



POLYTECHNIQUE  
MONTRÉAL

UNIVERSITÉ  
D'INGÉNIERIE

# INF4420: Éléments de Sécurité Informatique

Sécurité des réseaux : Partie 1



# Contenu du cours

- Les réseaux IP (Révision)
- NAT - Network Address Translation (Révision)
- Exemples d'attaques
- Pare-feu
- Exemple de pare-feu : NetFilter / IpTables
- Focus sur le filtrage à états



# Les réseaux IP (révision)

- Les 7 couches du modèle OSI

OSI Model			
	Layer	Protocol data unit (PDU)	Function <sup>[3]</sup>
Host layers	7. Application	Data	High-level APIs, including resource sharing, remote file access
	6. Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5. Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4. Transport	Segment (TCP) / Datagram (UDP)	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3. Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2. Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1. Physical	Bit	Transmission and reception of raw bit streams over a physical medium



# Les réseaux IP (révision)

- Réseaux LAN moderne
  - Couche 1 = 10 Base T,
  - Couche 2 = Ethernet (adresse MAC)
    - Commuté 100%
    - Isolation des ports de chaque commutateur
    - Le commutateur décide où envoyer le paquet en gardant une table des adresses
    - MAC actives sur un port déterminé (lien entre couche 1 et 2)
  - Couche 3 = IP
    - La carte réseau de l'ordinateur ne connaît pas sur quel port se trouve son correspondant
    - Address Resolution Protocol permet au clients et au commutateur de repérer où se trouve une adresse IP sur le réseau via son adresse MAC (lien entre couche 2 et 3)
    - Ces informations sont gardés dans la cache ARP de l'ordi ou du commutateur
    - Le protocole ARP est "stateless"
- Principe de l'attaque ARP Poisoning
  - Inonder le réseaux de fausses réponses au requêtes ARP ...



# Les réseaux IP (révision)

- IP = Internet Protocol
  - Date de création : 1974
  - A Protocol for Packet Network Intercommunication
  - IPv4 : 1978 et RFC en 1981
- Protocole de la couche 3 (réseau)
- Définition du format des paquets
  - Datagramme
  - Entête (Header)
  - Contenu (Payload)
  - Taille maximale d'un paquet :  $2^{16} = 65535$  octets



# Les réseaux IP (révision)

- Adresse IP
  - Adresse globale unique associée à une interface réseau
  - Codée sur 32 bits pour le protocole IPv4
  - Codée sur 128 bits pour le protocole IPv6
- Masque réseau
  - Généralement, on utilise une notation similaire à celle d'une adresse IP
  - Mais ce n'est pas une adresse IP
  - Ou alors, utilisation de la notation /n
  - Exemple :

Classe du réseau	Masque réseau (binaire)	Masque réseau (décimal)
/8 (classe A)	11111111.00000000.00000000.00000000	255.0.0.0
/16 (classe B)	11111111.11111111.00000000.00000000	255.255.0.0
/24 (classe C)	11111111.11111111.11111111.00000000	255.255.255.0



# Les réseaux IP (révision)

- Adresse de réseau
  - Identificateur du réseau suivi de bits à 0
  - Exemples :
  - 125.0.0.0 : Réseau 125 (classe A)
  - 129.15.0.0 : Réseau 129.15 (classe B)
  - 192.168.30.0 : Réseau 192.168.30 (classe C)
- Adresse de diffusion (ou broadcast)
  - Identificateur du réseau suivi de bits à 1
  - Exemples :
  - 125.255.255.255 : Broadcast du réseau 125 (classe A)
  - 129.15.255.255 : Broadcast du réseau 129.15 (classe B)
  - 192.168.30.255 : Broadcast du réseau 192.168.30 (classe C)
- Adresse de machine
  - Exemple :
  - 125.5.6.198 : Machine 5.6.198 du réseau 125



# Les réseaux IP (révision)

- Calcul de l'adresse d'un réseau (ou d'un sous réseau)
  - Soit une adresse de machine et un masque réseau
  - Calculer le « ET » binaire de l'adresse de la machine et du masque réseau
  - Exemple : 125.5.6.198/26
    - $125.5.6.198 = 11111101.00000101.00000110.11000110$
    - $/26 = 11111111.11111111.11111111.11000000$
    - $125.5.6.198/26 = 11111101.00000101.00000110.11000000 = 125.5.6.192$



