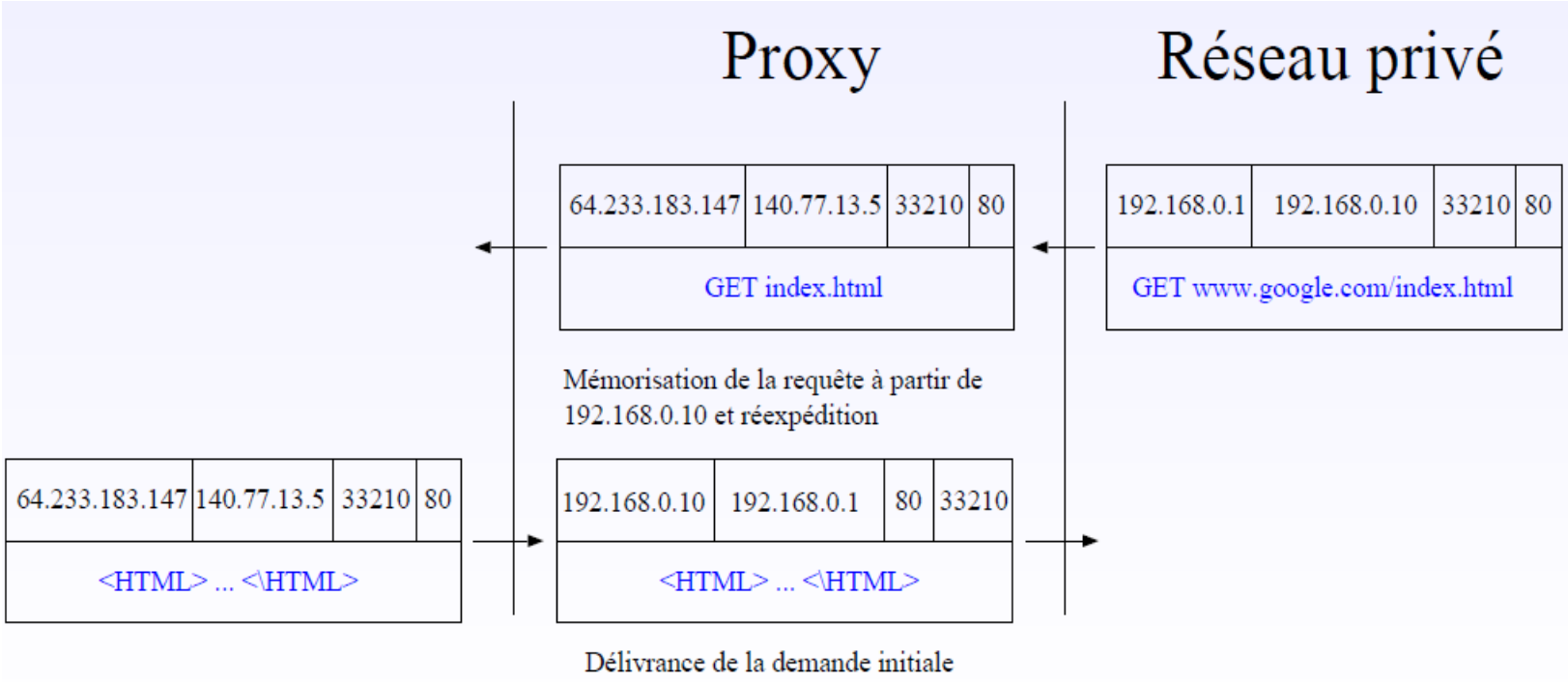
- **Définition :**
  - Un proxy est un intermédiaire dans une connexion entre le client et le serveur,
  - Le client s'adresse toujours au proxy
  - Le proxy est spécifique à une application donnée (HTTP, FTP,. . . ) parce qu'il faut interpréter les messages de façon différente pour chaque application.

- **Le Proxy** analyse du trafic échangé au niveau application (niveau 7) pour appliquer une politique de sécurité spécifique de chaque application.
  - ! Possibilité de modification des informations échangées entre le client et le serveur.
- Le **Proxy** peut faire aussi du cache et de la sécurité.

- **Avantage :** on peut intervenir de manière fine sur chaque zone des charges utiles transportées au niveau applicatif.

# Filtre applicatif (proxy ou mandataire)

- Pour accéder à l'extérieur il faut que le proxy transfert la requête vers le bon destinataire suivant le numéro de port de destination.
- Le proxy sert d'intermédiaire entre le client interne privé et le serveur externe public. Le client accède au proxy qui possède généralement une adresse IP publique et ce dernier transfert la requête vers le serveur convenable.

