# Sécurité des réseaux informatiques

## Firewall : les iptables

Victor OYETOLA

Université d'Abomey-Calavi, chef Service de la Promotion des TICs

August 19, 2022

# Firewall

## Firewall: 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.

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.

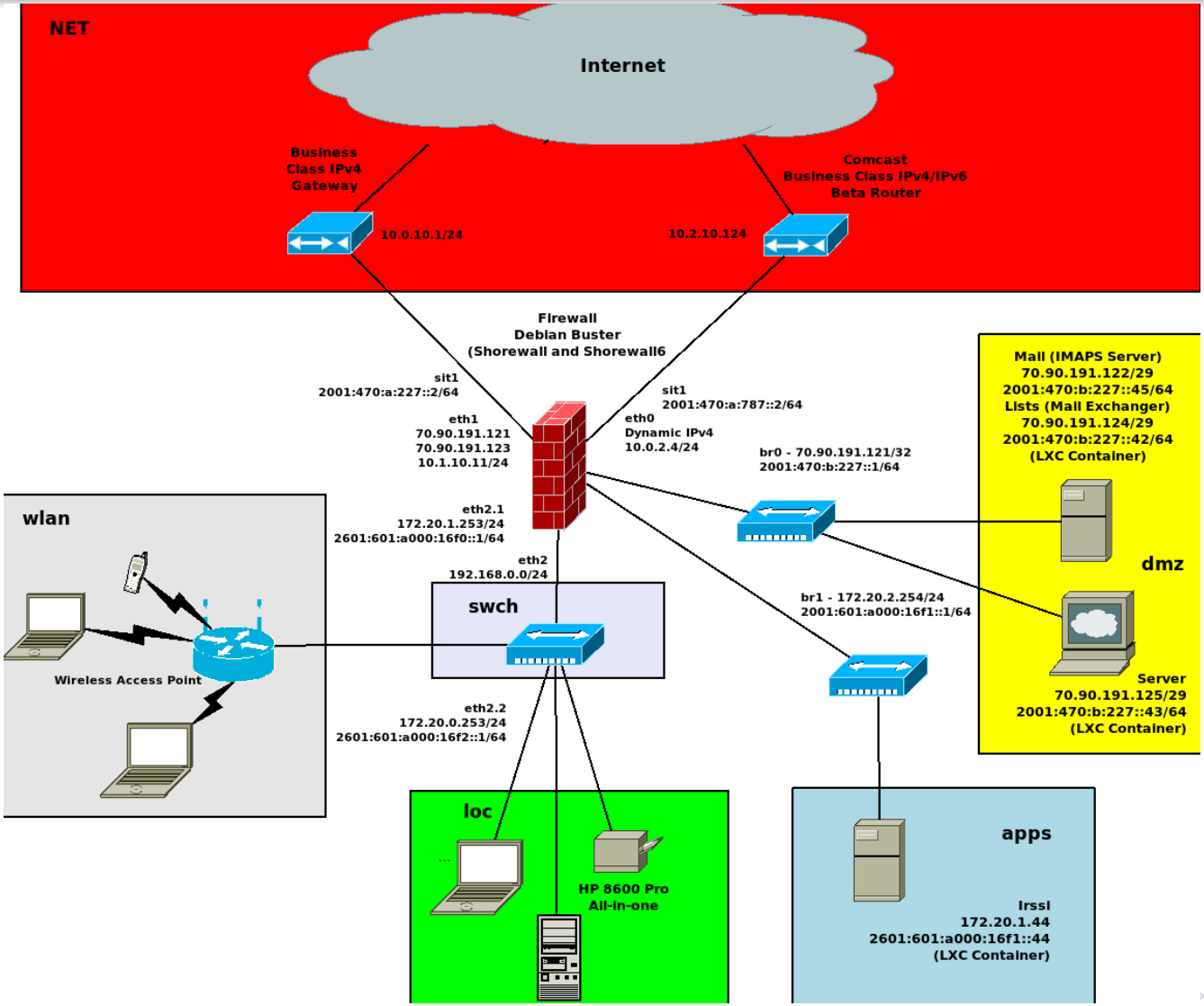
## Les types de Firewall

Quelques types de firewalls:

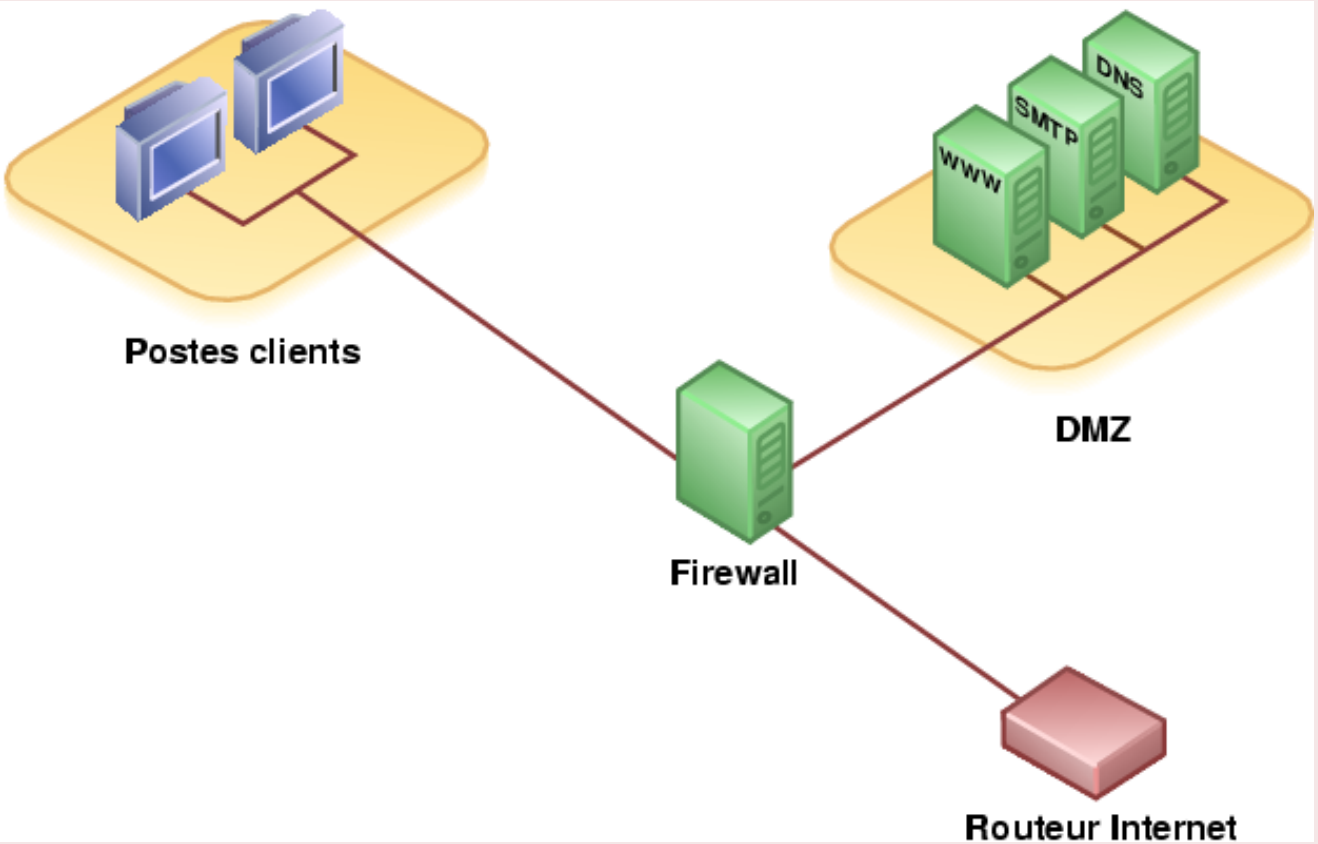
- **Les firewalls nouvelle génération (NGFW):** combinent technologie traditionnelle et fonctionnalités supplémentaires: inspection du trafic crypté, des systèmes de prévention des intrusions, des antivirus, (DPI, ou deep packet inspection).
- **Les firewalls à proxy:** filtrent le trafic réseau au niveau de l'application.
- **Les firewalls traditionnels**

NB: Dans tous les cas on distingue:

- les firewalls matériels: Juniper, Fortinet, Mikrotik, Cisco
- les firewalls logiciels: shorewall, pfsense, ufw, firewalld etc.