# Les réseaux IP (révision)

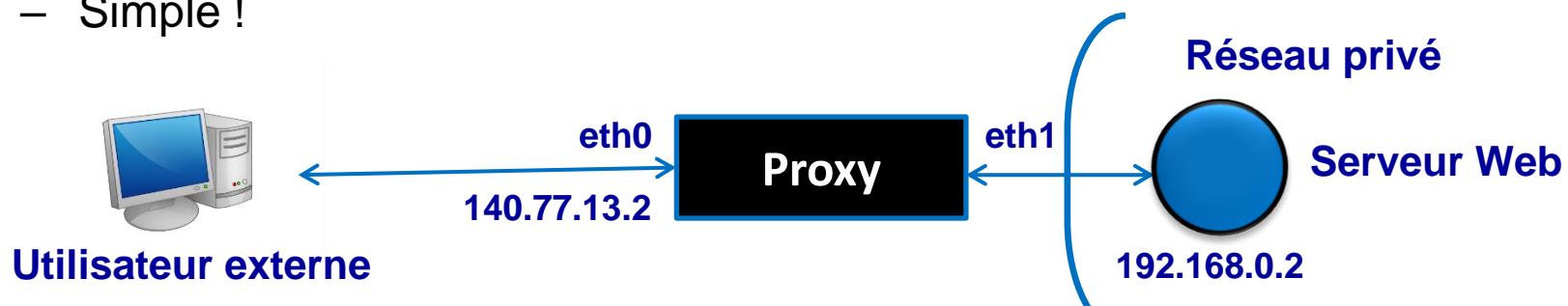
- Le service fourni par IP est minimal
  - Unreliable : pas de garantie de récupération des paquets perdus
  - Connectionless : chaque paquet est géré de façon indépendante
  - Best effort : pas de garantie de qualité de service
- Pénurie d'adresses
  - $2^{32} = 4$  milliards d'adresses
  - Au début d'Internet, c'était énorme !
  - Aujourd'hui, c'est très peu
- Et naturellement pas de sécurité !

# NAT (Network Address Translation)

- Pour communiquer sur Internet, on doit avoir une adresse IP
- Pour répondre au manque d'adresses IP, l'IETF a mis en place les plages d'adresses privées que tous peuvent utiliser
  - 10.0.0.0 /8
  - 172.16.0.0 /12
  - 192.168.0.0 /16
- Les adresses contenues dans ces plages ne sont pas routables sur Internet
- On doit faire une conversion d'adresse NAT (network address translation) pour utiliser ces adresses
  - On obtient de la sécurité en prime