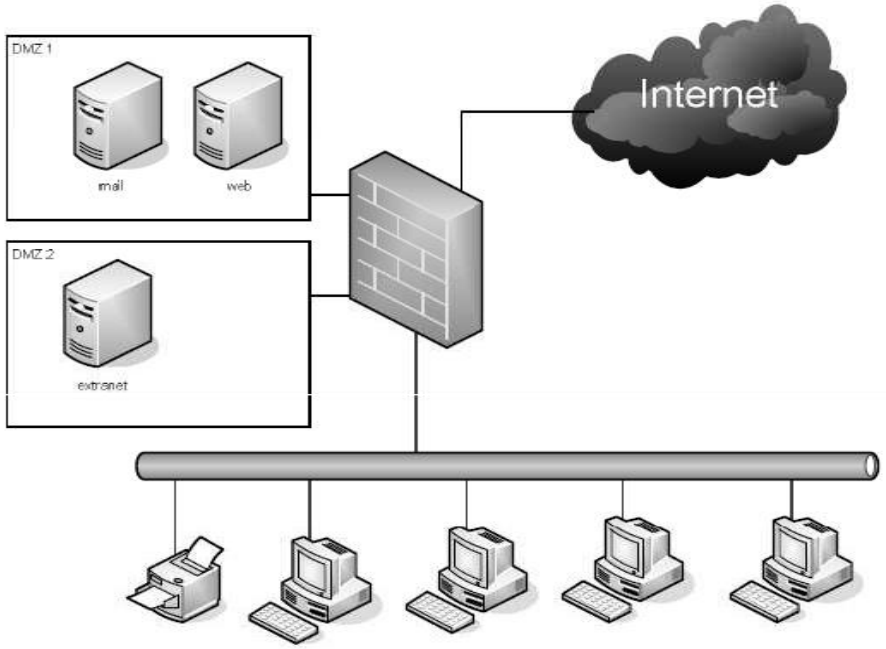


# Zone démilitarisée (DMZ)

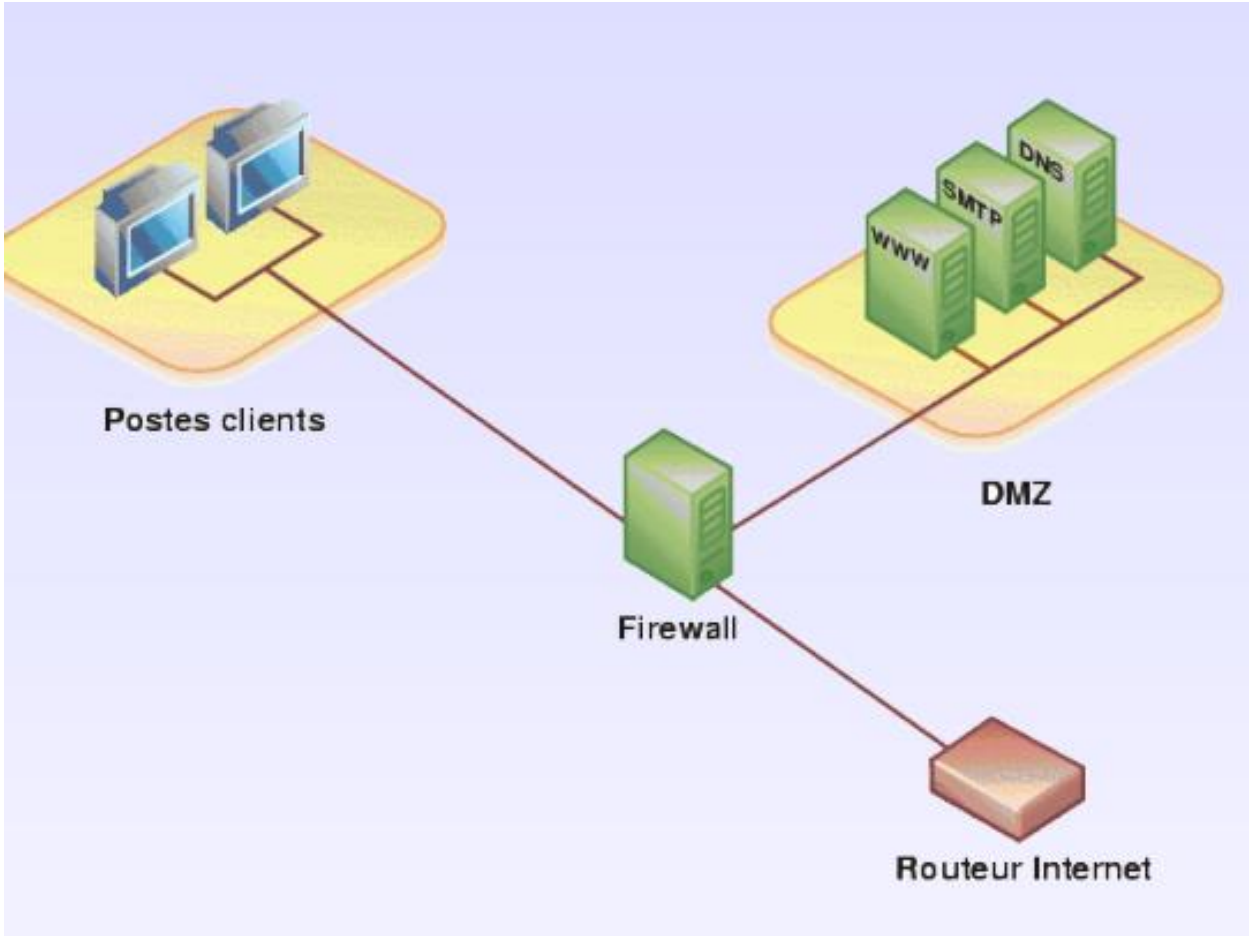
***Définition:***

*Une zone démilitarisée (DMZ) est un sous-réseau se trouvant entre le réseau local et le réseau extérieur. C'est une zone isolée à l'intérieur et de l'extérieur.*

- **Intérêt :**
  - La possibilité d'offrir des services sans les placer à l'intérieur
  - Rendre des machines accessible à partir de l'extérieur
  - (possibilité de mettre en place des serveurs (DNS, SMTP, . . . ).



