Sécurité Réseau

INF8085

Architecture des Réseaux

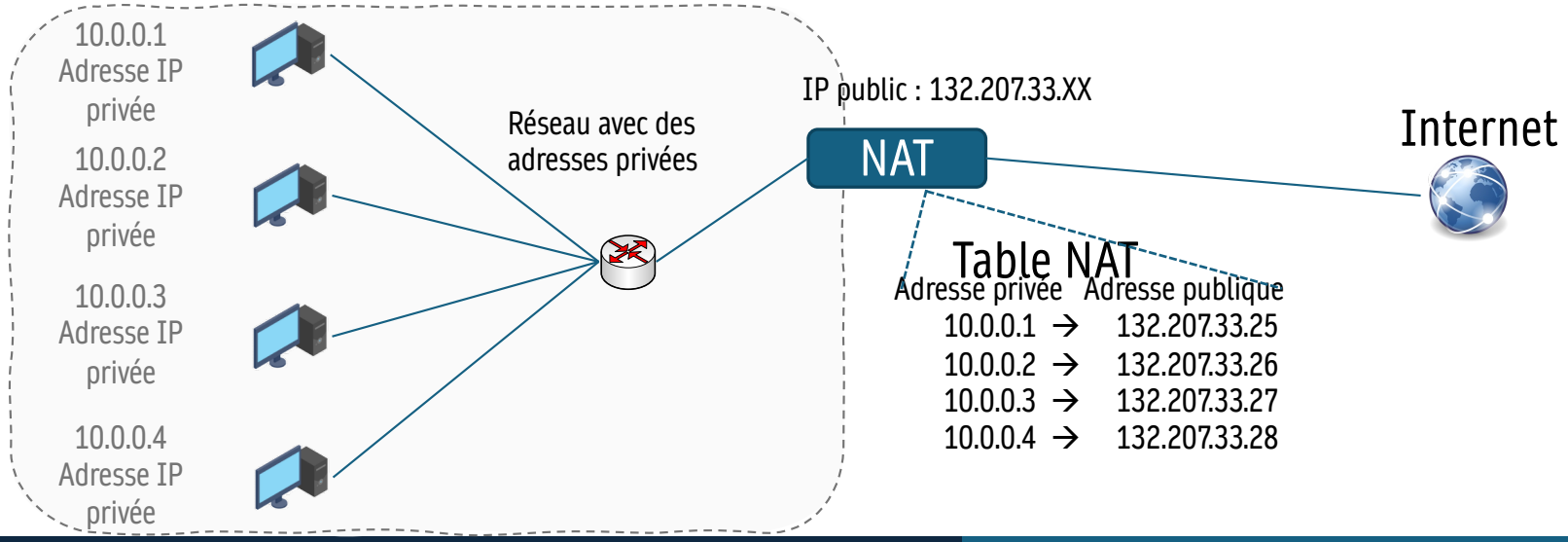
NAT

- NAT – Network Address Translation
- Fonction dans routeur d'accès (entre site et Internet)
- Traduit les adresses IP
  - Modifie l'entête des datagrammes IP échangés avec l'extérieur
  - Dans les sens sortant et entrant
- Une station du site
  - Possède une adresse interne 10.1.1.2
    - Elle est configurée avec cette adresse
    - Les machines internes communiquent avec elle avec cette adresse
  - Connue de l'extérieur avec l'adresse 193.96.49.64 (@ externe)
    - Les machines de l'Internet communiquent avec elle avec cette adresse
- Le système est transparent pour les stations
- Le routeur entre le site et l'Internet fait la traduction

Architecture des Réseaux

NAT
Static

- Dans le NAT statique, à toute adresse IP privée qui communique avec l'extérieur, une adresse IP publique fixe lui est affectée. Avec ce type de NAT, les utilisateurs du réseau local sont joignables de l'extérieur, car la passerelle réalise la correspondance d'une adresse IP locale privée en une adresse IP publique dans les deux sens.



Architecture des Réseaux

NAT

NAT dynamique

- Avec le NAT dynamique, une plage d'adresses IP publiques est disponible et partagée par tous les utilisateurs du réseau local.
- Chaque fois qu'une demande d'un utilisateur local (avec une adresse privée) parvient à la passerelle NAT, celle-ci lui concède dynamiquement une adresse IP publique.
- Les utilisateurs locaux ne sont joignables de l'extérieur que s'ils ont une entrée dans la table de la passerelle NAT.
- Si une machine interne n'a pas d'activité réseau, aucune entrée ne lui est attribué dans la table de NAT.
- L'adresse IP qui leur est affectée est temporaire et peut être différente à la prochaine connexion.