NET
Internet
Business
Class IPv4
Gateway
10.0.10.1/24
Comcast
Business Class IPv4/IPv6
Beta Router
10.2.10.124
Firewall
Debian Buster
(Shorewall and Shorewall6
sit1
2001:470:a:227::2/64
eth1
70.90.191.121
70.90.191.123
10.1.10.11/24
sit1
2001:470:a:787::2/64
eth0
Dynamic IPv4
10.0.2.4/24
br0 - 70.90.191.121/32
2001:470:b:227::1/64
Mail (IMAPS Server)
70.90.191.122/29
2001:470:b:227::45/64
Lists (Mail Exchanger)
70.90.191.124/29
2001:470:b:227::42/64
(LXC Container)
dmz
wlan
eth2.1
172.20.1.253/24
2601:601:a000:16f0::1/64
eth2
192.168.0.0/24
swch
br1 - 172.20.2.254/24
2001:601:a000:16f1::1/64
Wireless Access Point
Server
70.90.191.125/29
2001:470:b:227::43/64
(LXC Container)
eth2.2
172.20.0.253/24
2601:601:a000:16f2::1/64
loc
HP 8600 Pro
All-in-one
apps
Irssi
172.20.1.44
2601:601:a000:16f1::44
(LXC Container)

# La DMZ et le firewall



## DMZ: Définition

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

## DMZ: 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.

## DMZ: Intérêt

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

## iptables

iptables est un logiciel de filtrage réseau utilisé sur une machine linux pour empêcher les accès non autorisés entre le réseau internet et le reseau local (DMZ et LAN)

## iptables -L (cette commande affiche les lignes suivantes:)

Chain INPUT (policy ACCEPT)
target prot opt source destination
Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination

# tables et chaines

Au début, comme le montre l'option -L de la commande iptables, tout passe dans toutes les directions **(policy ACCEPT)**, c'est-'a-dire pour chacune des trois chaînes par defaut de la table filter (Elle même par défaut):

- INPUT (destination le routeur firewall lui même)
- OUTPUT (La source le routeur firewall lui même)
- FORWARD (la paquet traverse le routeur)

# tables et chaines

## à destination du routeur

| table | Chaîne |
|---|---|
| raw | PREROUTING |
| mangle | PREROUTING |
| nat | PREROUTING |
| mangle | INPUT |
| filter | INPUT |

# tables et chaines

Nous rencontrons quatre tables :

1. la table raw (brute) ou conntrack ;
2. la table mangle utilisée en général pour changer certains champs en-tete de la trame;
3. la table nat, utilisée pour la traduction d'adresses;
4. la table filter, utilisée pour le filtrage d'adresses.

Les chaînes PREROUTING sont utilisées avant toute décision de routage par Linux et INPUTaprès celles-ci. De même, le noyau de Linux fait passer les trames crées par l'ordinateur et destinées à un autre hôte par les chaînes suivantes dans cet ordre avant de les envoyer à l'interface réseau :

# tables et chaines

## Le routeur est la source du paquet

| table | Chaîne |
|---|---|
| raw | OUTPUT |
| mangle | OUTPUT |
| nat | OUTPUT |
| filter | OUTPUT |
| mangle | POSTROUTING |
| nat | POSTROUTING |

Les chaînes OUTPUT sont traitées avant toute décision de routage et les chaînes POSTROUTING après celles-ci.