# Architecture de pare-feu avec DMZ



# Politique de sécurité
# 1) Interdire tout par défaut

- **Présentation générale de cette approche:**
  - Tout ce qui n'est pas explicitement permis est interdit.
- **Mise en œuvre** :
  - **Analyse** des services utilisés par les utilisateurs.
  - **Exemple 1 :** Un hôte serveur de courrier doit pouvoir utiliser SMTP.
  - **Exemple 2 :** Un hôte devant accéder au Web doit pouvoir utiliser HTTP.
- **Définition des droits à donner :** définition de la politique de sécurité.
  - **Attribution des droits** dans un outil appliquant la politique,
  - **Suppression de tous les autres droits.**
- **Avantages/inconvénients**
  - Solution la plus sécuritaire et la plus confortable pour l'administrateur de la sécurité.
  - Solution qui limite considérablement les droits des usagers.
  - Solution la plus recommandée et la plus souvent utilisée.

# Politiques de sécurité : 2) Autoriser tout par défaut

- **Présentation générale de la solution :**
  - On permet tout sauf ce qui est considéré comme dangereux
    => Tout ce qui n'est pas explicitement interdit est autorisé.
- **Mise en oeuvre :**
  - On analyse les différents risques des applications qui doivent s'exécuter.
    => On en déduit les interdictions à appliquer, on autorise tout le reste.
  - **Exemple 1 :** Interdire complètement les accès en Telnet depuis l'extérieur ou les autoriser sur une seule machine.
  - **Exemple 2 :** Interdire les transferts FTP ou les partages NFS depuis l'intérieur (protection des données).
- **Avantages/inconvénients**
  - Solution inconfortable pour l'administrateur de la sécurité.
  - Solution qui facilite l'accès des usagers au réseau.
  - Solution peu recommandée et peu utilisée.

# Filtrage de paquets

- Filtrage statique
- But : autoriser / interdire le passage d'un paquet selon :
  - le port source/destination
  - le protocole UDP/TCP
  - l'adresse source/destination
  - le type de paquet ICMP
  - la taille du paquet
  - interface d'entrée / de sortie

# Filtrage de paquets

- Filtrage dynamique. En plus :
  - l'état des connexions du pare feu (statefull)
    - ce paquet est-il une réponse ?
    - ce paquet a-t-il des analogies avec un précédent (taille, source...) ?
  - modification du paquet à la volée (NAT)
  - redirection transparente du paquet
  - adapter dynamiquement ses règles (blocage du trafic de et vers un site anormal)

# Filtrage de paquets

- Décision
  - ACCEPT / DROP / REJECT
  - Inscriptions dans les logs
  - Déclencher une alarme
- Exemples de règles restrictives pour une salle TP étudiants
  - Autoriser en sortie :
    - le port 80 (http)
    - le port 443 (https)
    - le port 21 (ftp)
    - le port 22 (ssh)
    - le port 25 (smtp)
    - le port 110 (pop)
    - le port 53 (dns)

# Filtrage de paquets

Exemple plus complexe :

- Trafic du service comptabilité
- Trafic du service développement
- Accès pour le fournisseur X au service Y
- Accès du partenaire X au service Y pour une durée d
- ...

Attention : tout oubli est une porte d'accès de l'extérieur

# Zones importantes pour les pare-feux

- **Protocole de niveau liaison (PPP, niveau mac):**
  - Zone protocole utilisateur (démultiplexage): IP, IPX …
  - Zones adresses: Adresses Ethernet IEEE802, (permettent de déterminer la source et la destination sur la liaison donc l'entrée ou la sortie)
- **Protocole IP:**
  - Adresses source et destination.
  - Drapeaux (Flags): en particulier ceux qui concernent la fragmentation.
  - Le type de protocole destinataire: TCP, UDP, ICMP, ...
  - Analyse des zones d'extension par exemple en routage par la source => Demande de destruction de ces datagrammes car utilisation possible du routage par la source en attaque.

# Notions importantes pour les pare-feux

- **Protocole TCP :**
  - **Numéros de port source et destination:** permet d'estimer quel est le service concerné dans la mesure ou l'on respecte l'utilisation des numéros bien connus => Il est toujours possible pour un attaquant d'usurper un numéro de port.
  - **Drapeaux de contrôle (flags)**
    - **ACK:** positionné sauf dans le premier segment (utilisation possible pour bloquer des connexions).
    - **SYN:** positionné dans les deux premiers segments (permet d'identifier les connexions).
    - **RST:** fermeture non négociée de connexion.
- **Protocole d'application :**
  - Analyse non réalisée par les filtres de paquets mais par les proxys serveurs.
  - L'application filtrée doit être très stabilisée.

# Les principaux services Internet à contrôler

- **Le courrier électronique SMTP** Simple Mail Transfer Protocol.
- **Le transfert de fichiers FTP** File Transfer Protocol et **l'accès fichier distant** NFS Network File System.
- **Les accès à distance** protocoles telnet, rlogin, ssh
- **Le web** HTTP (Hypertext Transfer Protocol)
- **Les informations sur les utilisateurs:** finger
- **L'annuaire des noms de domaine DNS** Domain Name Service.
- **L'administration de réseau SNMP** Simple Network Management Protocol.
- **La synchronisation d'horloges NTP** Network Time Protocol.

# Les trois étapes du filtrage : Etape 1 : Spécification abstraite

- **Définir abstraitement la politique de sécurité :** ce qui est autorisé et ce qui est interdit
- **Choisir une politique d'ensemble:**
  - Solution 1) Tout ce qui n'est pas explicitement autorisé est interdit.
  - Solution 2) Tout ce qui n'est pas explicitement interdit est autorisé.
- **Enoncer des règles**
  - **Exemple de règle 1)** Autoriser un hôte intrene à recevoir du courrier électronique de toute provenance parceque c'est un serveur de courrier smtp.
  - **Exemple de règle 2)** Interdire à un hôte externe précis d'envoyer du courrier SMTP à un serveur de courrier interne parcequ'il est en liste noire.