Architecture des Réseaux

NAPT (Network Address Port Translation)

- Il consiste à attribuer une même adresse IP à plusieurs utilisateurs d'un même réseau local.
- Pour associer une même adresse IP publique à deux machines ayant une adresse privée distincte, la passerelle NAT joue sur les ports des applications : une requête envoyée à partir du port A d'une source est retransmise avec le port B de la passerelle, tandis qu'une requête émise à partir du port C d'une autre source est retransmise avec le port D de la passerelle.
- Seuls les utilisateurs du réseau local peuvent commencer une communication vers l'extérieur.
- **Le NAPT est la méthode la plus utilisée puisqu'elle permet de masquer tout un réseau local avec une seule adresse IP.**

Architecture des Réseaux

NAT

- La fonction NAT utilise les ports définis dans TCP et UDP pour associer les paquets entrants dans le réseau local (ayant la même adresse IP publique dans l'entête) vers le bon ordinateur dans le réseau local.
- Dans l'exemple, la passerelle NAT pourrait utiliser les correspondances suivantes.

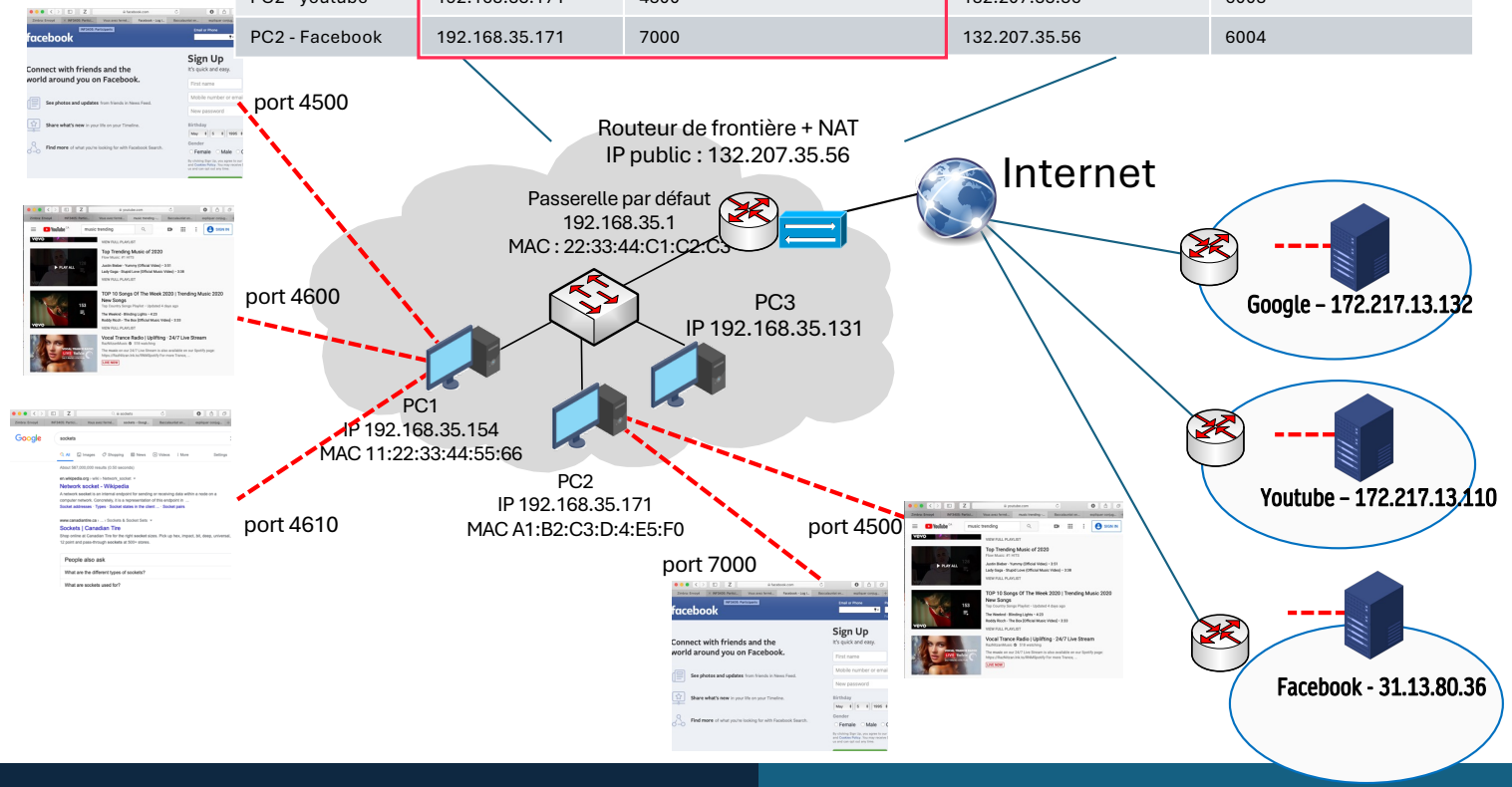
Adresse IP de la source	Port (source et destination)	Nouvelle adresse IP (utilisée dans internet)	Nouveau port (source)
10.0.0.2	80	157.55.0.1	2000
10.0.0.3	20	157.55.0.1	2001
10.0.0.2	23	157.55.0.1	2002

