Pare-feux ou firewalls

Module R4 – semestre 2 – RT1

IUT de La Rochelle

Université de La Rochelle

romain.raveaux01@univ-lr.fr

Plan

- Introduction
- Fonctionnement
- Types de pare-feu
- Exemples de pare-feu
  - pare-feu avec le NetFilter **`iptables`**
  - pare-feu avec ACL Cisco

Les types d’attaques

- **Accès physique** : il s'agit d'un cas où l'attaquant à accès aux locaux, éventuellement même aux machines :
  - Ecoute du trafic sur le réseau
- **Interception de communications** :
  - Vol de session (*session hijacking*) , Usurpation d'identité , Détournement ou altération de messages
- **Dénis de service** : il s'agit d'attaques visant à perturber le bon fonctionnement d'un service. On distingue habituellement les types de déni de service suivant :
  - Exploitation de faiblesses des protocoles TCP/IP
  - Exploitation de vulnérabilité des logiciels serveurs

Types d’attaques (2)

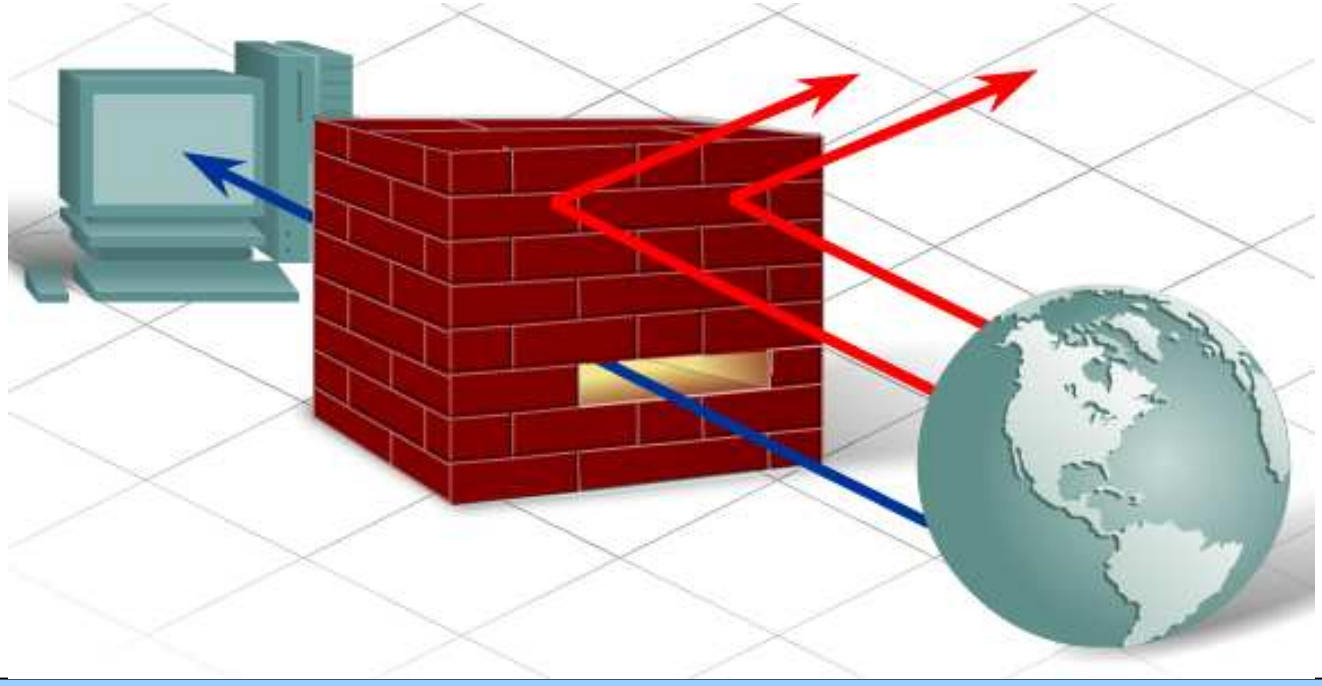
- **Intrusions** :
  - Balayage de ports
  - Elévation de privilèges : ce type d'attaque consiste à exploiter une vulnérabilité d'une application en envoyant une requête spécifique, non prévue par son concepteur, ayant pour effet un comportement anormal conduisant parfois à un accès au système avec les droits de l'application. Les attaques par **débordement de tampon** (en anglais *buffer overflow*) utilisent ce principe.
  - Maliciels (virus, vers et chevaux de Troie)
- **Ingénierie sociale** : Dans la majeure partie des cas le maillon faible est l'utilisateur lui-même !
- **Trappes** : il s'agit d'une porte dérobée (en anglais *backdoor*) dissimulée dans un logiciel, permettant un accès ultérieur à son concepteur.

Protéger un réseau pourquoi ?

- Pour protéger des données sensibles.
- Confidentialité.
- La loi.
  - Laisser sa « box » ouverte vous rend responsable pénalement.

Introduction

- Firewall, pare-feu ou encore garde-barrière
  - structure (logicielle ou matérielle) située entre l'utilisateur et le monde extérieur afin de protéger le réseau interne des intrus.



Introduction (suite)