# Les trois étapes du filtrage : Etape 2: Etablir des règles précises

- **Traduire la politique de sécurité en des règles précises concernant des communications IP**
  - Régles concernant des datagrammes IP : adresses source/destination, protocole utilisateur TCP port source/destination, indicateurs TCP, autres
  - Exemple 1) Règle interdisant tout par défaut

| Règle | IP Source | Port Source | IP Dest | Port Dest | Complément |
|---|---|---|---|---|---|
| Interdire | * | * | * | * | TCP |

  - Exemple 2) Règle autorisant la reception du courrier par un serveur

| Règle | IP Source | Port Source | IP Dest | Port Dest | Complément |
|---|---|---|---|---|---|
| Autoriser | * | * | Smtp-local | 25 | TCP |

  - Exemple 3)Règle interdisant l'émission par un serveur de courrier suspect

| Règle | IP Source | Port Source | IP Dest | Port Dest | Complément |
|---|---|---|---|---|---|
| Interdire | Smtp_distant | * | * | * | TCP |

26

# Les trois étapes du filtrage : Etape 3 : Configurer un outil précis

- **Rentrer les règles dans un pare-feu réel en utilisant la syntaxe et la sémantique de l'interface de l'outil**
  - Sémantique la plus fréquente : exploitation des règles dans l'ordre.
  - La première règle satisfaite provoque l'autorisation de la transmission ou la destruction du datagramme.
  - En fait un pare-feu interprète un programme qui est une suite de: si (condition sur datagramme) alors action (autoriser/interdire).
  - Exemple : Liste ordonnée des règles précédentes.

| Règle | IP Source | Port Source | IP Dest | Port Dest | Complément |
|---|---|---|---|---|---|
| Interdire | Smtp-distant | * | * | * | TCP |
| Autoriser | * | * | Smtp_local | 25 | TCP |
| Interdire | * | * | * | * | TCP |

  - Transformer les règles dans la syntaxe du pare-feu disponible

Exemple : Syntaxe de règle avec le pare-feu LINUX (iptables).

[root@ firewall]#iptables –A FORWARD –p smtp –d 192.168.0.10 –j ACCEPT

# Exemple de règles de filtrage

Les adresses IP contenues dans les paquets permettent d'identifier la machine émettrice et la machine cible, tandis que le type de paquet et le numéro de port donnent une indication sur le type de service utilisé.

- Le tableau ci-dessous donne des exemples de règles de pare-feu

| Règle | Action | IP source | IP dest | Protocol | Port source | Port dest |
|---|---|---|---|---|---|---|
| 1 | Accept | 192.168.10.20 | 194.154.192.3 | tcp | any | 25 |
| 2 | Accept | any | 192.168.10.3 | tcp | any | 80 |
| 3 | Accept | 192.168.10.0/24 | any | tcp | any | 80 |
| 4 | Deny | any | any | any | any | any |

# Règles de filtrage: **Le suivi de connexion**

- Le suivi de connexion est un concept essentiel dans le filtrage.
- Le principe du suivi de connexion permet de réaliser un "firewall statefull", c'est à dire qu'il va réagir intelligemment sur une connexion donnée, suivant son état.
- L'établissement d'une connexion TCP suit un protocole strict :
  - Une requête de synchronisation [SYN] de la part de l'initiateur du dialogue (le client). Une nouvelle connexion qui commence.
  - une réponse d'accusé réception de la synchronisation [SYN,ACK] de la part du serveur. La connexion est acceptée, elle est donc établie
  - un accusé réception du client [ACK]. La connexion se continue.

# Le suivi de connexion

Exemple: Nous allons établir une connexion TCP à partir du protocole HTTP :

| No. | Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|
| 10 | 10.181832 | 192.168.0.10 | 212.27.35.1 | TCP | 4252 > http [**SYN**] |
| 11 | 10.204707 | 212.27.35.1 | 192.168.0.10 | TCP | http > 4252 [**SYN, ACK**] |
| 12 | 10.204848 | 192.168.0.10 | 212.27.35.1 | TCP | 4252 > http [**ACK**] |
| 13 | 10.205333 | 192.168.0.10 | 212.27.35.1 | HTTP | GET / HTTP/1.1 |

Observons également les sockets.

- Trame n°10 :

  Le client [192.168.0.10] s'adresse au serveur [212.27.35.1] sur le port http (80). Il attend une réponse sur le port 4252 (c'est un numéro pris plus ou moins au hasard)

- Trame n°11 :

  Le serveur répond au client sur le port demandé (4252)

# Le suivi de connexion

Voyons la suite du dialogues entre les 2 machines précédente...

| No. | Time | Source | Destination | Protocol | Info |
|---|---|---|---|---|---|
| 29 | 10.427697 | 192.168.0.10 | 212.27.35.1 | TCP | 4252 > http [ACK] |
| 30 | 10.731147 | 192.168.0.10 | 212.27.35.1 | TCP | 4253 > http [SYN] |
| 31 | 10.752981 | 212.27.35.1 | 192.168.0.10 | TCP | http > 4253 [SYN, ACK] |
| 32 | 10.753165 | 192.168.0.10 | 212.27.35.1 | TCP | 4253 > http [ACK] |
| 33 | 10.753707 | 192.168.0.10 | 212.27.35.1 | HTTP | GET /images/titre.gif HTTP/1.1 |
| 34 | 10.780941 | 192.168.0.10 | 212.27.35.1 | TCP | 4252 > http [FIN, ACK] |

- Ce que nous observons ici, c'est l'établissement d'une nouvelle connexion TCP entre les mêmes machines. Pour éviter les  mélanges, le port du client n'est plus le même. C'est cette fois-ci 4253.
- Autrement dit, le même client (192.168.0.10) ouvre une nouvelle connexion TCP sur le même serveur (212.27.35.1), toujours sur le port 80, mais attend les réponses sur un nouveau port, alors que la connexion précédente existe toujours.
- Le client met fin à la précédente (trame 34) avec le flag [FIN] une fois seulement que la seconde connexion est établie.
- Dans ces conditions, il est pertinent de penser que cette nouvelle connexion est en relation directe avec la première. Nous dirons que c'est une connexion en relation avec la première.