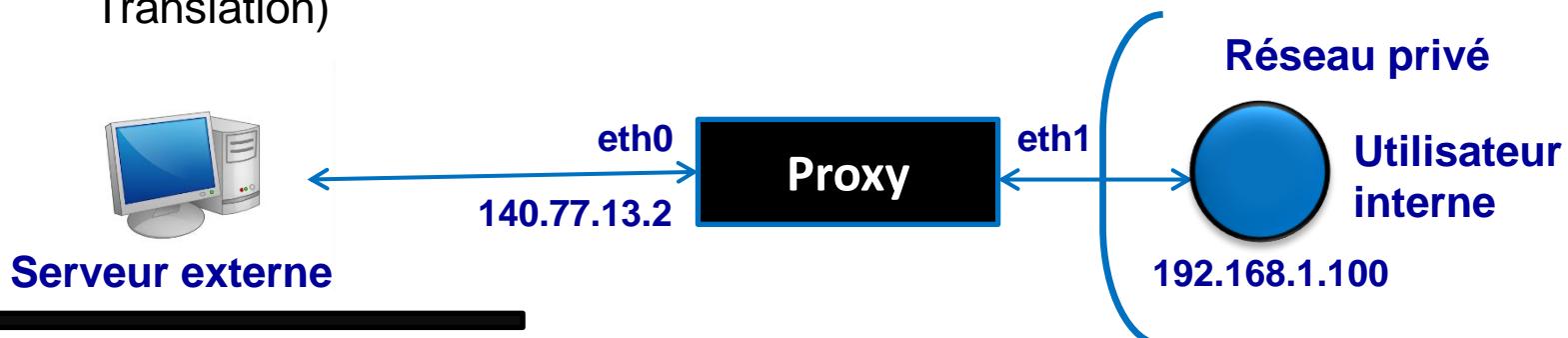


- Sens de l'extérieur vers l'intérieur (reverse proxy)
  - Supposons qu'un utilisateur externe souhaite accéder au serveur web du réseau privé
  - Il envoie sa requête à l'adresse publique 140.77.13.2 (adresse externe du proxy, c'est la seule adresse visible de l'extérieure) sur le port destination 80
  - Le proxy utilise sa table de correspondance pour remplacer l'adresse 140.77.13.2 par l'adresse privée 192.168.0.2 du serveur Web
  - Port 80 = Serveur Web = 192.168.0.2
  - C'est simple tant qu'il n'y a qu'un seul serveur Web dans le réseau privé !
- Lorsque le serveur Web veut répondre à l'utilisateur externe, le proxy :
  - Intercepte la communication
  - Remplace l'adresse privée 192.168.0.2 par l'adresse publique 140.77.13.2
  - Simple !





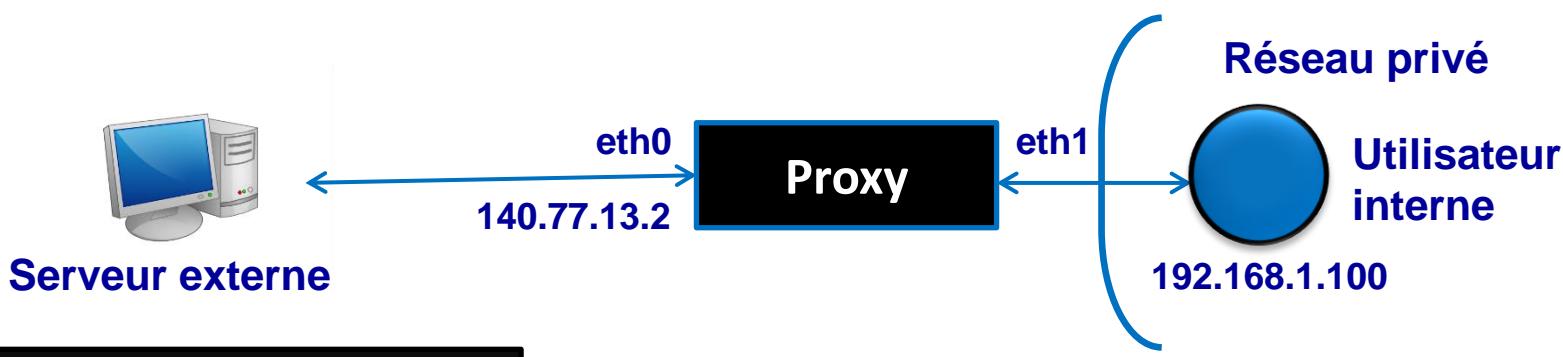
- Sens de l'intérieur vers l'extérieur (proxy)
  - Supposons qu'un utilisateur interne souhaite accéder à un serveur externe
  - Il envoie sa requête à l'adresse publique du serveur externe (en fait, peut-être une adresse de proxy derrière lequel se trouve le serveur externe !)
  - L'utilisateur choisit aléatoirement un numéro de port source > 1024
  - Le proxy remplace l'adresse privée de l'utilisateur interne par son adresse publique 140.77.13.2 du proxy
- Lorsque le serveur externe veut répondre à l'utilisateur interne, le proxy :
  - Intercepte la communication
  - Mais problème : comment savoir qui est le destinataire si, par hasard, deux utilisateurs internes ont utilisé le même numéro de port source
  - Réponse : dans ce sens, il faut aussi faire du PAT (Port Address Translation)