Architecture des Réseaux

NAT dynamique

Réseau intérieur - Réseau extérieur

Connexion logique		Port source machine	IP public	Port source-NAT
PC1 - Facebook	192.168.35.154	4500	132.207.35.56	6000
PC1 - youtube	192.168.35.154	4600	132.207.35.56	6001
PC1 - google	192.168.35.154	4610	132.207.35.56	6002
PC2 - youtube	192.168.35.171	4500	132.207.35.56	6003
PC2 - Facebook	192.168.35.171	7000	132.207.35.56	6004

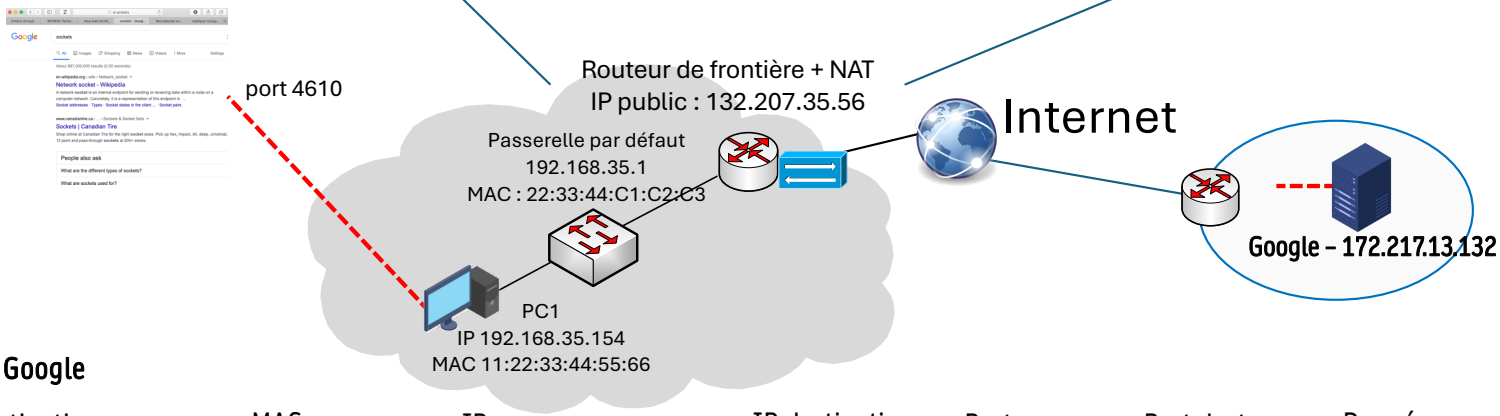


Architecture des Réseaux

NAT dynamique

Réseau intérieur - Réseau extérieur

Connexion logique		Port source machine	IP public	Port source-NAT
PC1 - Facebook	192.168.35.154	4500	132.207.35.56	6000
PC1 - youtube	192.168.35.154	4600	132.207.35.56	6001
PC1 - google	192.168.35.154	4610	132.207.35.56	6002
PC2 - youtube	192.168.35.171	4500	132.207.35.56	6003
PC2 - Facebook	192.168.35.171	7000	132.207.35.56	6004



