ASR4
réseaux

Barrière de Sécurité

Introduction

Serveur de proximité, pare-feu IP, filtrage, architecture

1

Objectif des gardes barrières

- Protéger un environnement (vis à vis de l'extérieur et de l'intérieur)
  - tout n'est pas bien administré
  - des machines ne doivent pas être accessibles par tous
  - certaines doivent être « accessibles" (serveur WWW, FTP, Courriel)
- Contrôler les accès entrant et sortant
  - Contrôler/espionner
  - autoriser certains services seulement
    - dans un sens pas dans l'autre
    - vers/depuis certaines machines seulement

2

Une vue d'ensemble

- Nécessiter de définir une politique sécurité
  - tout interdire ou ouverture sélective ?
- Structurer le réseau
  - pour séparer les communautés
  - permettre des délégations de "pouvoir"
    - une partie "ouverte" (sans risques)
    - le reste accessible sur critères
- Penser sécurisation dans les 2 sens !

3

Pare-feu

firewall,

Mur pare-feu , coupe-feu

4

Pare-feu

- Dispositif informatique
  - qui **filtre** les paquets IP les segments TCP et ou les datagramme UDP entre un réseau interne et un réseau public
  - qui effectue de la translation d'adresses IP

Niveau OSI

• Mandataire -Proxy	applicatif
• Pare-feu - Firewall	Transport
	Réseau

5

Types de Pare-feu

- Dispositif informatique
  - qui **filtre** les paquets IP les segments TCP et ou les datagramme UDP entre un réseau interne et un réseau public
  - qui effectue de la translation d'adresses IP
- pare-feu :
  - routeur filtrant
  - une station équipée de deux interfaces réseaux – appelé parfois bastion

6

Translation d'adresses - NAT

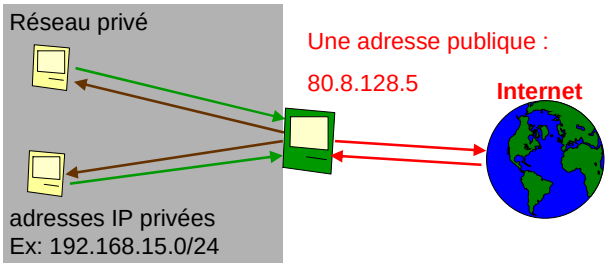
- Network Adresse Translation - Masquerading –

Mascarade - usurpation d'identité

Partager une connexion permet de relier plusieurs machines à Internet (ou à un autre réseau) au travers d'une seule machine (la passerelle)

7

Partage de connexion



Les hôtes du réseau privé ont accès à tous les services sur Internet

8

Parefeu sous Linux

module destiné au filtrage réseau: netfilter

Commande: iptables

9

Tables – type de traitement

- filter Cette table permet de filtrer les paquets Typiquement ce sera pour les accepter ou non
- nat translations d'adresse (ou de ports) utiliser pour partager une connexion
- mangle modification des entêtes des paquets

10

Filtrage sous Linux

11

Chaînes prédéfinies : Points de filtrage

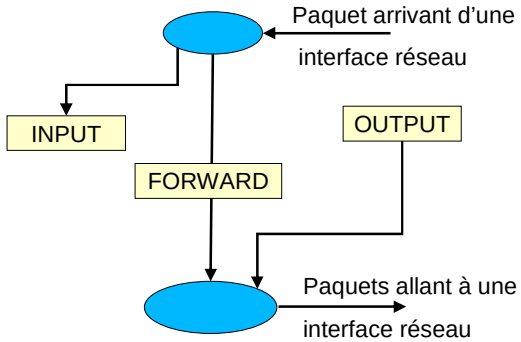
Chaînes prédéfinies déterminent les paquets/trames qui seront traités:

- INPUT les paquets entrants à destination de l'hôte
- OUTPUT des paquets sortant dont la source est l'hôte
- FORWARD les paquets en transit (entrants ou sortants) sur l'hôte

Une trame/paquet est de type « INPUT », « OUTPUT », ou exclusif « FORWARD »