# NAT

- PAT (Port Address Translation)
- Quand le proxy reçoit la demande du l'utilisateur interne (adresse privée  $as1 = 192.168.1.100$ , port source  $ps1$ ) :
  - La proxy choisit un nouveau numéro de port  $pp1$  non utilisé (s'il en existe !)
  - Le proxy enregistre la correspondance  $pp1 \rightarrow (as1, ps1)$
  - Il remplace le couple  $(a1, ps1)$  par  $(140.77.13.2, pp1)$  et envoie la requête au serveur externe
- Quand le serveur externe répond au proxy (adresse destination  $140.77.13.2$  sur le port destination  $pp1$ )
  - La proxy consulte sa table de correspondance
  - Le proxy remplace  $(140.77.13.2, pp1)$  par  $(as1, ps1)$  et envoie le paquet au bon destinataire





# Exemples d'attaque

- Lorsqu'ils ont été conçus, le protocole IP et les protocoles associés (TCP, UDP, ICMP, routage...) n'ont pas pris en compte la sécurité
  - « Concept sécurité » inconnu à l'époque, personne n'imaginait que ces protocoles pourraient être détournés à des fins malveillantes
  - Aucun mécanisme de sécurité n'est donc implémenté au sein de ces protocoles
- Quelques exemples de faiblesses de ces protocoles
  - Absence d'authentification des émetteurs et récepteurs d'un datagramme : usurpation d'adresse IP possible
  - Absence de chiffrement des données, celles-ci sont donc transmises en clair. Un hacker positionné sur un réseau peut donc écouter les connexions et accéder aux données
  - Le routage des datagrammes peut être modifié de façon à rediriger les datagrammes vers un autre destinataire

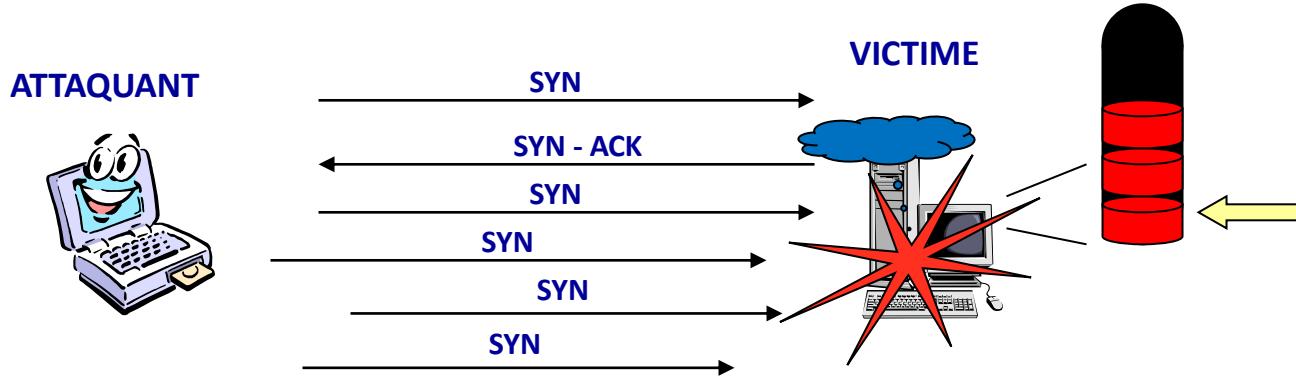


# Exemples d'attaque

- Exemple d'attaque par inondation (Syn flooding)

**Attaque DoS sur les réseaux IP**  
**Connexion à moitié ouverte : le serveur insère les informations d'ouverture dans sa pile**

**Le serveur attend la réponse (ack) du client et conserve dans sa pile des connexions à moitié ouvertes**



**Le client n'envoie pas de ack pour ouvrir la connexion**

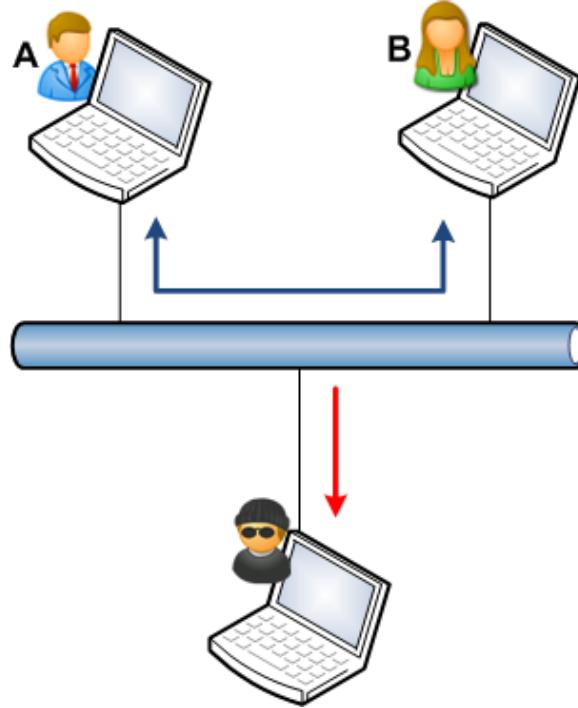
**Trop de connexions à moitié ouverte conduisent à un déni de service**