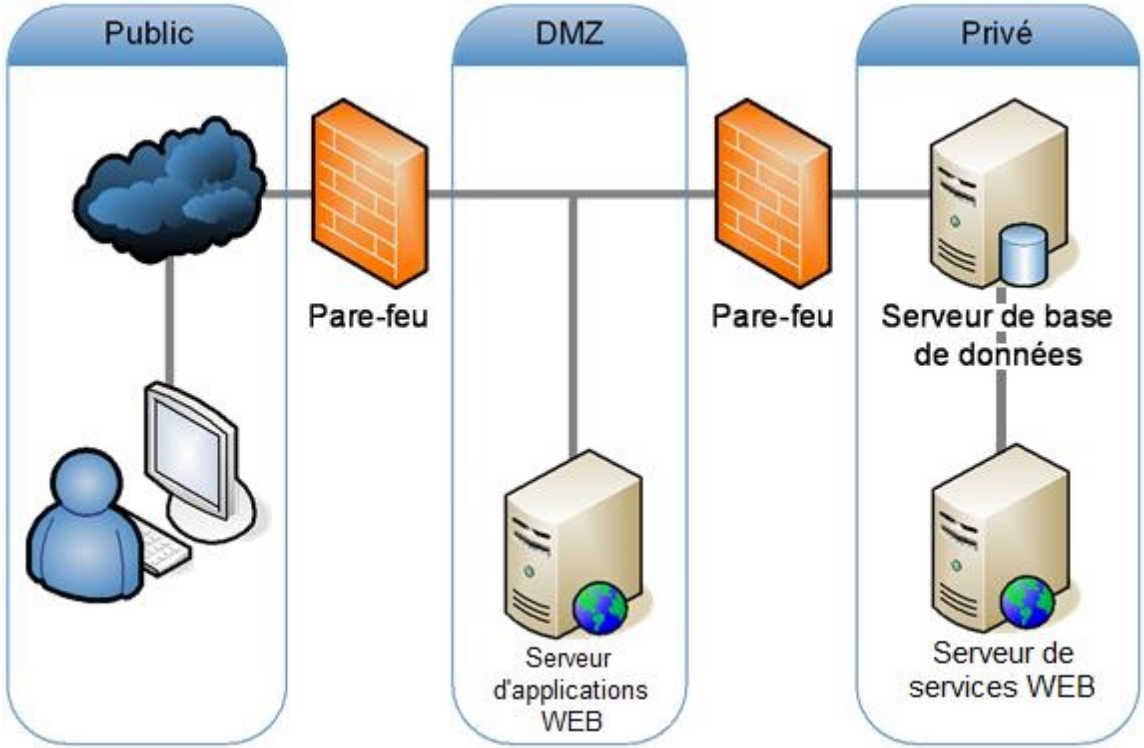
PC1 → Google

	MAC destination	MAC source	IP source	IP destination	Port src.	Port dest.	Données
À l'intérieur du LAN	22:33:44:C1:C2:C3	11:22:33:44:55:66	192.168.35.154	172.217.13.132	4610	80	HTTP+XXXX
Sur l'Internet	Couche Liaison		132.207.35.56	172.217.13.132	6002	80	HTTP+XXXX

Google → PC1

Sur l'Internet	Couche Liaison		172.217.13.132	132.207.35.56	80	6002	HTTP+XXXX
À l'intérieur du LAN	11:22:33:44:55:66	22:33:44:C1:C2:C3	172.217.13.132	192.168.35.154	80	4610	HTTP+XXXX

Note: une fois qu'on se connecte à Google, le serveur utilise un autre port (différent à 80) pour la communication avec chaque client


Public
DMZ
Privé
Pare-feu
Pare-feu
Serveur de base de données
Serveur d'applications WEB
Serveur de services WEB

De quoi protège un pare-feu?