# Affiner les règles de filtrage: Utilisation des indicateurs SYN, ACK

**Exemple d'utilisation d'indicateurs ('flags') TCP.**

- **Besoin fréquent :** autoriser la connexion d'un client interne sur un serveur interne (ou externe) mais refuser la connexion d'un client externe sur le même serveur interne.
- **Une solution :** l'utilisation de l'indicateur ACK en TCP.
- **Rappel :** Fonctionnement de l'ACK. ACK présent dans pratiquement tous les segments sauf le premier (exception les segments RST qu'on peut autoriser explicitement).
- Il suffit de bloquer un segment sans ACK pour interdire toute nouvelle connexion.

# Filtrage avec indicateur ACK TCP : Fonctionnement précis

- **Première politique : un client autorisé peut utiliser un service de numéro de port bien connu ou qu'il soit.**
  - La première régle autorise certains hôtes sélectionnés (possiblement tout mon domaine) à communiquer avec un service distant (interne ou externe) de numéro de port bien connu (noté ici service-tcp, par ex 80).
- **Seconde politique : un site externe ne peut commencer une communication avec un serveur interne sur le port bien connu mais il peut continuer une communication qui a été initialisée.**
  - La seconde régle autorise tous les hôtes de l'internet (internes ou externes) qui utilisent le numéro de port bien connu à communiquer à condition que le bit ACK soit positionné => on autorise en fait les réponses par un serveur à une communication initialisée en interne.

| Règle | IP Source | Port Source | IP Dest | Port Dest | Complément |
|---|---|---|---|---|---|
| Autoriser | Mes-hotes | * | * | Service-TCP | TCP |
| Autoriser | * | Service-TCP | * | * | TCP, $ACK_{29}$ |

# Pare-feux : le possible et l'impossible

- **Ce que peut faire un pare-feux :**
  - Etre un guichet de sécurité: un point central de contrôle de sécurité plutôt que de multiples contrôles dans différents logiciels clients ou serveurs.
  - Appliquer une politique de contrôle d'accès.
  - Enregistrer le trafic: construire des journaux de sécurité.
  - Appliquer une défense en profondeur (multiples pare-feux)

- **Ce que ne peut pas faire un pare-feux :**
  - Protéger contre les utilisateurs internes (selon leurs droits).
  - Protéger un réseau d'un trafic qui ne passe pas par le pare-feu (exemple de modems additionnels)
  - Protéger contre les virus.
  - Protéger contre des menaces imprévues (hors politique).
  - Etre gratuit et se configurer tout seul.

# Conclusion
# Avantages des filtres de paquets

- **Un filtre de paquets peut protéger** en un seul dispositif tout un réseau d'entreprise.
- **La mise en place du filtre peut être réalisée** par l'équipe système indépendamment des usagers qui gèrent les postes clients ou serveurs.
- **Les filtres de paquets sont très répandus** dans tous les routeurs sous forme de logiciels filtres logiciels libres ou propriétaires.

# En association avec le pare-feu : Traducteur d'adresses NAT

- **Traduction des adresses IP et TCP** : **NAT** 'Network Address Translation' et **PAT** 'Port Address Translation'

  => **Traduction combinée** de port et d'adresse

  - réseau: **NAPT** 'Network Address and Port Translation'.
  - Solution introduite pour économiser des adresses IPV4.

  - **Solution utilisée en sécurité**:
    - **NAPT permet de cacher le plan d'adressage interne:** les adresses IP et les numéros de port ne sont plus connus à l'extérieur.
    - **NAPT n'autorise pas les connexions initialisées** depuis le réseau externe.
    - Solution différente du filtrage de paquets mais solution complémentaire.

# Le NAT (Network Adress Translation)

- Permettre des utilisateurs dans un réseau IP privée (d'adresse privée) d'accéder au réseau IP public en utilisant des adresses publics.
- Deux types de Nating:
  - Statique: n adresse privées sont associes a n adresses publics en utilisant le table de nating statique et fixe.
  - Dynamique: association entre m<n adresse publique (généralement m=1) et n adresses privées.