12

Illustration des chaînes pour le filtrage



13

Cibles – actions pour le filtrage

- ACCEPT Les paquets/segments poursuivront leur cheminement au travers des couches réseaux
- DROP refus des paquets (qui seront donc ignorés)
- *REJECT* refus du paquets et envoie d'une réponse à l'émetteur pour lui signaler que son paquet a été refusé
- LOG enregistrement un message dans **`/var/log/messages`**

14

Option d'iptables

-L Affiche toutes les règles de la table indiquée
-F Supprime toutes les règles de la table sauf la politique par défaut
-P Modifie la politique par défaut
-A Ajoute une règle à la fin de la table spécifiée
-I Insère la règle avant celle indiquée
-D Supprime une règle

15

Exemples de filtrage

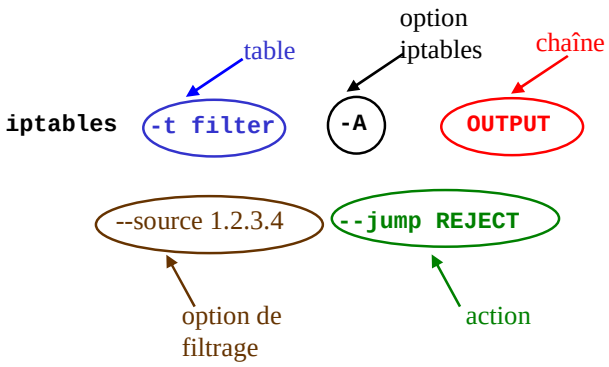
```
iptables -t filter –F
```
// plus de règles de filtrage mais
// Ne change pas les politiques par défaut

```
iptables -t filter -P OUTPUT DROP
```
//Politique par défaut pour la chaîne OUTPUT est « DROP »

```
iptables -t filter –L
```
// affiche les règles de filtrage et les politiques de filtrage par défaut

16

Format des règles de filtrage

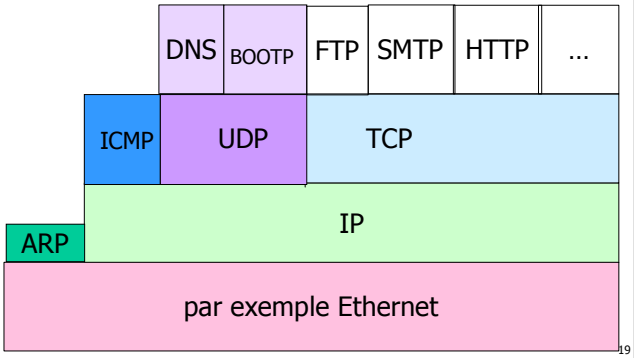


17

Option de filtrage

18

Protocoles liés à TCP/IP



19

Option de filtrages

Trames
- --in-interface : Interface réseau d'entrée
- --out-interface : Interface réseau de sortie

Paquet IP
- --source : Adresse IP origine du paquet
- --destination : Adresse IP de destination
- --protocol : tcp, udp, icmp ou all
  correspondant au champ « protocole » de l'entête IP

20

Exemples de filtrage

```
iptables -t filter -A OUTPUT
 --destination 192.168.30.45
   --jump ACCEPT
```
// les paquets à destination de 192.168.30.45 (sortant) peuvent partir

```
iptables -t filter -A INPUT
 --source 192.168.30.45
  --jump ACCEPT
```
// les paquets venant de 192.168.30.45 (entrant) sont acceptés

21

Exemples de filtrage

```
iptables -t filter -A FORWARD
  --protocol ICMP  --jump ACCEPT
```
// les paquets ICMP en transit sont routés

```
iptables -t filter -A FORWARD
  --protocol TCP  --jump ACCEPT
```
// les datagrammes TCP en transit sont routés

22

Option de filtrages (suites)

Uniquement segment TCP ou datagramme UDP
- --source-port : port de la source du segment
- --destination-port : port de la destination du segment

23

Option de filtrages (suites)

Uniquement segment TCP
- --state : [« ajouter module state » : -m state]
  - NEW : ouverture de connexion
  - ESTABLISHED : déjà établie
  - RELATED : nouvelle connexion liée à une connexion déjà établie