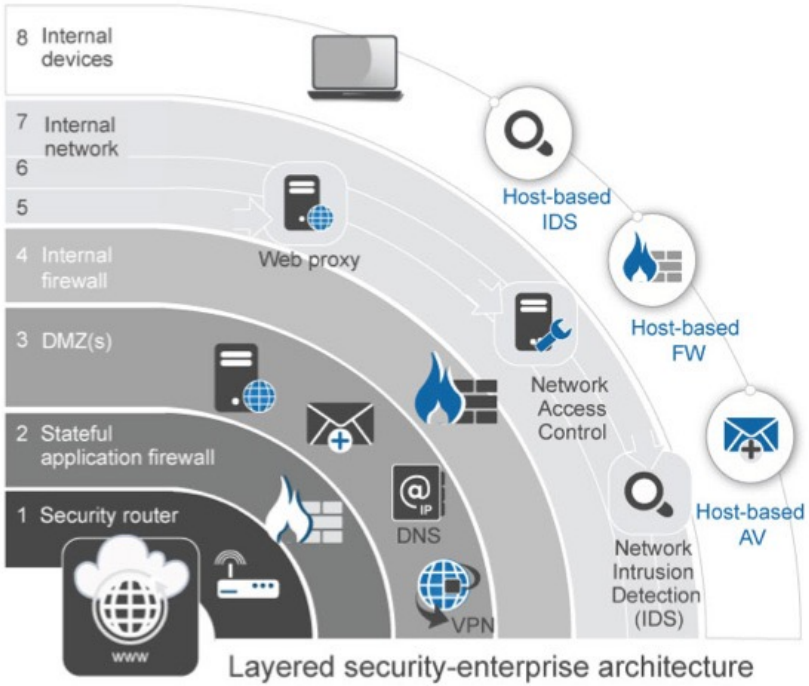
- Protège l'exposition des systèmes sensitives : cache les fonctionnalités de diapositives, du réseau, des systèmes informatiques, …
- Evite l'exploitation de failles des protocoles réseau
- Evite l'accès des utilisateurs non autorisés : méthodes d'authentification, use authentification, autorisation, comptabilité 'accounting' (AAA)
- Protège des données malicieuses

De quoi ne protège pas un pare-feu?

- Les attaques qui ne passent pas par lui (i.e. réseaux sans fil, modem, ...)
- Mauvaise configuration du réseau ou pare-feu
- Application qui ne fonctionne pas avec le pare-feu
- Attaques internes
- Les virus importés par Laptot, PDA, etc.

De quoi ne protège pas un pare-feu?

- Source : Julie Tinnes



Niveaux sécurité

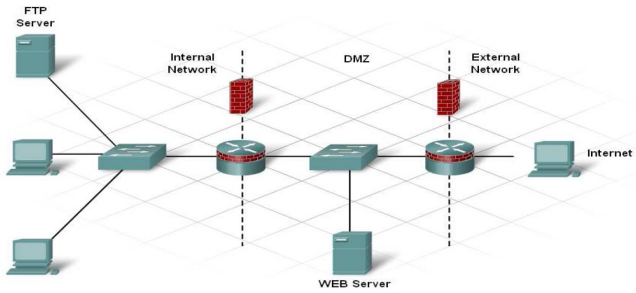
Rôles d’un pare-feu

- 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)

Architecture pare-feu

Architecture du réseau avec des pare-feu

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



DMZ

- Zones intermédiaires dans lesquelles des serveurs réalisent des traitements sur des flux de données

- Crées pour :
  - faciliter la gestion des flux de données au niveau du point de cloisonnement
  - empêcher les flux de données de passer d'une zone sûre à une zone non sûre directement et inversement
  - séparer les grandes fonctionnalités
- Publique, privée, semi-privée

Types de pare-feu

- Filtrage de paquets ‘stateless’
- Pare-feu – filtrage de paquets ‘stateful’
- Pare-feu mandataire –
- Inspection Niveau Application ‘layer Gateway or Proxy’
- Pare-feu transparent
- Prochaine génération (NGF)
- NAT

Types de pare-feu

1. Pare-feu à filtrage de paquets (Packet Filtering Firewall)

- Inspecte-les en-têtes des paquets IP (adresse source/destination, port, protocole).
- Rapide, peu gourmand en ressources.
- Limité : ne comprend pas le contenu applicatif.
- Exemple : ACL (Access Control List) sur routeur Cisco.

2. Pare-feu à inspection avec état (Stateful Firewall)

- Suit l'état des connexions (sessions TCP, UDP, etc.).
- Peut bloquer les paquets qui n'appartiennent pas à une session valide.
- Plus sécurisé qu'un simple filtrage de paquets.
  Exemple : Cisco ASA, pfSense.

3. Pare-feu de couche application (Application-Level Firewall / Proxy)

- Fonctionne comme un proxy : analyse le contenu (HTTP, FTP, DNS, etc.).
- Permet un contrôle très fin (ex : bloquer un site web spécifique).
- Plus lourd (ralentit le trafic).
  Exemple : Squid Proxy, FortiGate avec inspection HTTPS.