# NAT Statique

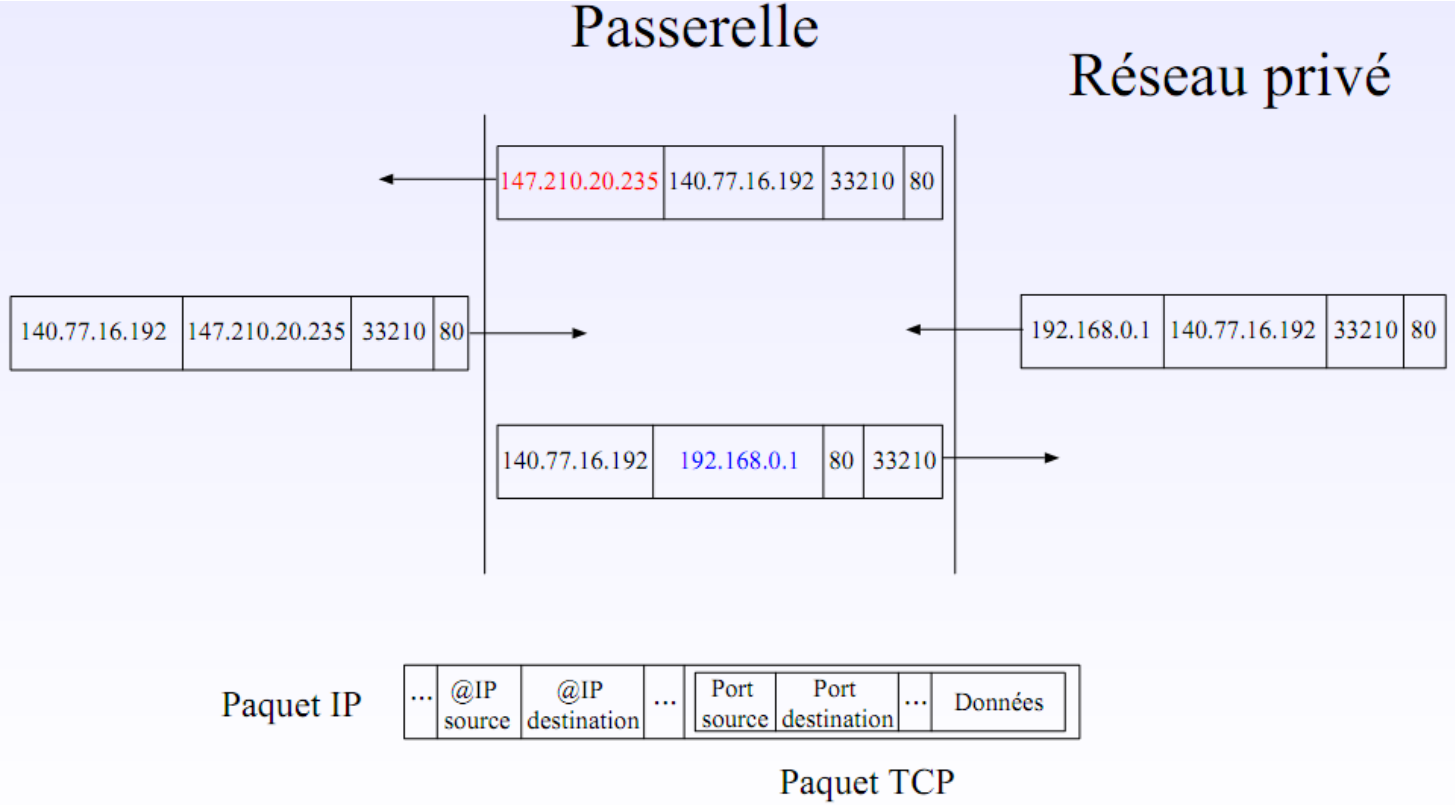
Tableau de Nating

| Adresse privée | Adresse publique |
|---|---|
| 10.0.0.1 | 193.22.35.43 |


- Dans le sens de sortie on change l'adresse privée en adresse public suivant le tableau de Nating.
- Dans le sens d'entrée on trouve la bonne destination dans le tableau de routage et on change l'adresse publique en adresse privée suivant le tableau de Nating.