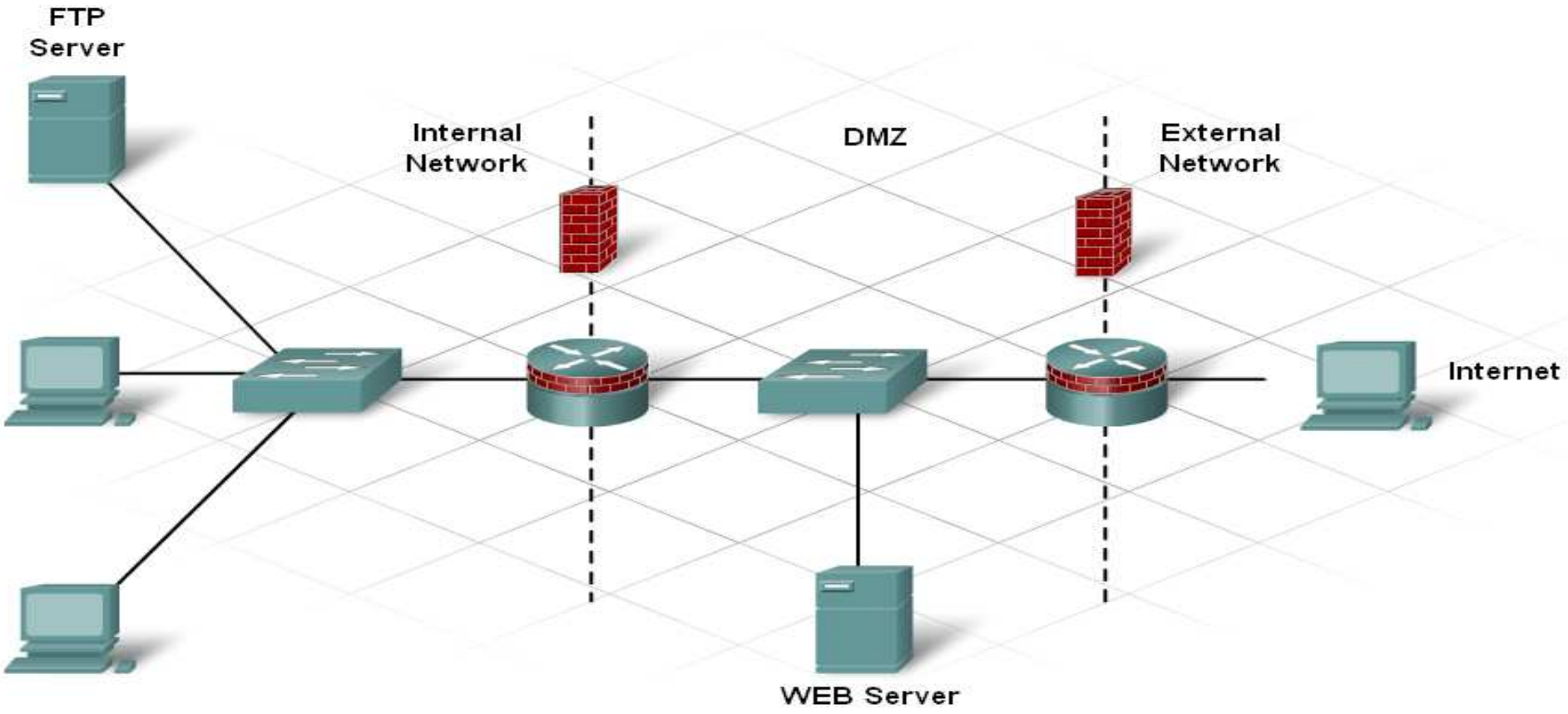
- Rôles d’un pare-feu :
  - déterminer le type de trafic qui sera acheminé ou bloqué
  - limiter le trafic réseau et accroître les performances
  - contrôler le flux de trafic
  - fournir un niveau de sécurité d’accès réseau de base
  - autoriser un administrateur à contrôler les zones auxquelles un client peut accéder sur un réseau
  - filtrer certains hôtes afin de leur accorder ou de leur refuser l’accès à une section de réseau
  - translation d’adresses ou de ports (connexion et protection des réseaux à adressage privé)

Introduction (suite)

- Architecture du réseau avec des pare-feux
  - Un réseau privé : des clients et des serveurs inaccessibles depuis l'Internet.
    - normalement, aucune connexion TCP, aucun échange UDP ne peuvent être initiés depuis le Net vers cette zone.
  - Une « DMZ » (DeMilitarized Zone) : contient des serveurs accessibles depuis le Net et depuis le réseau privé.
  - Les réseaux sont séparés par des pare-feux

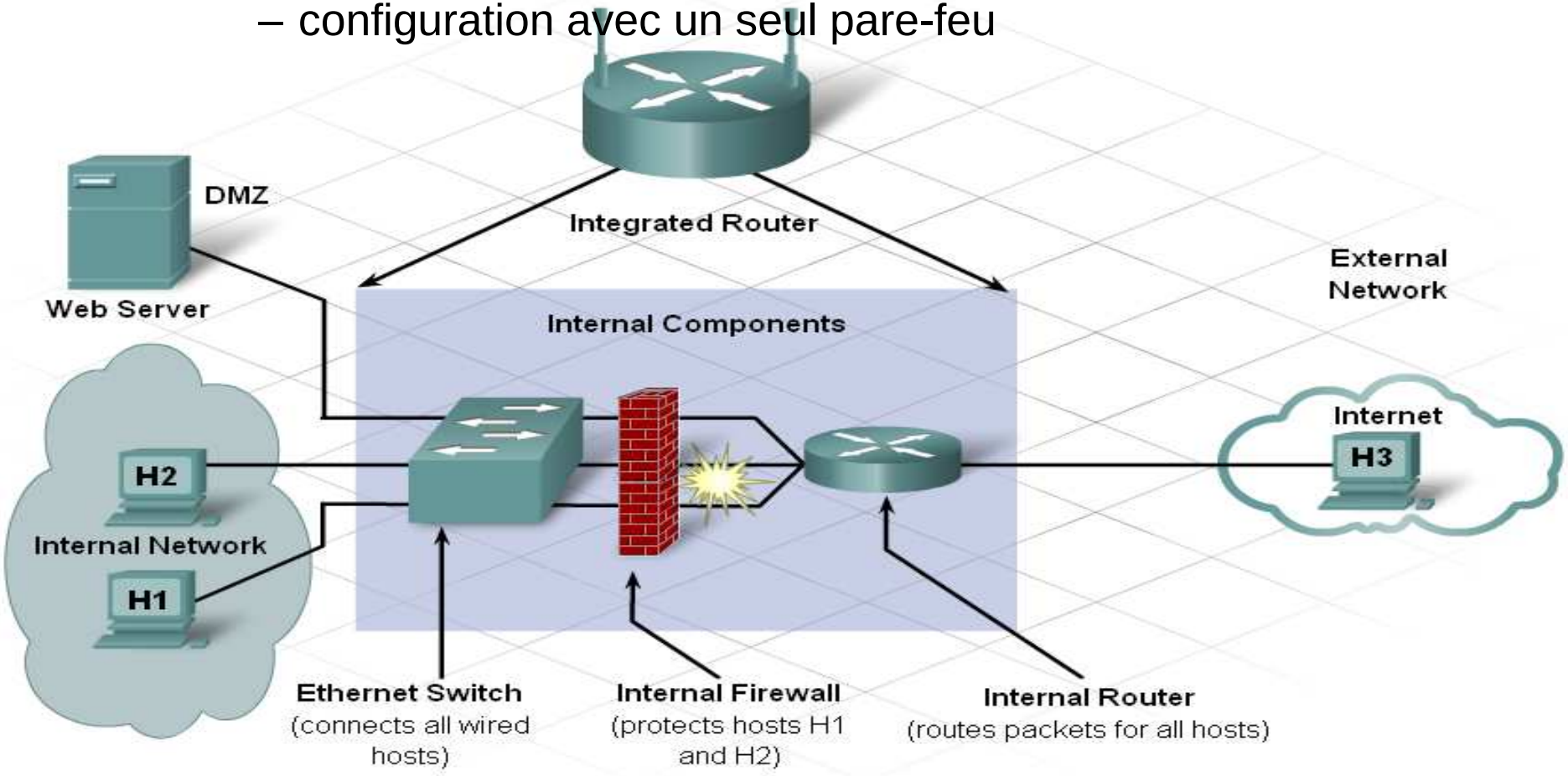
Introduction (suite)