**NB : L'attaquant forge des paquets SYN avec des adresses IP usurpées (spoofing)**



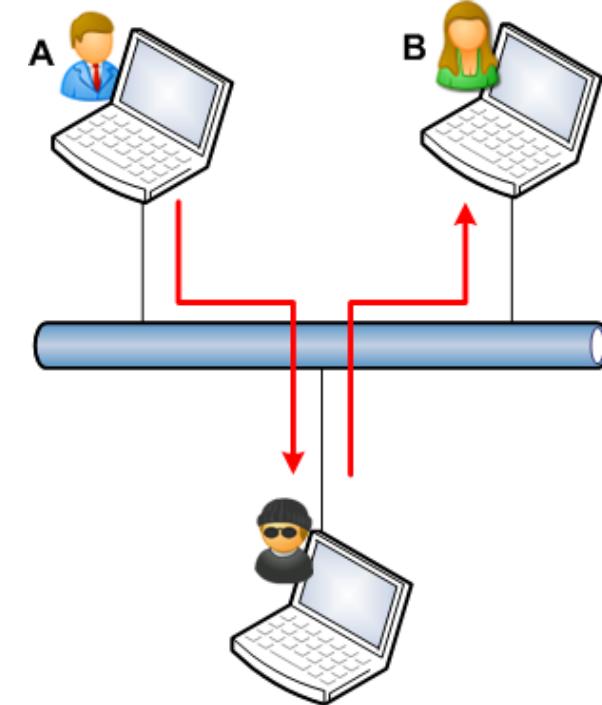
# Exemples d'attaque

- Ecoute de trafic



## Ecoute passive (sniffing)

PC en mode « Promiscuous »  
L'attaquant est en mesure d'écouter les conversations entre A et B (atteinte à la confidentialité des échanges)



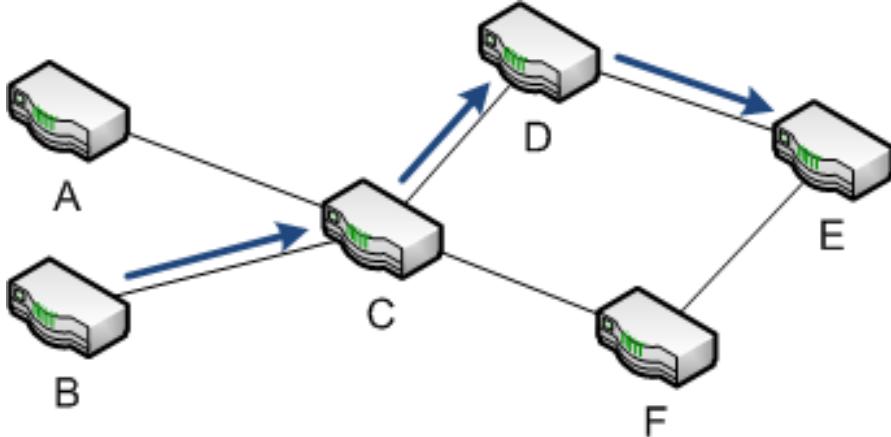
## Ecoute active (Man in the Middle)

L'attaquant est en mesure de s'insérer dans la conversation entre A et B sans que ceux-ci le sachent (atteinte à la confidentialité et à l'intégrité des échanges)