Types de pare-feu

4. Pare-feu de nouvelle génération (Next-Generation Firewall – NGFW)

- Combine **stateful + inspection applicative + prévention d'intrusion (IPS/IDS)**.
- Peut identifier les applications (ex : Skype, WhatsApp) même si elles utilisent des ports standards.
- Souvent avec intégration antivirus, sandboxing, contrôle utilisateur.
  Exemple : Palo Alto, Fortinet FortiGate, Check Point.

5. Pare-feu personnel (Host-based Firewall)

- Installé sur un poste ou serveur individuel.
- Protège la machine contre le trafic non autorisé.
  Exemple : Pare-feu Windows Defender, iptables sur Linux.
-

6. Pare-feu basé sur le cloud (Cloud Firewall / FWaaS)

- Hébergé dans le cloud, protège les ressources distantes (VM, applications SaaS).
- Flexible et adapté aux environnements multi-cloud et SD-WAN.
  Exemple : Zscaler, Azure Firewall.

IPTABLES
Le pare-feu historique de Linux basé sur Netfilter

- Un outil en ligne de commande pour créer, lister et gérer des règles de filtrage réseau au niveau du noyau Linux.

- Il agit sur le cadre Netfilter, qui intercepte les paquets aux différents stades de leur parcours

IPTABLES
Le pare-feu historique de Linux basé sur Netfilter

- Concepts clés
  - Tables (finalité) :
    - filter : filtrage (autoriser/bloquer).
    - nat : traduction d'adresses/ports (SNAT, DNAT, MASQUERADE).
    - mangle : modification avancée (TTL, marquage, QoS).
    - raw : exceptions au suivi d'état (conntrack).
  - La table mangle sert à modifier ou marquer les paquets (leurs en-têtes ou leurs métadonnées) pour influencer la QoS, le routage, ou certains détails TCP.
    Elle n'est pas faite pour autoriser/bloquer (ça, c'est filter) ni pour la traduction d'adresses (nat).
  - Chaînes (moment du traitement) :
    - PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING.
  - Règles : testent des critères (IP, port, protocole, état de connexion…) et appliquent une cible (target)
    : ACCEPT, DROP, REJECT, LOG, DNAT, SNAT, etc.
  - Politique par défaut (policy) : action appliquée si aucune règle ne matche (ACCEPT ou DROP).

IPTABLES - exemples

Politique par défaut stricte

- sudo iptables -P INPUT DROP #règle 1
  - Tout trafic entrant destiné à la machine est refusé par défaut (si aucune règle ne l'autorise explicitement).
  - Effet : vous ne recevez pas de connexions entrantes (SSH, HTTP, ping…), sauf si vous ajoutez des règles ACCEPT plus bas.
- sudo iptables -P FORWARD DROP #règle 2
  - La machine ne route pas de paquets à travers elle (mode routeur/bridge) par défaut.
  - Effet : même si l'IP forwarding est activé, rien ne passe d'une interface à l'autre sans règle FORWARD explicite.
- sudo iptables -P OUTPUT ACCEPT #règle 3
  - Tout trafic sortant généré par la machine est autorisé par défaut.
  - Effet : la machine peut initier des connexions (apt, curl, ping, etc.) sans restriction initiale.

IPTABLES - exemples

- sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
  - -A INPUT : ajoute une règle (Append) à la chaîne INPUT → paquets entrant vers la machine.
  - -p tcp : ne concerne que le protocole TCP.
  - --dport 22 : cible le port de destination 22 (service SSH).
  - -j ACCEPT : si le paquet correspond, on l'autorise.

- sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
  - Chaîne INPUT : trafic entrant.
  - TCP uniquement.
  - --dport 80 : HTTP (serveur web non chiffré).
  - ACCEPT : on autorise ces connexions.

- sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
  - Chaîne INPUT : trafic entrant.
  - TCP uniquement.
  - --dport 443 : HTTPS (serveur web chiffré TLS).
  - ACCEPT : on autorise ces connexions.

IPTABLES – Ordre règles

L'ordre des règles dans une chaîne est crucial.