# tables et chaines

| table | Chaîne |
|---|---|
| raw | PREROUTING |
| mangle | PREROUTING |
| nat | PREROUTING |
| mangle | FORWARD |
| filter | FORWARD |
| mangle | POSTROUTING |
| nat | POSTROUTING |

Enfin une trame récupérée sur une interface réseau mais destinée à un autre hôte passe par les chaînes suivantes dans cet ordre avant d'être envoyée à une autre interface réseau :

*

# les commandes iptables

## Syntaxe générale

iptables [-t table] command [match] [target/jump]

Où le nom de la table est **raw, mangle,nat ou filter**. Par défaut il s'agit de la table filter.
Si tous les critères spécifiés par le paramètre match sont rencontrés alors l'instruction [target/jump] est exécutée

## exemple de commande

iptables -t filter -A INPUT -p ICMP -j DROP

# les commandes d'affichage

La commande -L ou - -list affiche toutes les entrées de la chaîne spécifiée (ou de toutes les chaînes si aucune chaîne n'est spécifiée)

## exemple de commande

iptables -t nat -L INPUT

# les commandes d'affichage

Pour choisir la politique de filtrage, on utilise la commande :

```
iptables -P chain policy
```

où chain est le nom d'une chaîne et policy l'une des deux valeurs:

- ACCEPT (pour tout accepter sauf ce qui sera explicitement rejeté)
- DROP (pour tout rejeter sauf ce qui sera explicitement accepté)

**exemple de politique de filtrage**

```
iptables -P INPUT DROP
```

# Ajouter une règle

Une fois la politique choisie, on peut ajouter des règles aux chaînes pour affiner le filtage

## exemple de politique suivie d'un filtrage

```
iptables -P INPUT DROP
iptables -A INPUT -s 10.10.30.0/24 -j ACCEPT
iptables -A INPUT -s 10.10.20.0/24 -j ACCEPT
```

# Supprimer une règle

On peut supprimer des règles d'une chaîne une à une grâce à la commande -D (ou - -delete ), avec les mêmes paramêtres que pour -A, ou vider la chaîne complète grâce à la commande : **iptables -F**

## exemple de suppression d'une règle

```
iptables -D INPUT -s 10.10.30.0/24 -j ACCEPT
iptables -D INPUT -s 10.10.20.0/24 -j ACCEPT
```

# utilisation de script

## monscript.sh

```
# !/bin/bash
iptables -P INPUT DROP
iptables -A INPUT -s 10.10.20.0/24 -j ACCEPT
iptables -A INPUT -s 10.10.30.0/24 -j ACCEPT
```

net Rendre le script exécutable avec la commande chmod +x ./monscript.sh

Un filtrage général permet d'accepter (ou d e rejeter) :
tous les paquets d'une interface réseau donnée (en entrée ou en sortie) :

```
iptables -A chain -i interface [jump]
```

cette première règle va avec les chaines INPUT, FORWARD et PREROUTING

```
iptables -A chain -o interface [jump]
```

cette deuxième règle va avec les chaines OUTPUT, FORWARD et POSTROUTING

Un filtrage général permet d'accepter (ou de rejeter) :

- une adresse IP source ou destination donnée:

```
iptables -A chain -s adresse [jump]
iptables -A chain -d adresse [jump]
```

l'adresse peut être 192.168.0.0/24 ou 192.168.0.0/255.255.255.0

- un protocole donné:

```
iptables -A chain -p protocol [jump]
```

Le protocol peut être TCP, UDP et ICMP ou l'un de ceux spécifiés dans /**etc**/**protocols**. Il peut également être une valeur entière ( 1 pour ICMP par exemple)

En bloquant tout par défaut dans la politique de la chaîne INPUT, même le loopback sera bloqué. Il serait par exemple utile de l'autoriser

```
iptables -P INPUT DROP
iptables -A INPUT -i lo -j ACCEPT
```

Un filtrage UDP permet d'accepter (ou de rejeter) suivant le port source ou le port de destination :

iptables -A chain -p udp - -sport port [jump]
iptables -A chain -p udp - -dport port [jump]
ex:iptables -A INPUT -p udp - -dport 53 -j ACCEPT

Un filtrage TCP permet d'accepter (ou de rejeter) suivant le port source ou le port de destination :