24

Exemples de filtrage de segments TCP

```
iptables -t filter -A OUTPUT
    --protocol tcp --source-port 80
    –-jump ACCEPT
iptables -t filter -A OUTPUT
    --protocol tcp ! --source-port  80
    --jump DROP
```

//Ces règles permettent de laisser passer tout le trafic TCP sortant du port 80.
Par contre les autres segments sortants TCP sont ignorés

25

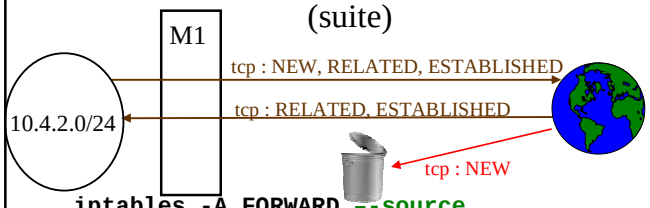
Exemples de filtrage de segments TCP

```
iptables -t filter -A INPUT
   --protocol tcp
   --destination-port 80 –-jump ACCEPT
iptables -t filter -A INPUT
      --protocol tcp
      ! --destination-port 80
      –-jump DROP
```

Ces règles permettent de laisser passer tout le trafic TCP entrant sur le port 80
Par contre les autres segments entrants TCP sont ignorés

26

Exemples de filtrage (suite)




```
iptables -A FORWARD --source
 10.4.2.0/24
  --protocol tcp --jump ACCEPT
 iptables -A FORWARD --destination
 10.4.2.0/24 --protocol tcp
  -m state --state   ESTABLISHED,
 RELATED
  --jump ACCEPT
```

27

Mascarade sous Linux

Translation d'adresses

28

Chaînes prédéfinies – Points de mascarade

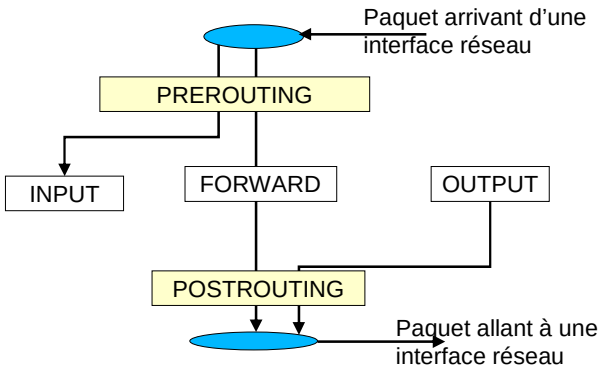
Pour « NAT – network adresse translation» -

Déterminer les paquets qui seront traités:
- PREROUTING les paquets entrants (destination hôte ou paquet en transit)
- POSTROUTING Les paquets sortants (en transmis ou crées par l'hôte)

Une trame/paquet est de type « PREROUTING », ou non exclusif « POSTROUTING»

29

Illustration des chaînes pour NAT



30

Cibles – actions pour le NAT

- **MASQUERADE** - POSTROUTING
  Elle change l'adresse IP de l'émetteur par adresse IP de l'interface spécifiée
- **SNAT** - POSTROUTING
  Elle change l'adresse IP de l'émetteur par la valeur fixe spécifiée
- **DNAT** - PREROUTING et OUTPUT
  Elle change l'adresse IP du destinataire par la valeur fixe spécifiée

31

Option de NAT

Trames

--in-interface	Interface réseau d'entrée
--out-interface	Interface réseau de sortie

Paquet IP

--source	Adresse IP origine du paquet
--destination	Adresse IP de destination
--protocol	tcp, udp, icmp ou all

32

Option de NAT (suites)

segment TCP ou datagramme UDP

--source-port	port de la source du segment
--destination-port	port de la destination du segment
--state	[« ajouter module state » : -m state]

NEW : ouverture de connexion
ESTABLISHED : dans conversation déjà établie
RELATED : nouvelle connexion liée à une connexion déjà établie