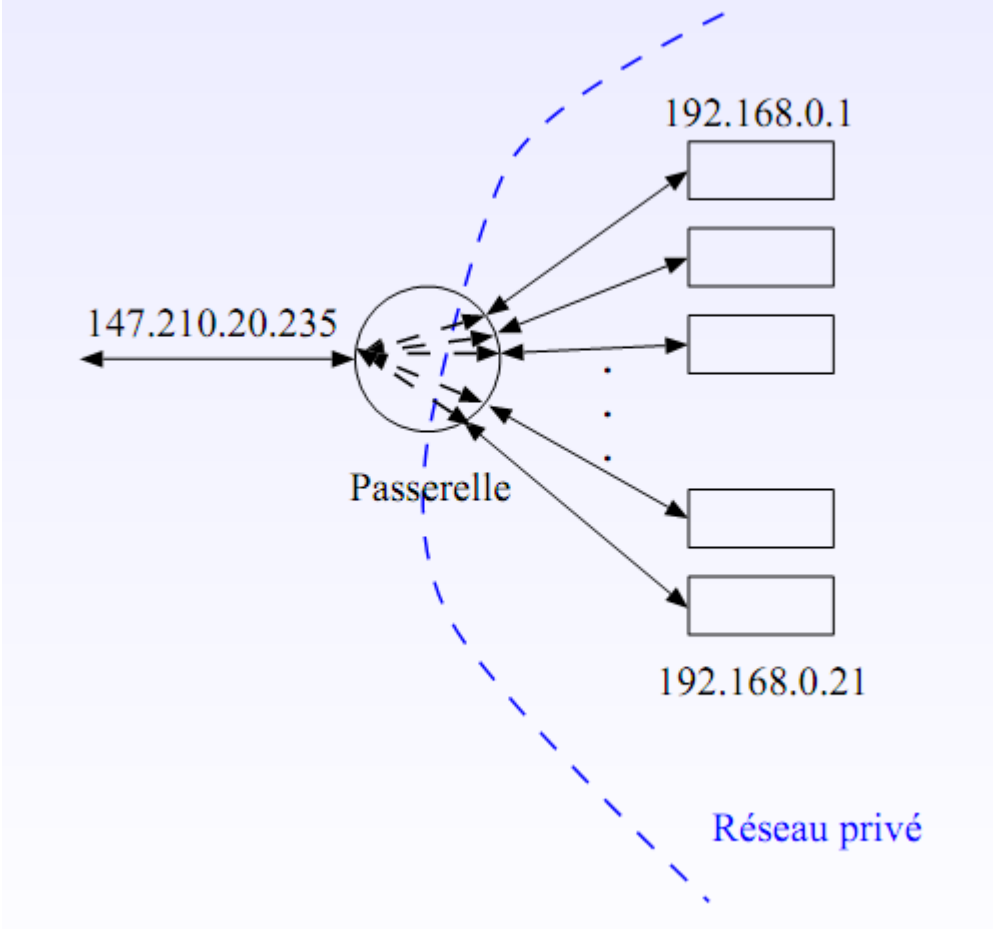
# NAT statique : Principe

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



# NAT dynamique ou Masquerading

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



# NAT dynamique ou Masquerading

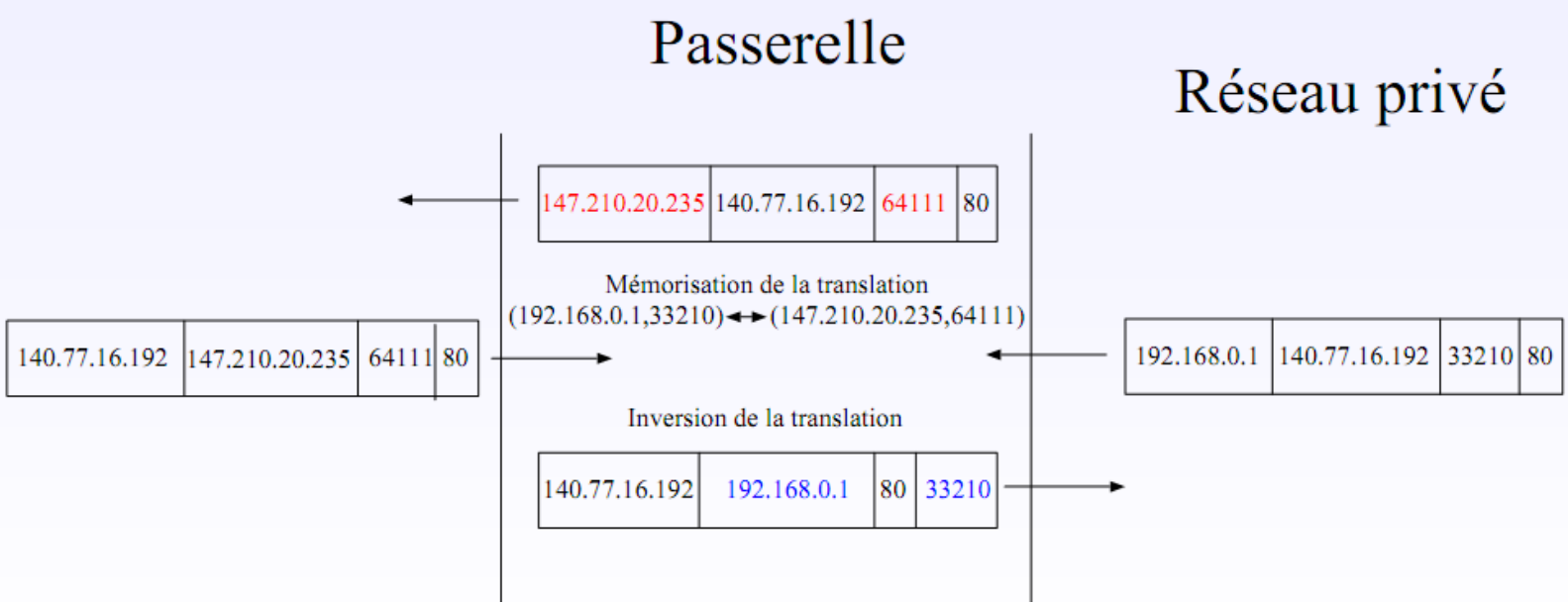
Association entre m adresses publiques et n adresses privées (m < n).

- **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

- 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 192.168.0.1



# NAT dynamique : Principe

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** ∃ 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 */

# Logiciels de filtrage de paquets

- Fonctionnalités de “firewall” filtrant directement implémentée dans le noyau Linux (NetFilter).
- 3 types de firewall filtrants Linux:
  - Ipfwadm. Jusqu’à la version 2.1.102 du noyau linux (obsolète)
  - Ipchains. Entre les versions 2.2.0 et 2.4 du noyau linux (également obsolète)
  - Iptables. À partir des noyaux 2.4

# Iptables: Fonctionnement

- 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.

# Iptables: Fonctionnement

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

**Si** le paquet est destiné à l'hôte local **Alors**
il traverse la chaîne INPUT.
**Si** il n'est pas rejeté **Alors**
il est transmis au processus impliqué.
**Sinon**
**Si** le paquet est destiné à un hôte d'un autre réseau **Alors**
il traverse la chaîne FORWARD
**Si** il n'est pas rejeté **Alors**
il poursuit alors sa route

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

# Iptables:

Fonctionnalités :