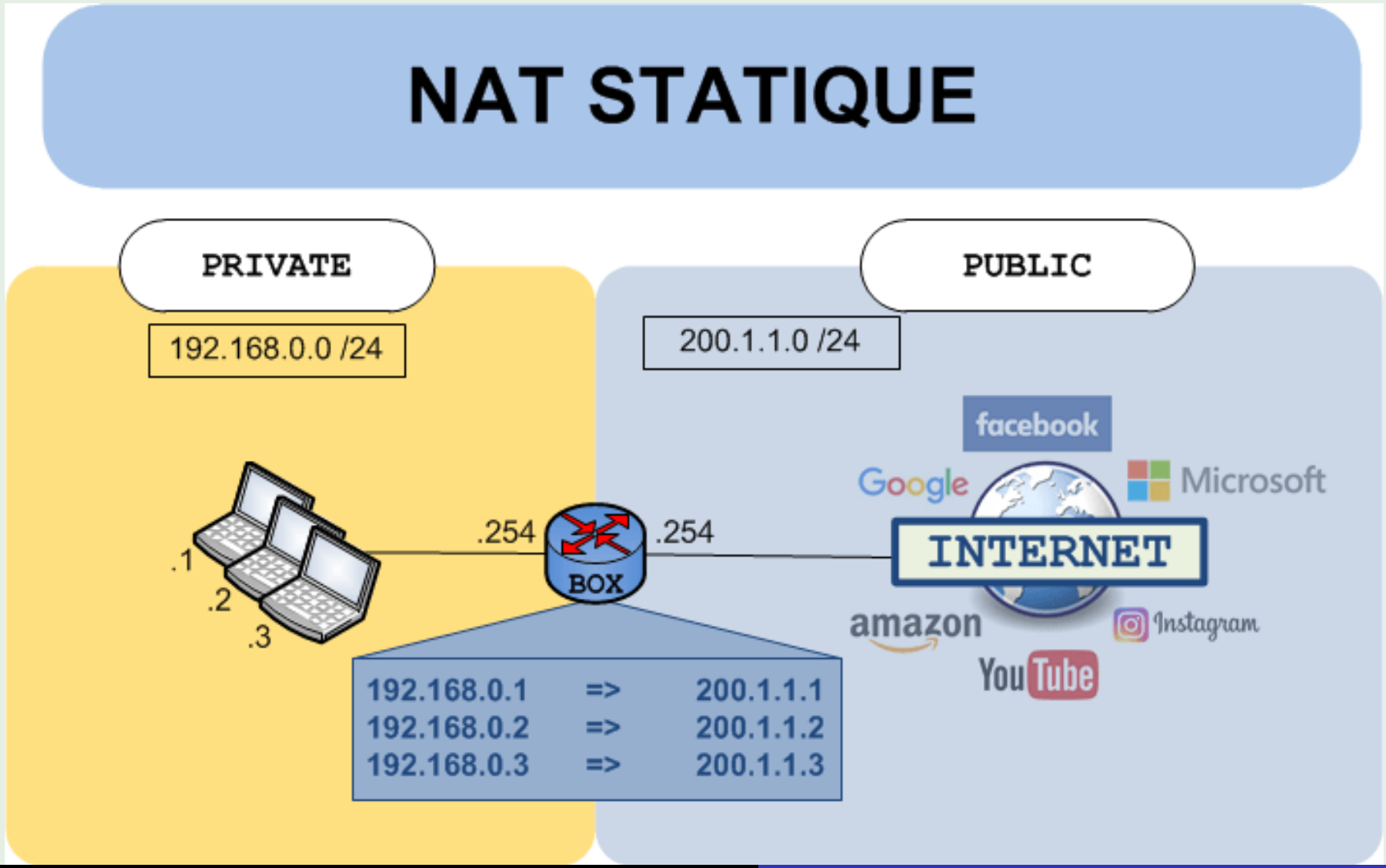
iptables -A chain -p tcp - -sport port [jump]
iptables -A chain -p tcp - -dport port [jump]
ex:iptables -A INPUT -p tcp - -dport 22 -j ACCEPT

un serveur NAT ( Network Address Translation ) traduit les adresses IP source et/ou destination des paquets en adresses différentes en jouant sur les numéros de port. Le serveur NAT reçoit le paquet, change l'en-tête IP avant l'envoi. La partie netfilter de Linux permet de mettre en place facilement ces traductions d'adresse avec la table nat l'une des cinq cibles suivantes :