33

Exemple de mascarade

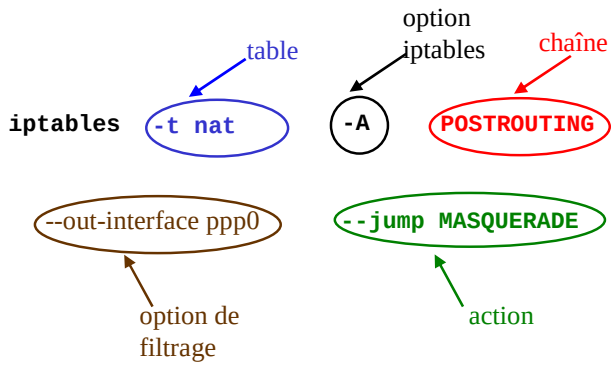
```
iptables -t nat –F

iptables -t nat -A POSTROUTING
 --out-interface ppp0 --jump MASQUERADE

iptables -t nat -A POSTROUTING
 --out-interface ppp0 --jump SNAT
 --to-source   xxx.xxx.xxx.xxx
// xxx.xxx.xxx.xxx est l'adresse IP
   remplaçant l'adresse IP de
   l'émetteur
```

34

Format des règles NAT



35

Enregistre les règles de iptables sous Unix

- Sauver les règles avec **iptables-save** :
  iptables-save >/etc/iptables.rules
- Restaurer les règles avec iptables-restore :
  iptables-restore < /etc/iptables.rules

36

Usage des règles de manière permanente (sous Linux Debian)

- Dans `/etc/network/interfaces`

```
iface eth0 inet static
        address x.x.x.x
        netmask 255.255.0.0
        network x.x.0.0
        broadcast x.x.255.255
pre-up iptables-restore < /etc/firewall
```

37

Usage des règles de manière permanente (sous Linux Debian)

- Dans `/etc/network/interfaces`

```
iface eth0 inet dhcp
        [.. option ..]
pre-up iptables-restore < /etc/firewall
```

38

Serveur de Proximité

Synonymes :
Proxy, Mandataire

39

Serveur de Proximité

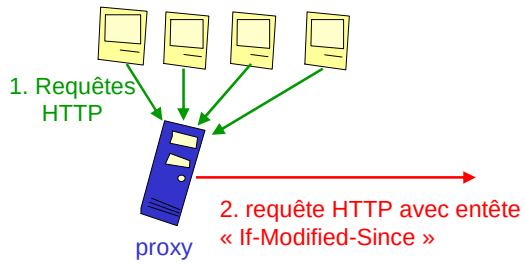
- Dispositif informatique
  - Une application sur un poste qui prend en charge certaines fonctions applicatives à la place d'un ensemble de postes

Niveau OSI

• Mandataire -Proxy	applicatif
• Pare-feu - Firewall	Transport
	Réseau

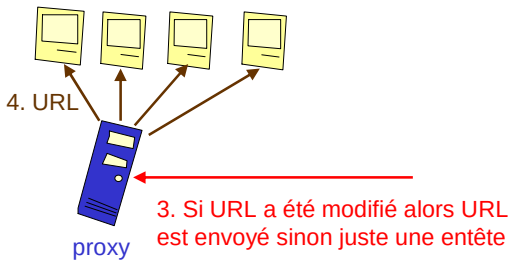
40

Fonctionnement d'un serveur de proximité WEB



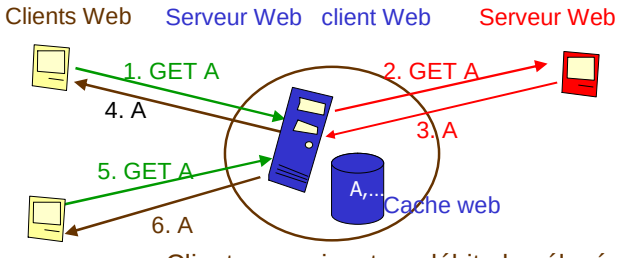
41

Fonctionnement d'un serveur de proximité WEB



42

économiseur de ressources réseaux



Clients perçoivent un débit plus élevé

Diminution du trafic Web

Plus de services

43

Exemple – visualisation des paquet IP

- No.Source Destination Protocol Info
- 4 192.168.0.10 11.169.0.253 HTTP GET …

La requête a été faite au proxy et non pas à la cible. Le proxy transmet la requête à la cible.