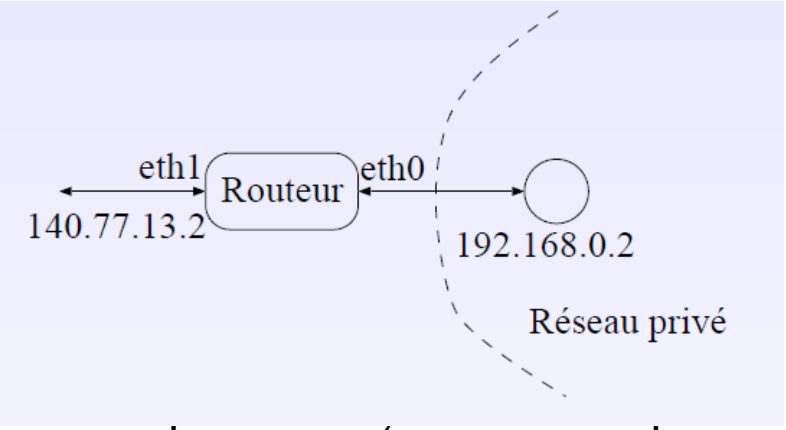
- Filtrage de paquets
- NAT
- Marquage de paquets

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




- 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):

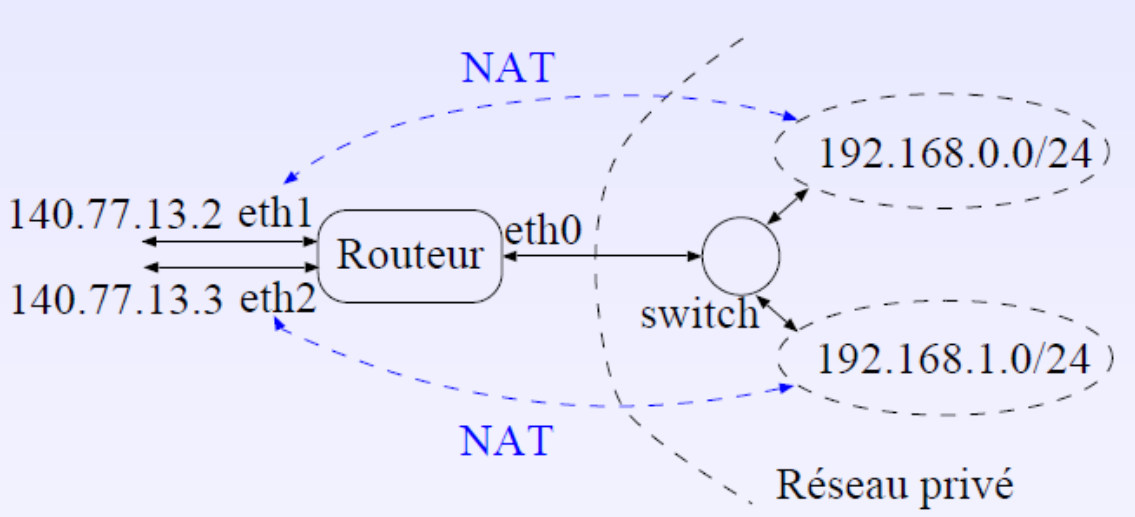
  ***iptables -t nat -A POSTROUTING -s 192.168.0.2 -o eth1 –j SNAT -to-source 140.77.13.2***

- 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



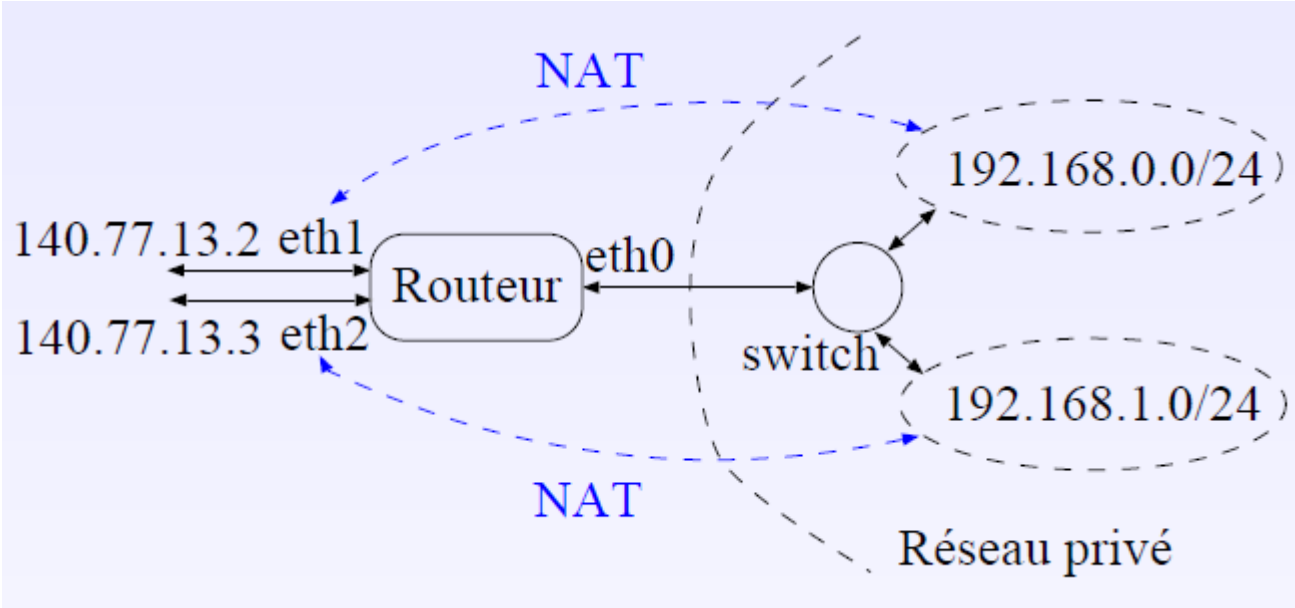

- Association entre toutes les adresses privées du sous-réseau 192.168.0.0/24 avec l'interface eth1.

  *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*

# Iptables: filtrage

- 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: filtrage

- 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: 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
```

# 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, . . . )