- La cibleSNAT, pour changer l'adresse source des paquets.
- La cible DNAT, pour changer l'adresse de destination.
- La cible MASQUERADE est analogue à SNAT et s'utilise lorsque le serveur nat est en DHCP
- La dernière cible s'appelle REDIRECT

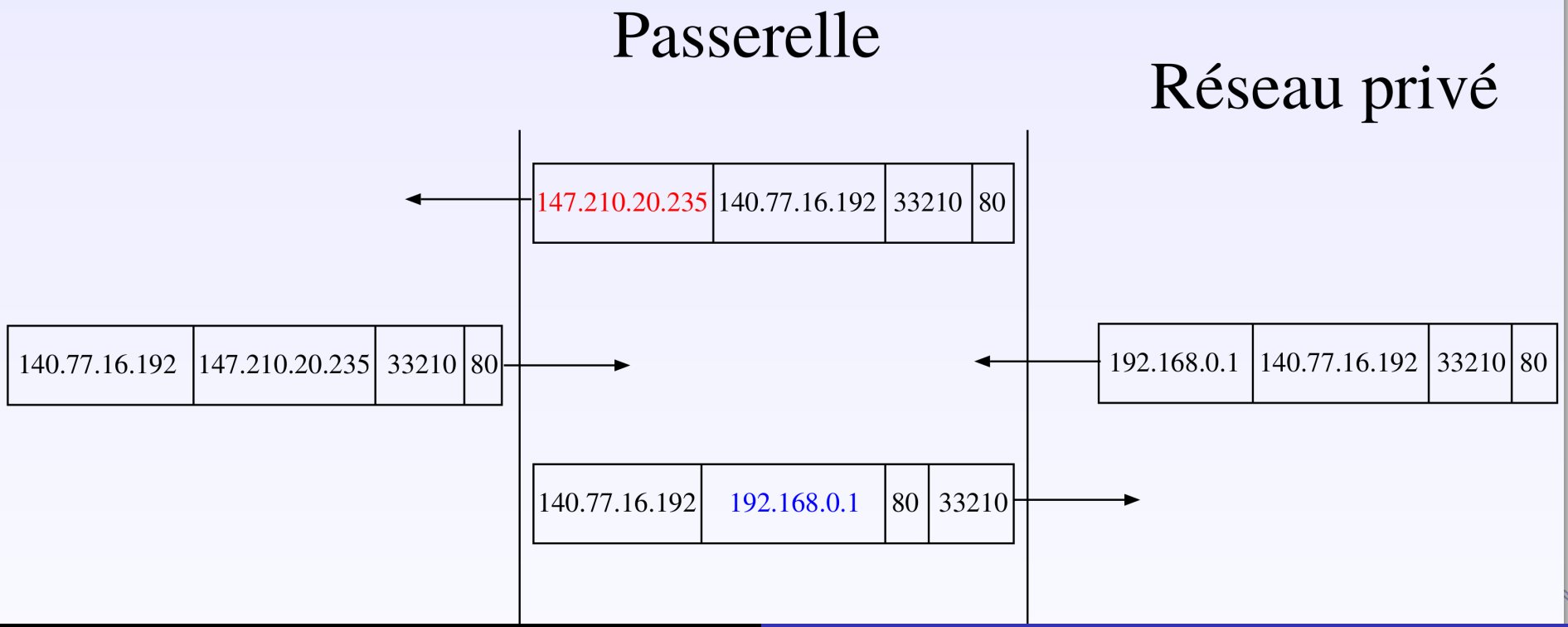
Pour netfilter, la cible SNAT et MASQUERADE n'est valide que dans la table nat, à l'intérieur de la chaîne POSTROUTING.