# Exemples d'attaque

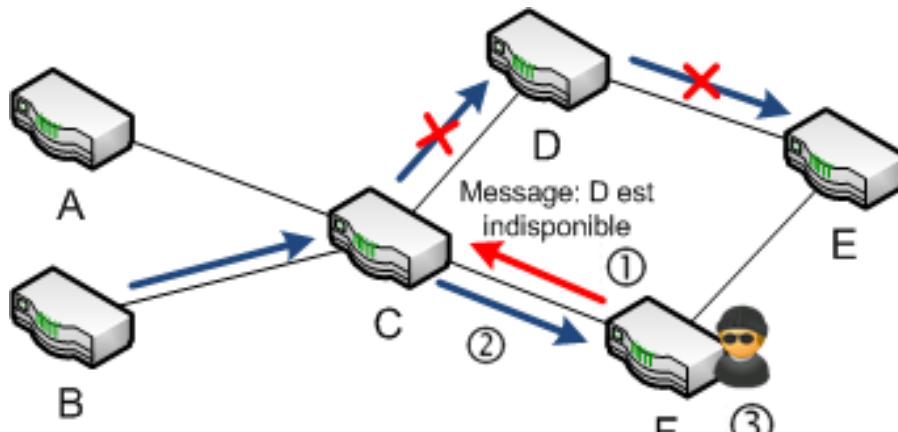
- Modification du routage des datagrammes IP



Chaque routeur possède une table de routage qui indique vers quel routeur voisin transmettre les datagrammes. Cette table peut être mise à jour dynamiquement en fonction des évènements réseaux (protocoles BGP, RIP, OSPF, etc.)

But de l'attaque : dérouter les paquets à destination du réseau E, vers le réseau F maîtrisé par l'attaquant

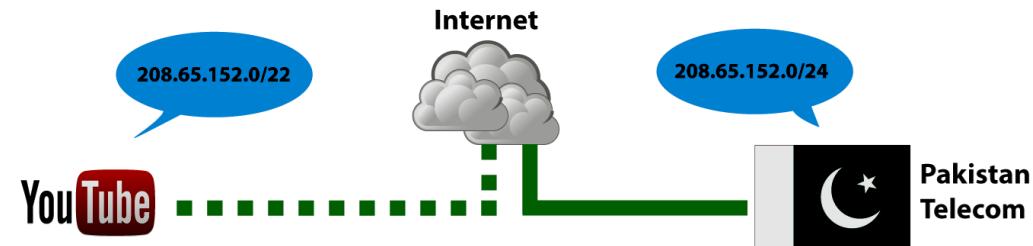
Méthode :



1. L'attaquant utilise le protocole de routage pour indiquer au routeur C que le routeur D est indisponible, et que le routeur F peut router les paquets vers E ;
2. le routeur C transfère donc à F les paquets pour E, afin qu'ils puissent être routés à destination ;
3. Selon le but visé par l'attaquant, celui-ci peut décider de router ou non les paquets vers E.

# Exemples d'attaque

- Modification du routage des datagrammes IP (suite)
- Fonctions de sécurité de BGP (Border Gate Protocol)
  - Pas de mécanisme pour assurer l'intégrité des messages
  - Pas de mécanisme pour assurer l'authenticité des messages
- La sécurité repose essentiellement sur la confiance des opérateurs qui opèrent les routeurs BGP
- Exemple d'incident : Youtube en 2008
  - Le 24 février 2008, le gouvernement pakistanais ordonne le blocage de YouTube
  - Pakistan Telecom exécute l'ordre et annonce à tous les routeurs des fournisseurs d'accès qu'il est la meilleure route à qui envoyer le trafic YouTube
  - Conséquence : création d'un black hole rendant Youtube indisponible pendant 2 heures sur l'ensemble de la planète





# Exemples d'attaque

- Autres exemples d'attaques
- Exploitation de bugs d'implémentation (en général aujourd'hui corrigés)

## Xmass Tree

Envoi de paquets avec tous les flags

TCP à 1

## Land – Blat

Envoi de paquets avec l'adresse IP source égale à l'adresse IP de la cible

## Winnuke

Envoi de packet TCP sur la port 139 (Netbios) avec le pointeur Urgent positionné

## Ping of death

Envoi de paquets ICMP request (ping) dont la taille dépasse la taille maximale autorisée ( $2^{16} = 65535$  octets)

## Tear-Drop

Envoi de paquets mal fragmenté  
Déni de service lorsque le serveur essaye de défragmenter les paquets