- 11 11.169.0.253 42.16.0.251 HTTP GET …

La cible répond au proxy :

- 13 42.16.0.251 11.169.0.253 HTTP … 200

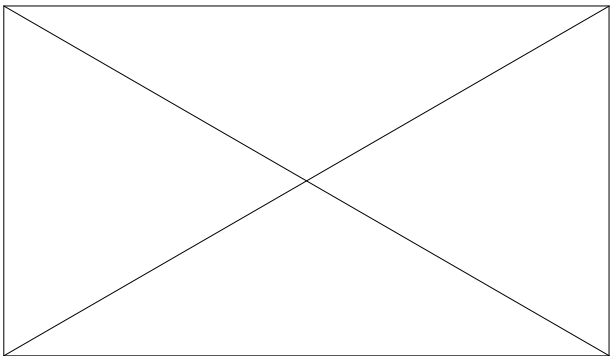
qui retransmet au client:

- 15 11.169.0.253 192.168.0.10 HTTP … 200

et ainsi de suite...

44

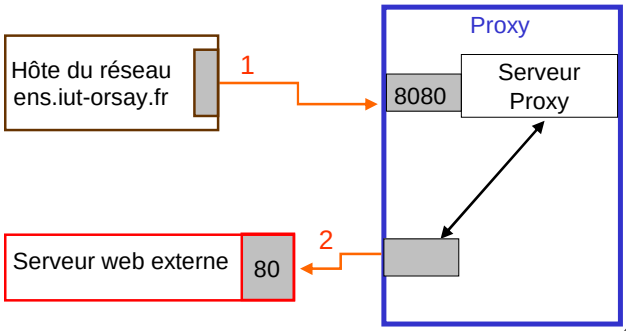
Rôle d'un serveur de proximités



45

A l'IUT

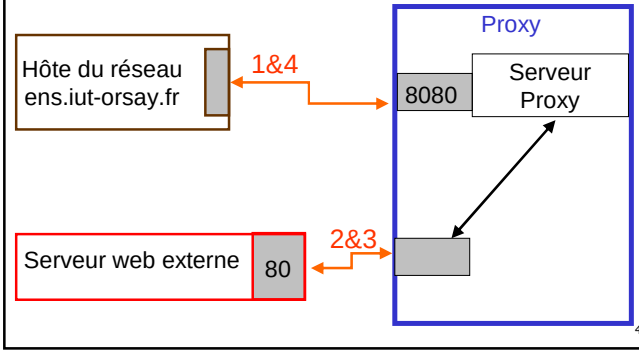
Serveur de proximité WEB : cache.ens.iut-orsay.fr



46

A l'IUT

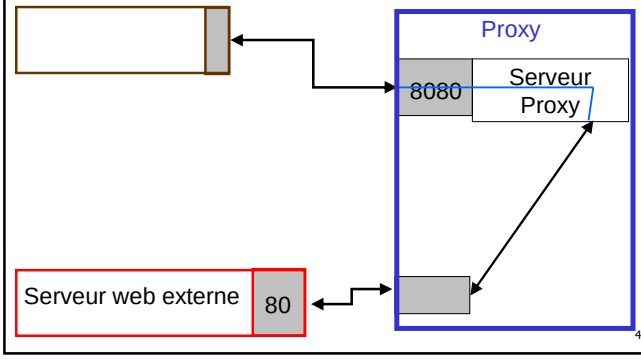
Serveur de proximité WEB : cache.ens.iut-orsay.fr



47

A l'IUT

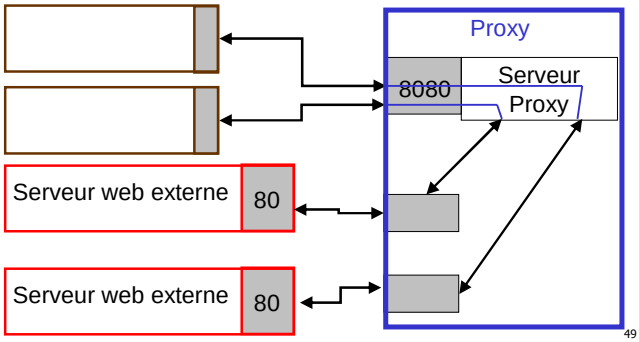
Serveur de proximité WEB : cache.ens.iut-orsay.fr