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT - -to 1.2.3.4
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 -j SNAT - -to 217.115.95.34
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 -j MASQUERADE
```

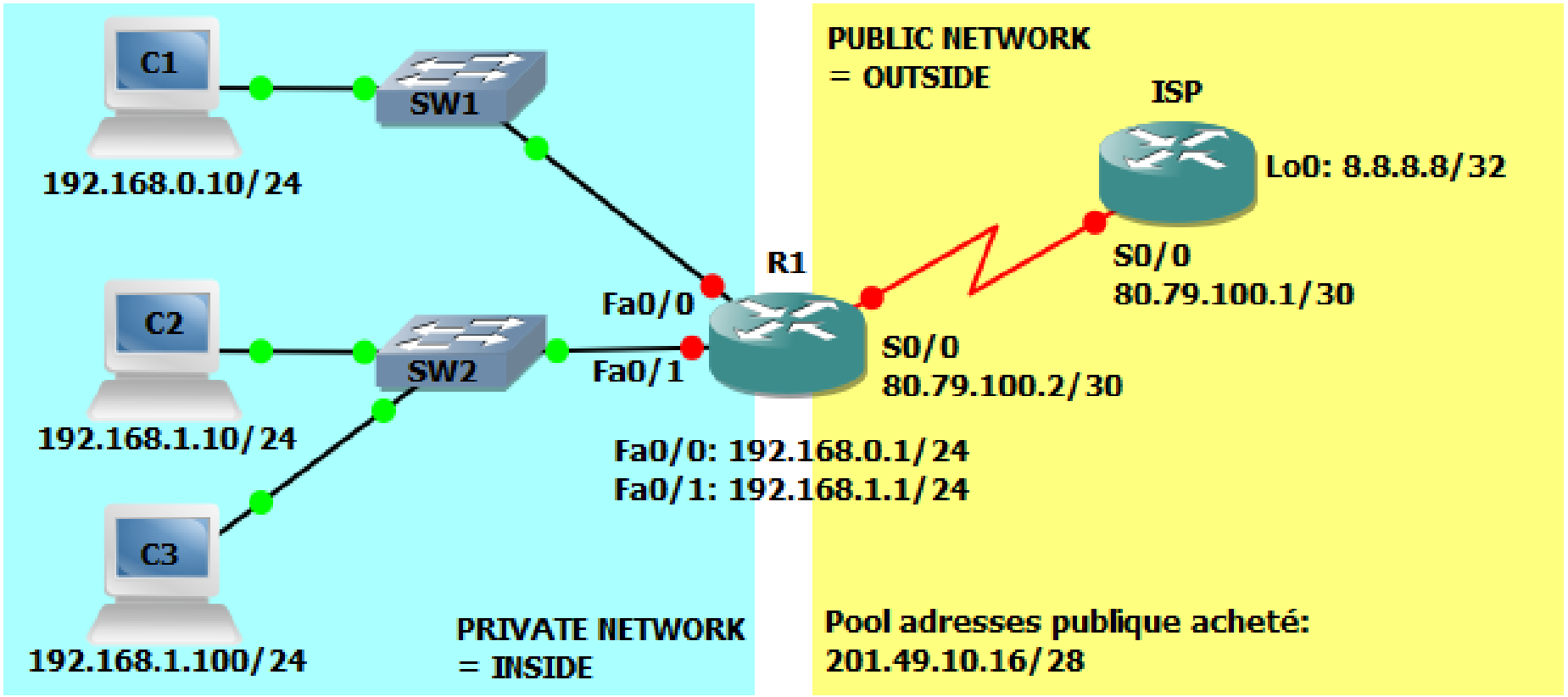

NAT STATIQUE
PRIVATE
192.168.0.0 /24
PUBLIC
200.1.1.0 /24
facebook
Google
Microsoft
INTERNET
amazon
Instagram
YouTube
.1
.2
.3
.254
.254
BOX
192.168.0.1 => 200.1.1.1
192.168.0.2 => 200.1.1.2
192.168.0.3 => 200.1.1.3

# NAT statique : Principe

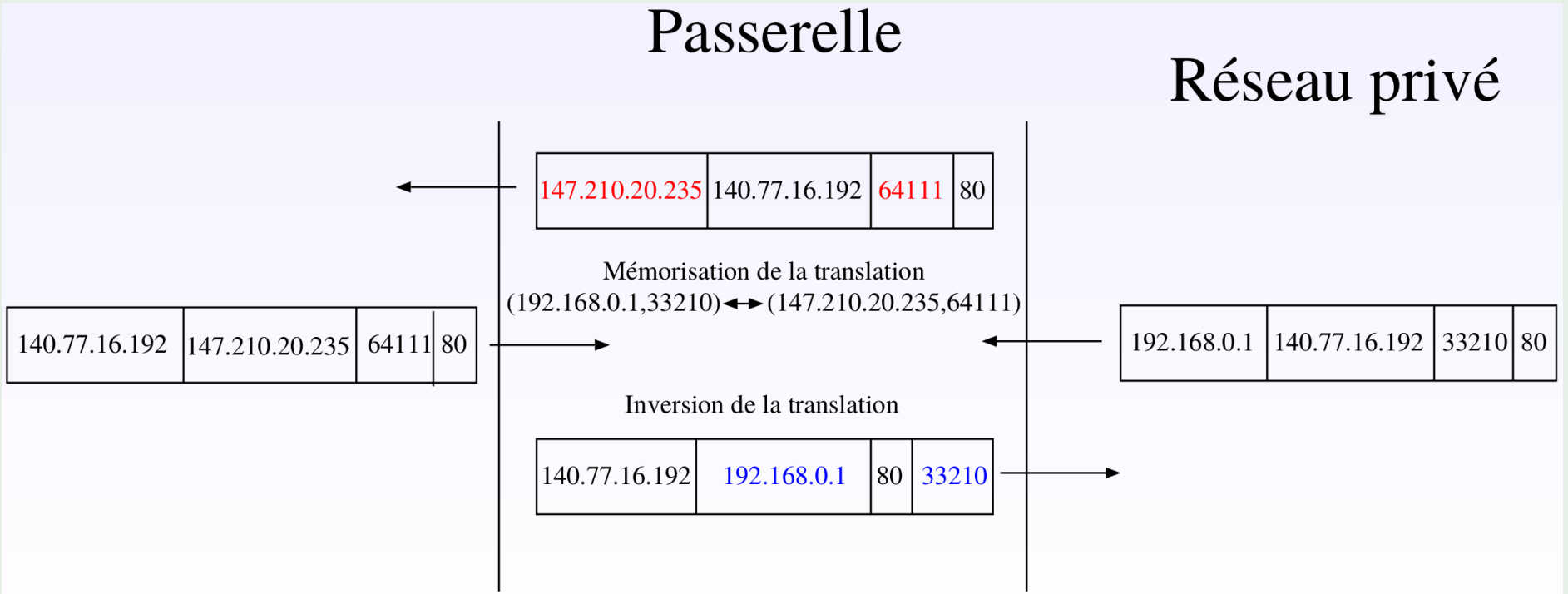
Pour chaque paquet sortant (resp. entrant), la passerelle modifie l'adresse source (resp. destination).



# Le nat dynamique



# Le nat dynamique: principe



Pour netfilter , la cible DNAT n'est valide que dans la table nat, à l'intérieur de la chaîne PREROUTING