- Pare-feu à séparation de réseau
  - configuration avec deux pare-feux



Introduction (suite)

- Pare-feu au fil de l'eau
  - configuration avec un seul pare-feu



Fonctionnement

- Communications possibles avec un pare-feu
  - Entre le réseau privé et l'extérieur
    - les utilisateurs internes initialisent des communications avec l'extérieur
    - seules les réponses à ces requêtes doivent entrer dans cette zone
    - cette zone est construite sur une classe d'adresses privées et nécessite donc une translation d'adresse (NAT) ou de port (PAT) pour accéder à l'extérieur
  - Entre la DMZ et l'extérieur
    - serveurs qui doivent être accessibles depuis l'extérieur
    - il faudra permettre de laisser passer des connexions initiées depuis l'extérieur vers ces serveurs uniquement (*à surveiller*)

Fonctionnement (suite)

- Communications possibles avec un pare-feu (suite)
  - Entre le réseau privé et la DMZ
    - les accès devraient être à peu près du même type qu'entre la zone privée et l’extérieur
    - avec un peu plus de souplesse pour mettre à jour les serveurs web, d’envoyer et recevoir les messages (puisque le SMTP est dedans)
    - en revanche, depuis la DMZ, il ne devrait y avoir aucune raison pour qu'une connexion soit initiée vers la zone privée

Types de pare-feux

- Pare-feu par filtrage simple de paquets ("stateless" ou sans état) :
  - vérifie si les "sockets" (couples @IP - numéro port) source et destination sont conformes ou non à des autorisations de passage
- Pare-feu par suivi de connexion ("statefull" ou à états) :
  - conserve les états des connexions : 4 types d'états
    - NEW : un client envoie sa première requête vers un serveur web.
    - ESTABLISHED : connexion a déjà été initiée (après un NEW).
    - RELATED : ce peut être une nouvelle connexion, mais elle présente un rapport direct avec une connexion déjà connue.
    - INVALID : un paquet qui n'a rien à faire là dedans.

Types de pare-feux (suite)

- Pare-feux applicatifs :
  - ici, on est au niveau application (HTTP, FTP…)
  - un tel pare-feu est en réalité un serveur "proxy" (appelé aussi serveur mandataire ou bastion)
    - le client s'adressera toujours à lui, quelle que soit la cible finale et n'acceptera de réponse que de sa part
    - le proxy reformule la requête pour son propre compte vers l’extérieur
    - lorsque le proxy reçoit la réponse, il la transmet au client comme si c'était lui qui répondait directement

Types de pare-feux (suite)

- Avantages et inconvénients
  - le pare-feu "stateless"
    - simple à utiliser, consomme très peu de ressources, mais ne permet pas une grande finesse de filtrage évolué
  - le pare-feu "statefull"
    - souple à paramétrer, consomme peu de ressources, mais le suivi des connexions est délicat
  - le proxy
    - barrière effective entre l’extérieur et la zone protégée, sauf que ces logiciels sont complexes et peuvent contenir des bugs, consomment des ressources et nécessitent machines dédiées
- Conclusion : personne n’est parfait