48

A l'IUT

Serveur de proximité WEB : cache.ens.iut-orsay.fr



49

Rôle d'un serveur de proximité

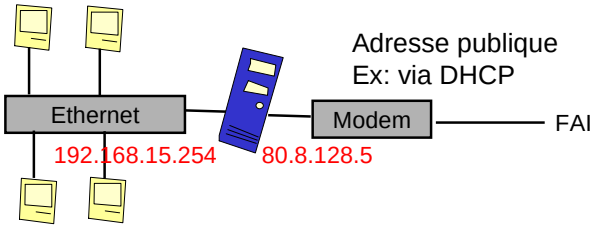
- Economiser les ressources réseaux - cache
- Enregistre les communications
- Sécuriser le réseau local (Filtre)
- Partager une connexion à Internet derrière un serveur de proximité, s'il y a un réseau privé
  - serveur de proximité remplacer un routeur à translation d'adresse

50

Réseau Domestique

Réseau privé = adresses IP privées
Ex: 192.168.15.0/24

Eviter le mascarade pour les services FTP/HTTP/webmail



Adresse publique
Ex: via DHCP

51

Logiciels et protocole

Logiciels

- Des modules « proxy » pour certains serveurs Web (Apache)
- Squid is.a full-featured Web proxy cache, Unix open-source

Protocoles

- SOCKS est un protocole « proxy » générique pour les applications communicantes [RFC 1928]

52

Architecture

53

Structuration

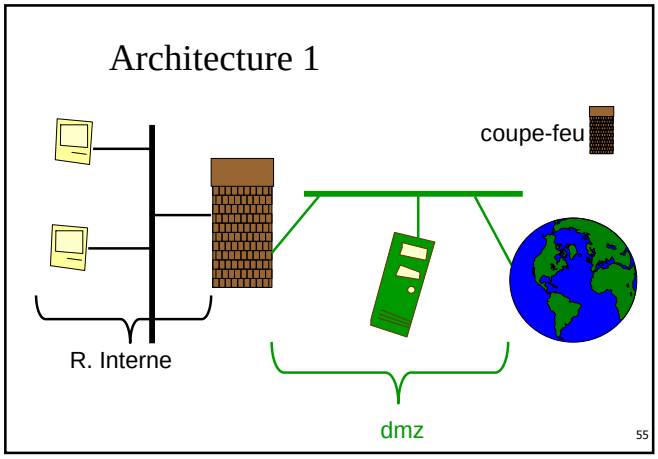
- 3 types de zones :
  - Réseau interne (les hôtes)

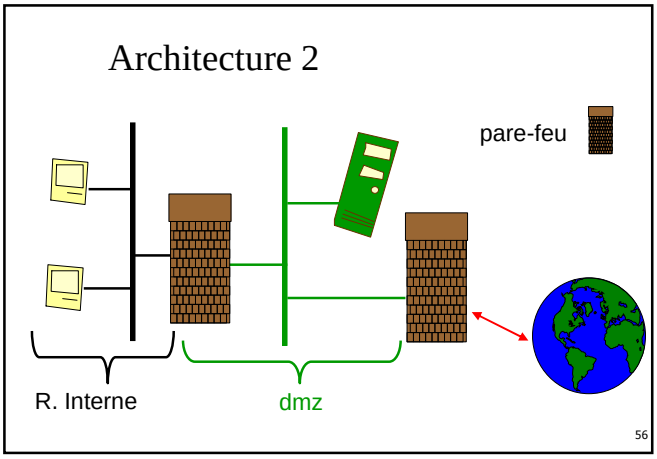
  Une zone démilitarisée - DMZ - demilitarized zone – serveurs du réseau interne qui doivent être accessibles de l'extérieur (DNS, SMTP, HTTP, FTP, News)

  Réseau Externe (Internet)

Pas d'accès directe de l'extérieur vers l'intérieur, intérieur vers extérieur ???

54


Architecture 1
coupe-feu
R. Interne
dmz
55


Architecture 2
pare-feu
R. Interne
dmz
56