```
iptables -t nat -A PREROUTING -i eth0 -j DNAT - -to 5.6.7.8
iptables -t nat -A PREROUTING -p tcp - -dport 80 -i eth0 -j DNAT - -to 5.6.7.8 :8080
```

Si la politique par défaut est DROP dans la chaîne FORWARD, on doit ajouter une règle autorisant la retransmission de requêtes HTTP entrantes afin que le routage NAT de destination soit possible

```
iptables -A FORWARD -i eth0 -p tcp –dport 80 -d 5.6.7.8 -j ACCEPT
```

## Exemple:

iptables -t nat -A PREROUTING -p tcp - -dport 80 -i eth0 -d 82.238.22.47 -j DNAT - -to 192.168.0.1

Cette commande va permettre de rediriger toutes les requêtes à destination du 82.238.22.47 au port 80 vers le 192.168.0.1 au même port. Si on veut changer rediriger un port différent vers la machine locale on procède comme suit:

## Exemple:

iptables -t nat -A PREROUTING -p tcp - -dport **8080** -i eth0 -d 82.238.22.47 -j DNAT - -to 192.168.0.1:**80**

## Le port forwading ou Traduction d'adresse destination

Le port forwading est une conséquence directe du NAT, le réseau interne étant privé, il est impossible de disposer d'un serveur d'application derrière un routeur NAT. le port forwading est une solution à ce problème et consite à rediriger un port du routeur externe vers un serveur présent dans le réseau privé.