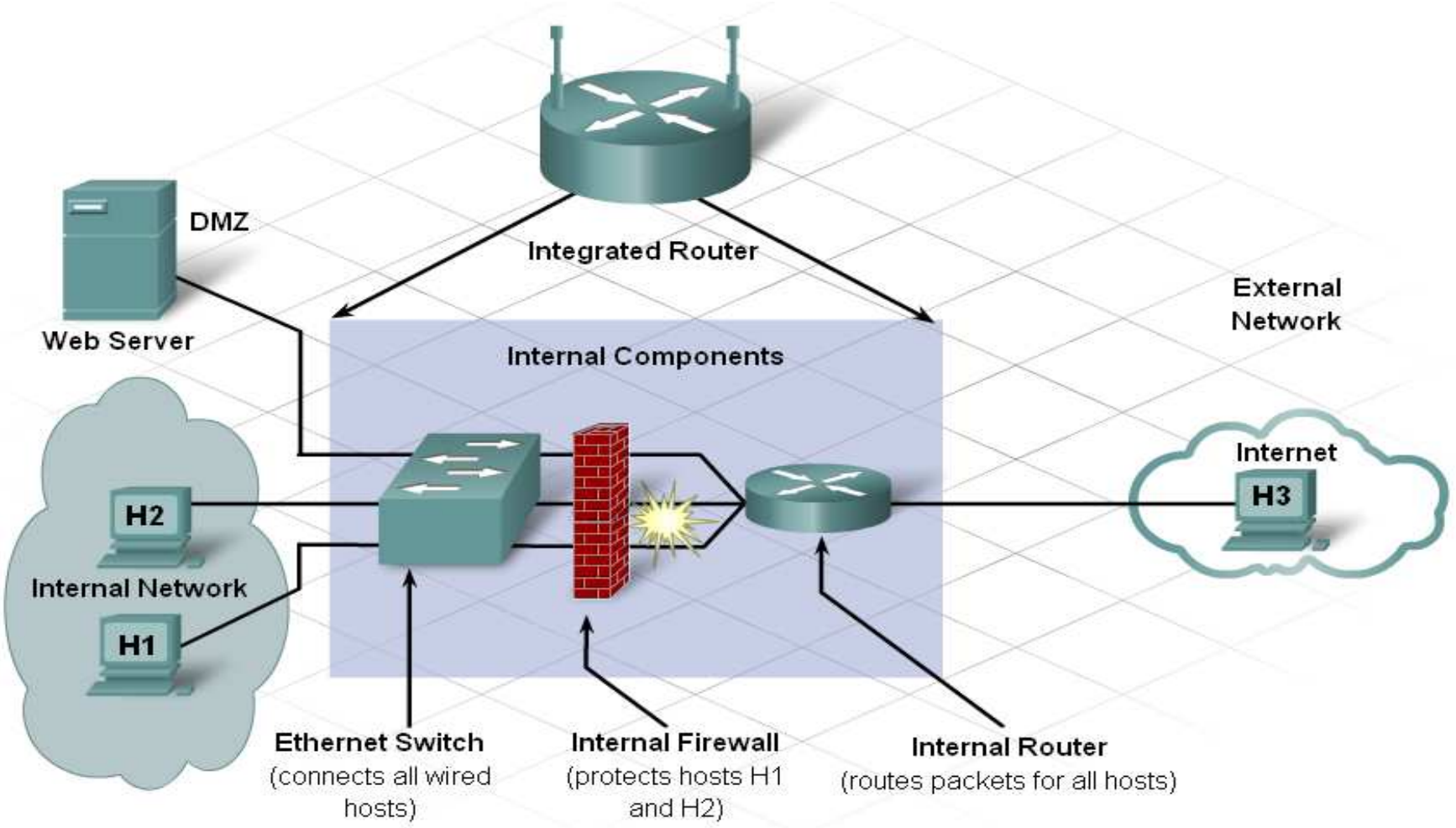
Recommandations

- Pour obtenir une sécurité maximale (mais jamais absolue) il faut travailler à tous les niveaux
  - configurer ses serveurs avec le plus grand soin, en éliminant tous les risques connus à leur niveau
  - protéger le tout avec un système de filtrage efficace, bien adapté à ses besoins
  - surveiller avec le plus grand soin tout se qui se passe, aussi bien sur les serveurs que sur les filtres, pour détecter le plus vite possible toute activité anormale
  - avoir toujours à l'esprit qu'il y a forcément quelque part une faille que l'on ne connaît pas, et qu'un pirate peut trouver
- Conclusion : personne n’est parfait

Exemple de Firewall

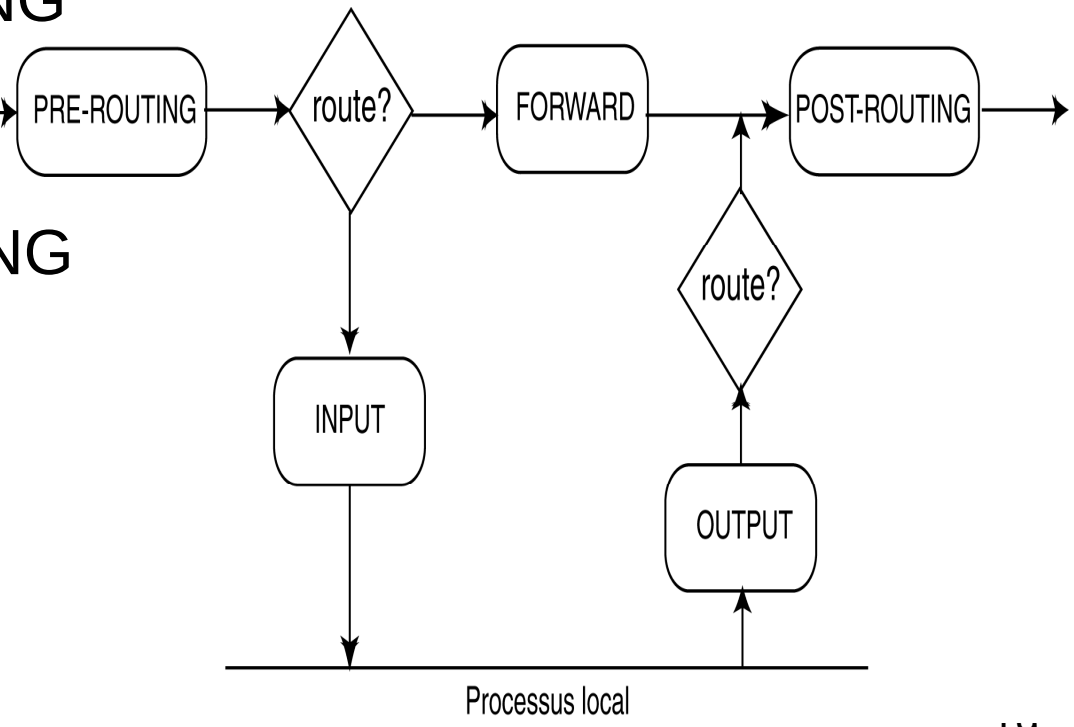
NetFilter `iptables` sous Linux

Rappel


DMZ
Web Server
Integrated Router
External
Network
Internal Components
Internet
H3
H2
Internal Network
H1
Ethernet Switch
(connects all wired
hosts)
Internal Firewall
(protects hosts H1
and H2)
Internal Router
(routes packets for all hosts)

Architecture de NetFilter

- Les hooks (ou points d'accrochage) définissent des points sur lesquels des modules de traitement des paquets vont se greffer
  - NF_IP_PRE_ROUTING
  - NF_IP_LOCAL_IN
  - NF_IP_FORWARD
  - NF_IP_POSTROUTING
  - NF_IP_LOCAL_OUT



Architecture de NetFilter (suite)

- À travers ces points d'insertion, Netfilter va pouvoir
  - effectuer des filtrages de paquets, principalement pour assurer des fonctions de pare-feu
  - effectuer des opérations de NAT ou PAT (Network/Port Address Translation)
  - effectuer des opérations de marquage des paquets, pour leur appliquer un traitement spécial.
- Il y a dans Netfilter trois tables qui correspondent aux trois principales fonctions vues plus haut
  - table « filter » : filtrage des datagrammes
  - table « NAT » : translation d’adresses (ou masquerading)
  - table « mangle » : marquage des datagrammes (en modifiant leur qualité de service)

Architecture de NetFilter (suite)

- Il y a dans NetFilter 5 chaînes regroupant les règles contenues dans les tables et identifiant les paquets qui correspondent à certains critères
  - chaîne INPUT
    - décide du sort des paquets entrant localement sur l'hôte
  - chaîne OUTPUT
    - décide du sort des paquets émis localement par l'hôte
  - chaîne FORWARD
    - décide du sort des paquets qui traversent l'hôte
  - chaîne PREROUTING
    - décide du sort des paquets qui arrive sur l'hôte
  - chaîne POSTROUTING
    - décide du sort des paquets qui sortent du hôte.