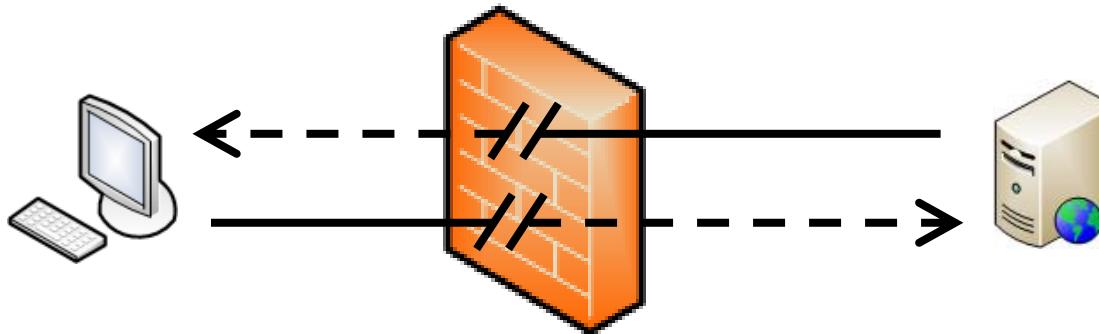
# Pare-feu

- Le pare-feu est un équipement ou logiciel qui agit en tant que filtre réseau
  - Firewall en Anglais
  - Ou coupe-feu (aussi occasionnellement garde-barrière)
- Un pare-feu doit être configuré conformément à une politique de sécurité qui définit les paquets autorisés à traverser le pare-feu (et aussi les interdictions)
- Politique de sécurité = Ensemble de règles
  - Règle = ACL (Access Control List)



# Pare-feu

- Équipement en coupure entre 2 ou plusieurs réseaux
  - Inspecte les paquets réseaux traversant le pare-feu
  - Le pare-feu ne transmet que les paquets qui respectent les règles de filtrage implémentées dans la configuration du pare-feu

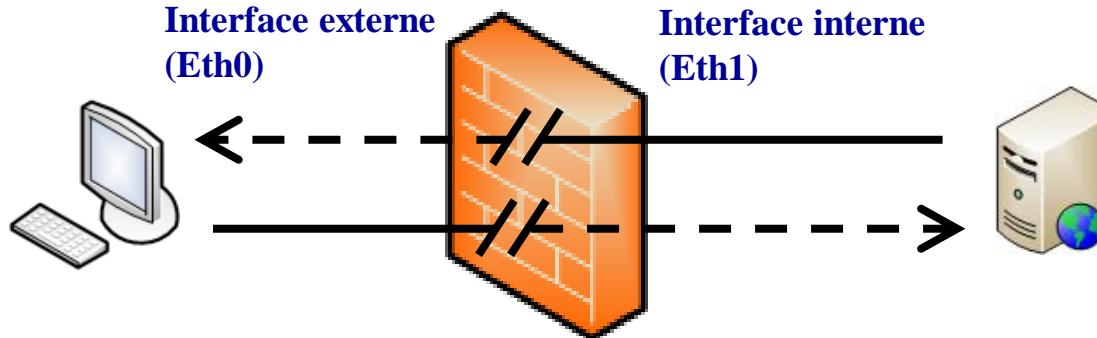


Pour chaque flux entrant ou sortant, le pare-feu interroge ses règles de filtrage pour déterminer s'il doit laisser passer le paquet réseau ou non.



# Pare-feu

- Équipement en coupure entre 2 ou plusieurs réseaux
  - Inspecte les paquets réseaux traversant le pare-feu
  - Le pare-feu ne transmet que les paquets qui respectent les règles de filtrage implémentées dans la configuration du pare-feu



**Exemple de pare-feu avec deux cartes réseau :**

- Interface externe
- Interface interne

**Un pare-feu peut avoir plus de deux cartes réseau**



# Différents types de pare-feu

- Selon l'implémentation
  - Matériel
    - Pare-feu filtrant
    - Parfois intégré dans le routeur
    - « Network appliances »
      - Combine d'autres fonctionnalités
  - Logiciel
    - Serveur pare-feu dédié
    - Pare-feu client ou personnel
- Selon les fonctionnalités
  - Filtrage statique
  - Filtrage dynamique
  - Pare-feu applicatif
    - Pare-feu filtrant
  - Serveur mandataire (proxy)