- Les règles d'une chaîne (INPUT, FORWARD, OUTPUT, etc.) sont évaluées de haut en bas.
- La première règle qui matche "termine" le traitement (targets ACCEPT/DROP/REJECT/RETURN).
- -A ajoute en fin ; -I CHAIN N ... insère en position N (utile pour mettre une règle "au-dessus")
- Pour voir les positions :
  - iptables -L INPUT --line-numbers -n -v

ACL pour le routeur de frontière (‘Internet border router’)

```
! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!--- Module 1: Anti-spoofing Denies
!--- These ACEs deny fragments, RFC 1918 space,
!--- invalid source addresses, and spoofs of
!--- internal space (space as an external source).
!
!--- Deny fragments.
access-list 110 deny tcp any 198.133.219.0 0.0.0.255 fragments
access-list 110 deny udp any 198.133.219.0 0.0.0.255 fragments
access-list 110 deny icmp any 198.133.219.0 0.0.0.255 fragments
!--- Deny special-use address sources.
!--- See RFC 3330 for additional special-use addresses.
access-list 110 deny ip host 0.0.0.0 any
access-list 110 deny ip 127.0.0.0 0.255.255.255 any
access-list 110 deny ip 192.0.2.0 0.0.0.255 any
access-list 110 deny ip 224.0.0.0 31.255.255.255 any
!--- Filter RFC 1918 space.
access-list 110 deny ip 10.0.0.0 0.255.255.255 any
access-list 110 deny ip 172.16.0.0 0.15.255.255 any
access-list 110 deny ip 192.168.0.0 0.0.255.255 any
!--- Deny packets spoofing the school’s public addresses
access-list 110 deny ip 198.133.219.0 0.0.0.255 any
'
```

ACL pour le routeur de frontière (‘Internet border router’)

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!--- Module 2:  Explicit Permit
!--- Permit only applications/protocols whose destination
!--- address is part of the infrastructure IP block.
!--- The source of the traffic should be known and authorized.
!
!--- Permit external BGP to peer 64.104.10.113
access-list 110 permit tcp host 64.104.10.114 host 64.104.10.113 eq bgp
access-list 110 permit tcp host 64.104.10.114 eq bgp host 64.104.10.113
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!--- Module 3:  Explicit Deny to Protect Infrastructure
access-list 110 deny ip 64.104.10.0 0.0.0.255 any
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!--- Module 4:  Explicit Permit for Traffic to School’s Public
!--- Subnet.
access-list 110 permit ip any 198.133.219.0 0.0.0.255
!
```

Politique global de Sécurité

1. Tout autoriser
2. Tout interdire
3. Tout autoriser et interdire ce qui est dangereux
   - Déconseillé
4. Tout interdire et autoriser ce qui est utile
   - Bonne stratégie
   - Ce qui n’est pas explicitement permis est interdit

Principes basiques de fonctionnement

- Les règles de filtrage déterminent le devenir des paquets grâce à une politique
- Un paquet entrant dans le pare-feu teste toutes les règles définies jusqu'à en trouver une qui lui corresponde. Et il obéit à la politique spécifiée par la règle trouvée.
- Si aucune règle s'appliquant au paquet n'a été trouvée, alors c'est la politique par défaut qui est utilisée pour savoir que faire de ce paquet.
- Il est donc important, avant de créer les règles, de définir la politique par défaut d'une chaîne.

Principes basiques de fonctionnement : Les politiques

Les politiques de base :

- ACCEPT : le paquet est accepté
- REJECT : le paquet est rejeté avec envoi d'un message d'explication ICMP
- DENY : le paquet est rejeté en mode silencieux
- MASQ : le paquet est redirigé par 'masquerading' (translation d'adresses IP NAT)
- …

Principes basiques de fonctionnement : Les critères de sélection pour les règles

Les règles reposent sur des critères de sélection très variés :

- Machine source
- Port source
- Machine destination
- Port destination
- Interface
- TOS (Type of service)
- Protocole
- Codes spéciaux du paquet TCP
- Types et codes spéciaux du paquet ICMP

Principes basiques de fonctionnement : Actions sur les paquets

Les règles peuvent définir différentes actions sur un paquet :

- Mise en fichier de log
- Application d'une politique
- Redirection vers une autre chaîne
- Marquage du paquet

Principes basiques de fonctionnement : Politique de sécurité

- IP masquerading (NAT)
- Interdire le Ping
- Éviter l'IP Spoofing
- Restreindre Telnet
- Gérer l'accès aux sites web
- Limiter les autres services
- Garder trace du trafic sensible