Fonctionnement de la table *filter*

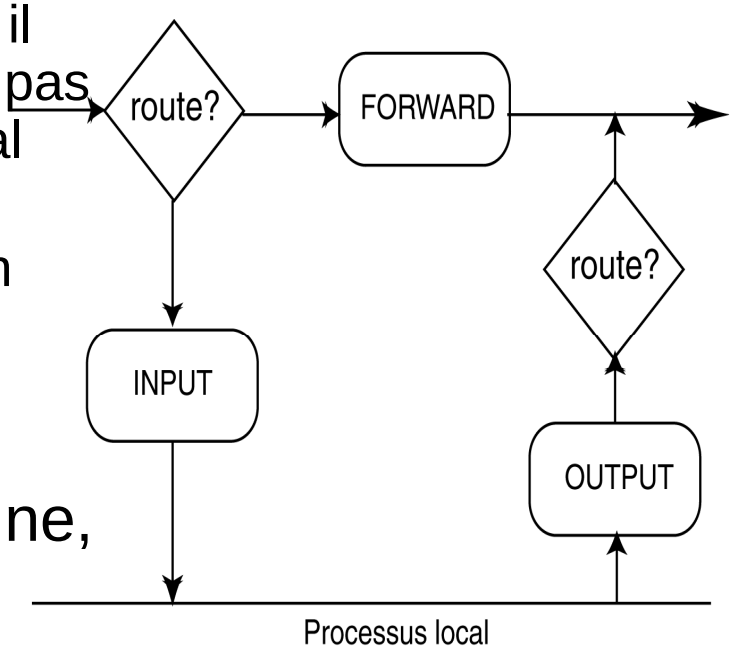
- Cette table contient toutes les règles qui permettront de filtrer les paquets
  - il n'y a ici aucune modification de ces paquets, ils seront comparés à des critères définis dans la table
- Elle contient 3 chaînes
  - chaîne INPUT
    - qui décide du sort des paquets entrant localement sur l'hôte.
  - chaîne OUTPUT
    - ici, ce ne sont que les paquets émis par l'hôte local qui seront filtrés
  - chaîne FORWARD
    - les paquets qui traversent l'hôte, suivant les routes configurées dans la table de routage, seront filtrés ici

Fonctionnement de la table *filter*

- Les cibles sont des sortes d'aiguillage qui dirigent les paquets satisfaisant aux critères. Les cibles préconstruites sont :
  - ACCEPT
    - les paquets qui satisfont aux critères sont acceptés, ils continuent leur chemin dans la pile
  - DROP
    - les paquets qui satisfont aux critères sont rejetés, on les oublie, on n'envoie même pas de message ICMP
  - LOG
    - c'est une cible particulière qui permet de tracer au moyen de syslog les paquets qui satisfont aux critères
- D'autres cibles deviennent accessibles suivant le contexte : REJECT, RETURN, REDIRECT…

Fonctionnement de la table *filter*

- Processus de filtrage
  - un paquet entrant, passe d'abord par la fonction de décision de routage
    - si le paquet est destiné au hôte local, il traverse la chaîne INPUT et, s'il n'est pas rejeté, est transmis au processus local impliqué.
    - si le paquet est destiné à un hôte d'un autre réseau, il traverse la chaîne FORWARD et s'il n'est pas rejeté, il poursuit alors sa route
  - un paquet envoyé par notre machine, traverse la chaîne OUTPUT et, s'il n'est pas rejeté, va vers la sortie



Fonctionnement de la table *NAT*

- NAT (*Network Address Translation*)
  - traduction (ou substitution) d'adresse au niveau d'un paquet (donc niveau réseau, niveau IP)
- PAT (*Port Address Translation*)
  - traduction (ou substitution) de port au niveau d'un datagramme ou un segment (donc niveau transport, niveau TCP ou UDP)
- NAT et PAT sous NetFilter
  - la table NAT sous NetFilter permet non seulement de faire de la translation stricte d'adresses, mais aussi de la translation de ports et un mélange des deux (NAPT), dont le masquage d'adresse est une forme particulière.

Fonctionnement de la table *NAT*

- Il y a 3 types de translations
  - NAT source (SNAT) : changement de l'adresse/port source d'un paquet
    - utile lorsque l'on souhaite connecter un réseau privé à l'Internet alors que l'on ne dispose que d'un nombre limité d'@IP valides (publiques) sur le Net (même si ces dernières sont dynamiques)
  - masquerade ou masquerading
    - forme particulière du SNAT lorsqu'on dispose d'une seule @IP publique dynamique et qu'on veut que les paquets sortants du réseau privé prennent cette adresse
  - NAT destination (DNAT) : changement d'adresse/port destination d'un paquet
    - utile lorsqu'on veut accéder à partir d'Internet à un serveur local dont l'adresse IP est non routable (redirection vers @IP privée)

Fonctionnement de la table *NAT*

- La table NAT regroupe les règles qui définissent les critères de translation des adresses et des ports
- Elle contient trois chaînes
  - chaîne PREROUTING
    - permet de faire la translation d'adresse de destination. Utilisée par exemple si l'on veut faire croire au monde extérieur qu'il y a un serveur WEB sur le port 80 de la passerelle, alors qu'il est hébergé sur le réseau privé et sur un autre port (tel que 8080)
  - chaîne POSTROUTING
    - permet de faire la translation d'adresse de la source. Connexion d'un réseau privé avec une seule @IP publique à l'Internet
  - chaîne OUTPUT
    - celle-ci va permettre de modifier la destination de paquets générés localement

Fonctionnement de la table *NAT*

- Trois nouvelles actions (cibles) sont possibles sur ces chaînes :
  - SNAT permet, dans la chaîne POSTROUTING, de translater les adresses et/ou les ports source.
  - DNAT permet, dans les chaîne PREROUTING et OUTPUT, de translater les adresses et/ou les ports destination
  - MASQUERADE permet une translation NAT simple et rapide (chaîne POSTROUTING) avec une seule @IP publique dynamique

Utilisation de iptables

- iptables [-t *table*] -[AD] *chaîne règle* [*options*]
- iptables [-t *table*] -I *chaîne* [*numéro-de-règle*] *règle* [*options*]
- iptables [-t *table*] -R *chaîne numéro-de-règle règle* [*options*]
- iptables [-t *table*] -D *chaîne numéro-de-règle* [*options*]
- iptables [-t *table*] -[LFZ] [*chaîne*] [*options*]
- iptables [-t *table*] -N *chaîne*
- iptables [-t *table*] -X [*chaîne*]
- iptables [-t *table*] -P *chaîne cible* [*options*]
- iptables [-t *table*] -E *ancien-nom-de-chaîne nouveau-nom-de-chaîne*

Utilisation de iptables

- utilisation d'une table : iptables -t *table*
  - *table* = filter, nat ou mangle (par défaut c'est « filter »)
- affichage de la table : iptables -t *table* -L
- ajout d'une règle : iptables -t *table* -A
- @source du paquet : -s ou --source *adresse/mask*
  - pour l'@dest du paquet : -d ou --destination
- interface d'entrée : -i ou --in-interface
- interface de sortie : -o ou --out-interface
- spécifier un protocole : -p ou --protocol
  - protocoles acceptés : icmp, tcp et udp
- spécifier un port avec TCP/UDP : --sport ou --dport

Utilisation de iptables

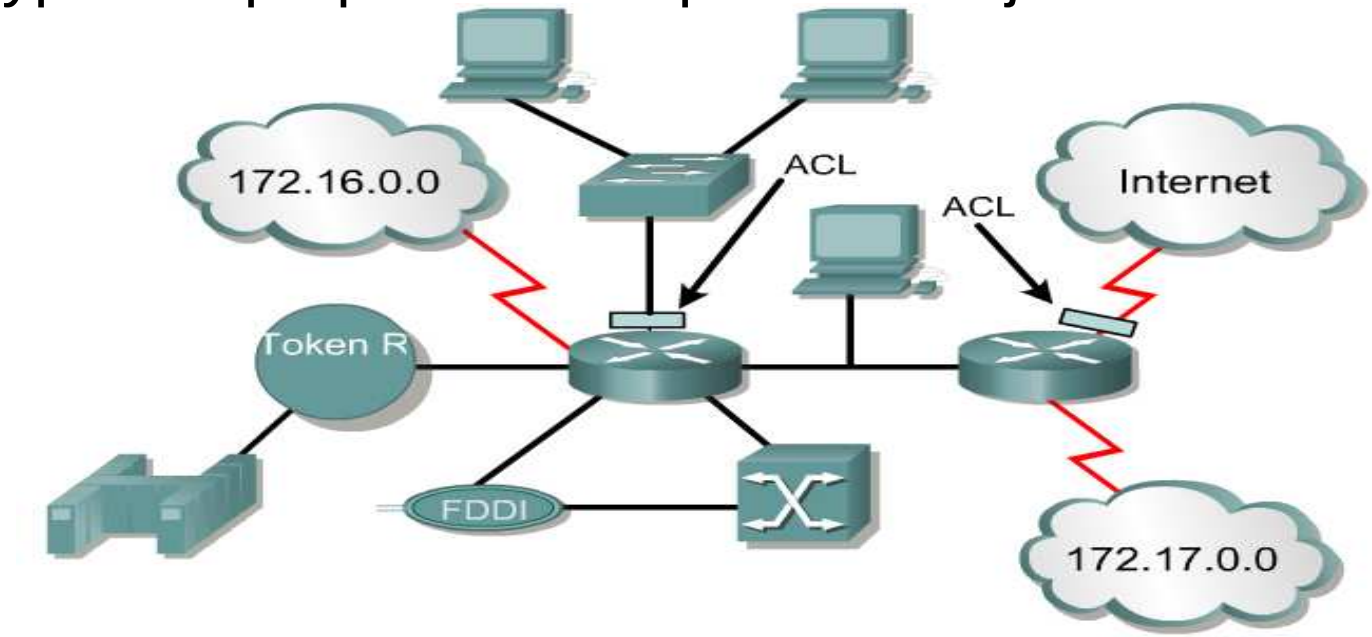
- ajout filtre du trafic TCP entrant sur le port 22
  - iptables -A INPUT -p tcp --dport 22 -j DROP
- insertion en 2e position filtre du trafic traversant issu de @IP A.B.C.D et arrivant sur interface eth0
  - iptables -I FORWARD 2 -i eth0 -s A.B.C.D -j REJECT
- changement des @source en 1.2.3.4
  - iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4
- masquerading de tout ce qui sort par interface ppp0
  - iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE

Exemple de Firewall

Listes de contrôle d’accès
(ACL Cisco)

Définition d’une ACL

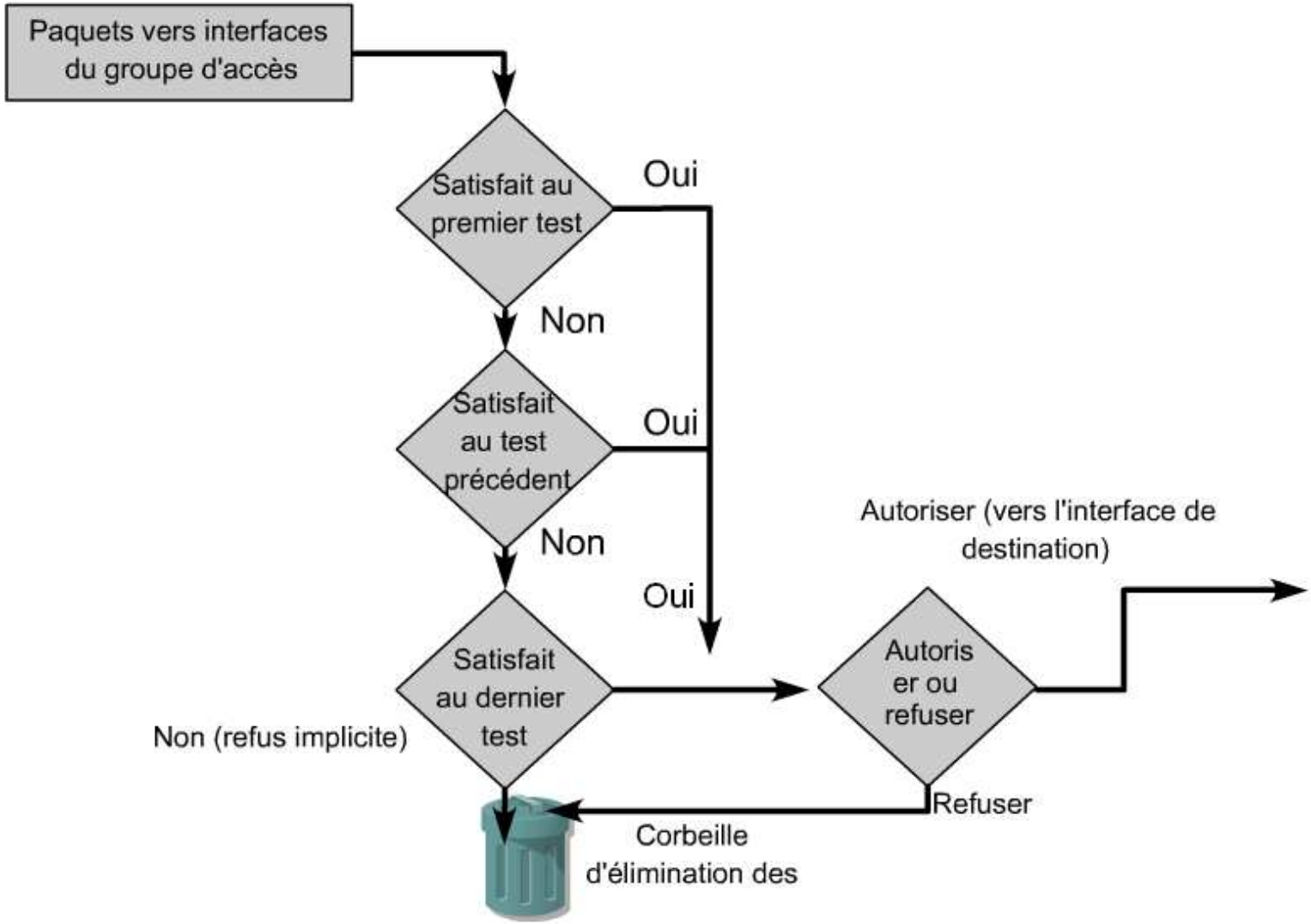
- ACL = bout d’un pare-feu dans un routeur
  - les listes de contrôle d’accès sont des listes de conditions qui sont appliquées au trafic circulant via une interface du routeur. Ces listes indiquent au routeur les types de paquets à accepter ou à rejeter.



Fonctionnement des ACL

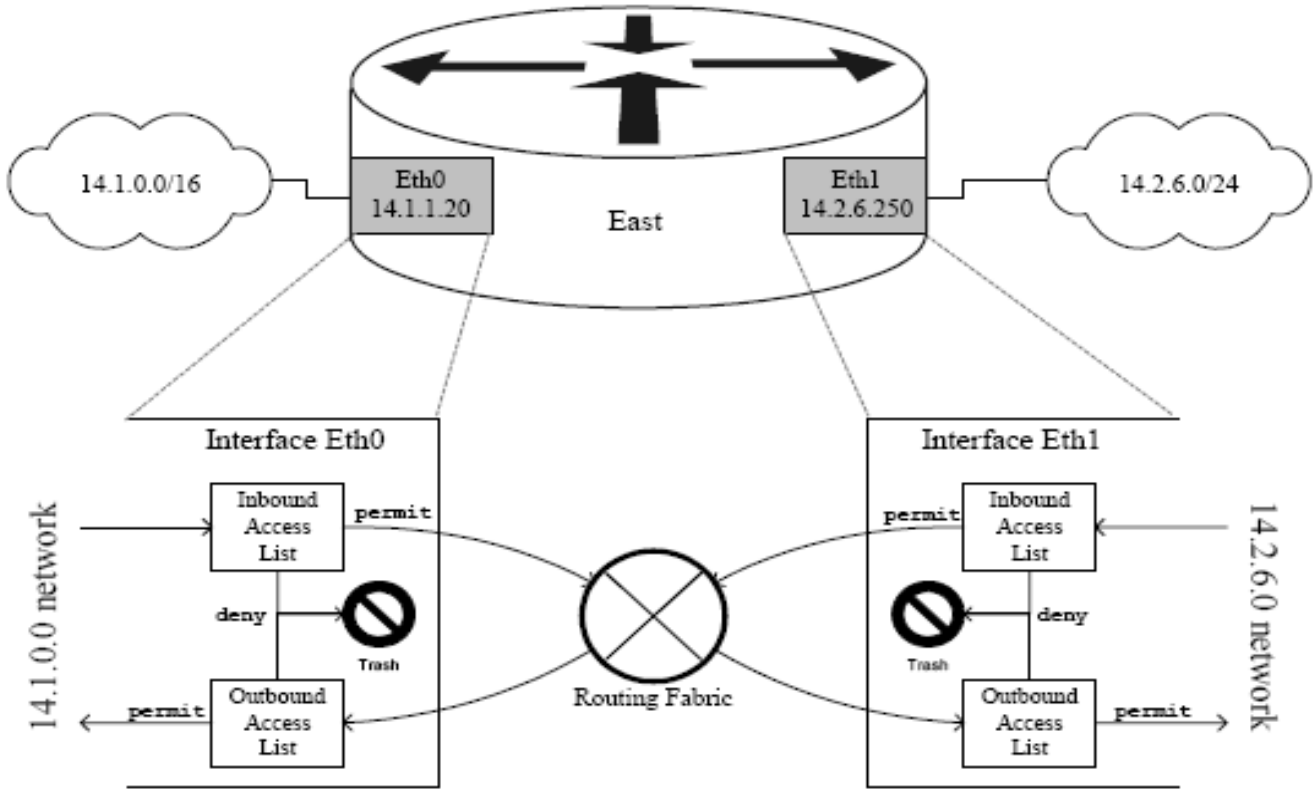
- Une ACL est une suite d'instructions, une par ligne, dont le routeur teste la validité une par une du haut au bas jusqu'à trouver une correspondance, il exécute alors le traitement correspondant (à la iptables sous Unix) et sort
- Liste ACL
  - Condition 1 traitement 1
  - Condition 2 traitement 2
  - Condition 3 traitement 3
  - ……..

Fonctionnement des ACL



Fonctionnement des ACL

- une ACL est appliqué à une interface et dans un sens (inbound/outbound).



Fonctionnement des ACL

- Types d'ACL (chaque ACL a un numéro)
  - ACL IP standard (numéros 1-99, 1399-1999)
    - filtrent l'accès à partir de l'adresse source uniquement
  - ACL IP étendu (numéros 100-199, 2000-2699)
    - selon l'adresse de source et de destination
    - selon les types de protocole de transport (TCP, UDP)
    - et le numéro de port (couche application)
    - messages ICMP et IGMP
    - supporte le « selective logging »
  - autres ACL : AppleTalk, IPX, IPX étendu…

Fonctionnement des ACL

- Notion de masque générique (wildcard mask)
- Il ne faut pas confondre un masque générique avec un masque de réseau
  - masque de réseau = masque de division/regroupement
    - une addition booléenne d'une @IP et d'un masque de réseau est utilisée pour distinguer la partie réseau de la partie hôte
    - un masque de réseau est nécessairement une suite homogène de 1 et puis de 0
  - masque générique = masque de filtrage
    - quand un bit aura une valeur de 0 dans le masque, il y aura vérification de ce bit sur l'adresse IP de référence
    - lorsque le bit aura une valeur de 1, il n'y en aura pas
    - un masque générique peut être une suite quelconque de 1 et de 0 en fonction du filtrage que l'on veut opérer sur des adresses IP

Fonctionnement des ACL

- Exemple de masque générique (wildcard mask)
  - adresse de référence
    - décimal : 10.1.1.0
    - binaire : 00001010.00000001.00000001.00000000
  - masque générique
    - décimal : 0.0.0.255
    - binaire : 00000000.00000000.00000000.11111111
  - les trois premiers octets de l'@IP doivent correspondre, le dernier octet n'a pas d'importance. Ainsi, toutes les adresses de 10.1.1.0 à 10.1.1.255 seront vérifiées
- Cas particulier
  - masque générique = complément du masque réseau
  - par exemple 255.255.255.0 devient 0.0.0.255

Syntaxe

- Créaction d'une ACL
  - (mode config) **access-list** list-number {deny | permit } condition
    - permit autorise la paquet à passer et deny le supprime
    - implicitement, une ACL se termine par un l'instruction «refuse tout» ('deny any').
    - ce qui diffère entre les ACL standard et étendues sont
      - le champ list-number renseigne sur le type de l'ACL et par la suite les valeurs du champ condition
      - le champ condition invoque une information sur un protocole ou une adresse sans aller jusqu'aux infos sur la couche application
- Créaction d'une ACL nommée
  - (mode config) **ip access-list** {standard | extended} name
    - name est utilisé pour référencer une ACL au lieu de son numéro

Syntaxe

- Application d'une ACL sur une interface
  - (mode interface) **ip access-group** {list-number | name} {in | out}
    - in applique la liste en entrée de l'interface
    - out applique la liste en sortie de l'interface
- Détails ACL standard
  - **access-list** list-number {deny | permit} source [source-wildcard] [log]
    - list-number est le numéro de la liste d'accès et peut être n'importe quel chiffre décimal entre 1 et 99.
    - deny refuse l'accès si la condition est vérifiée.
    - permit autorise l'accès si la condition est vérifiée.
    - source est l'adresse IP du réseau ou du host origine du paquet.
    - source-wildcard est le masque générique appliqué à la source
    - log est optionnel et permet de faire de la journalisation lorsque les règles sont appliquées

Syntaxe

- Détails ACL étendue
  - **access-list** list-number {deny | permit} protocol source source-wildcard source-qualifiers destination destination-wildcard destination-qualifiers [log | log-input]
  - list-number : 100 à 199.
  - protocol est le nom du protocole IP (eigrp, gre, icmp, igmp, igrp, ip, ipinip, nos, ospf, tcp ou udp).
  - source/destination est l'@IP du réseau ou le hôte origine/destinataire du paquet (mot clé any est possible)
  - source/destination-wildcard : masque générique appliqué à la source/destination (any est possible)
  - source/destination-qualifiers : détails optionnels sur l'origine/destination du paquet comme les numéros de ports et les autres informations spécifiques au protocole

Syntaxe

- Mots réservés
  - le mot "any" remplace le 0.0.0.0 255.255.255.255, autrement dit toute adresse IP
  - le mot "host" remplace le masque 0.0.0.0, par exemple, 10.1.1.1 0.0.0.0 peut être remplacé par « host 10.1.1.1 »
- Opérateurs (pour les numéros de port)
  - lt (less than)
  - gt (greater than)
  - eq (equal)
  - neq (not equal)
  - range (inclusive range)

Exemples

- Refuse tout accès aux membres du réseau 192.168.128.0 vers toutes les destinations
  - Router (config)# access-list 101 deny ip 192.168.128.0 0.0.0.255 any
  - Router (config)# access-list 101 permit ip any any
- Applique les restrictions d'accès à l'interface E1 vers l'intérieur
  - Router (config)# int e1
  - Router (config-if)# ip access-group 101 in
- Autorise l'accès au Web aux postes désignés (10.1.128.1 à 10.1.128.254)
  - Router (config)# access-list 102 permit tcp 10.1.128.0 0.0.0.255 host 10.1.96.1 eq 80

Exemples

- La mention established autorise les retours pour les applications fonctionnant à double sens, bit Ack à 1
    - permit tcp any eq 80 192.168.10.0 255.255.255.0 gt 1000 established
- Les mentions, lt, gt, eq, neq sont des opérateurs d’autres lignes de commandes logiques utilisés pour autoriser les accès par type d’application.
    - permit tcp host 10.14.72.10 host 192.168.10.3 lt 1024
- La mention range
    - permit tcp host 10.14.72.10 range 1024 1072 host 192.168.10.3 range 1024 1072

Exemples

- Liste nommée
  - Router(config)# ip access-list extended border-filter-14
  - Router (config-ext-nacl)# permit udp any host 14.1.1.2 eq 53
  - Router (config-ext-nacl)# deny udp any any log
  - Router (config-ext-nacl)# deny ip any any log
  - Router (config-ext-nacl)# exit