# Pare-feu statique

- Principes de fonctionnement
  - Examine paquet par paquet
- On parle aussi de pare-feu sans état (stateless)
  - Le pare-feu statique ne conserve pas d'information sur l'état
- Le pare-feu inspecte les paquets réseau en se basant sur les informations de l'en-tête du paquet
- La pare-feu prend un décision (pass, block, log) en fonction des règles de la politique de sécurité



# Pare-feu statique

- La décision de filtrage dépend uniquement des données des couches 2, 3 et 4
  - Adresse IP source/destination
  - Port source/destination
  - Type de protocole utilisé (TCP/UDP/ICMP)
  - Signalisation du paquet (SYN, ACK)
  - Adresses MAC
  - Etc.
- Peut être implémenté en logiciel (ex : IPtables) ou matériel (ex : routeur Cisco)



# Pare-feu dynamique

- Principes de bases
  - Examine paquet par paquet, mais essaie d'établir des relations entre paquets
    - UDP
      - Associe le paquet avec d'autre paquets sur mêmes ports et adresses
      - En général, permet seulement des réponses si requêtes originales venant d'adresses internes
    - TCP
      - Garde l'information sur état et direction de la session TCP
      - Regarde en plus les flags TCP pour déterminer si hors-protocole
    - Applications qui changent de port (e.g. FTP)
      - Suit et autorise les ports éphémères utilisés par les applications

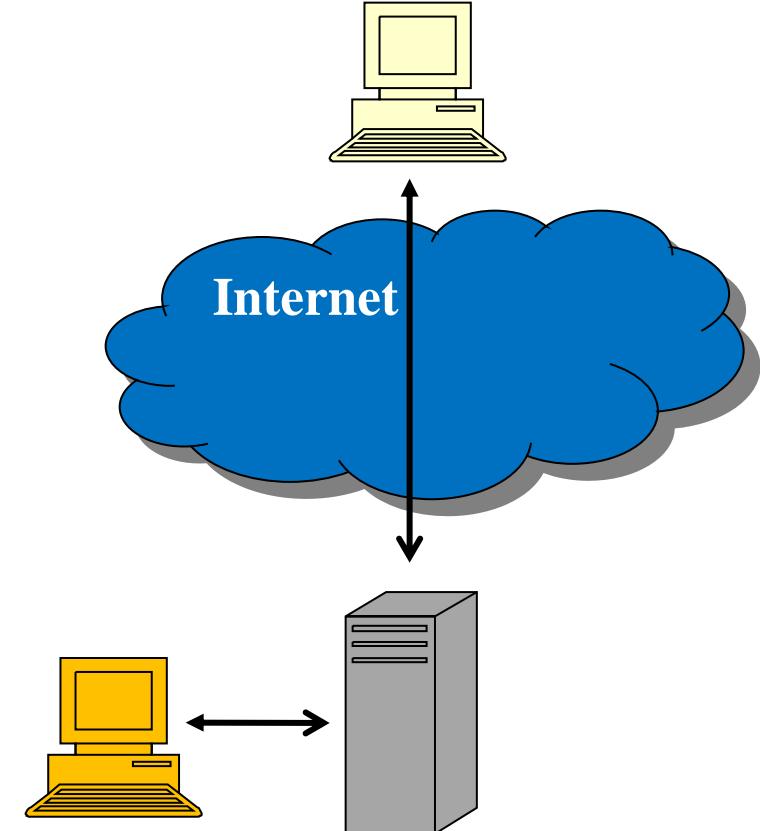
# Limites des pare-feu statiques et dynamiques

- Ces pare-feu sont limités aux couches 2 à 4 du modèle OSI
  - Pas de connaissance de l'état de la connexion au niveau applicatif
  - Usurpation de port
- Plusieurs attaques passent par les ports ouverts
  - Injection SQL
  - Attaques de force brute
- En augmentant l'intelligence d'un pare-feu pour interpréter les protocoles applicatifs on peut obtenir une meilleure analyse du trafic
  - Pare-feu applicatif
  - Serveur mandataire ou « proxy »



# Proxy

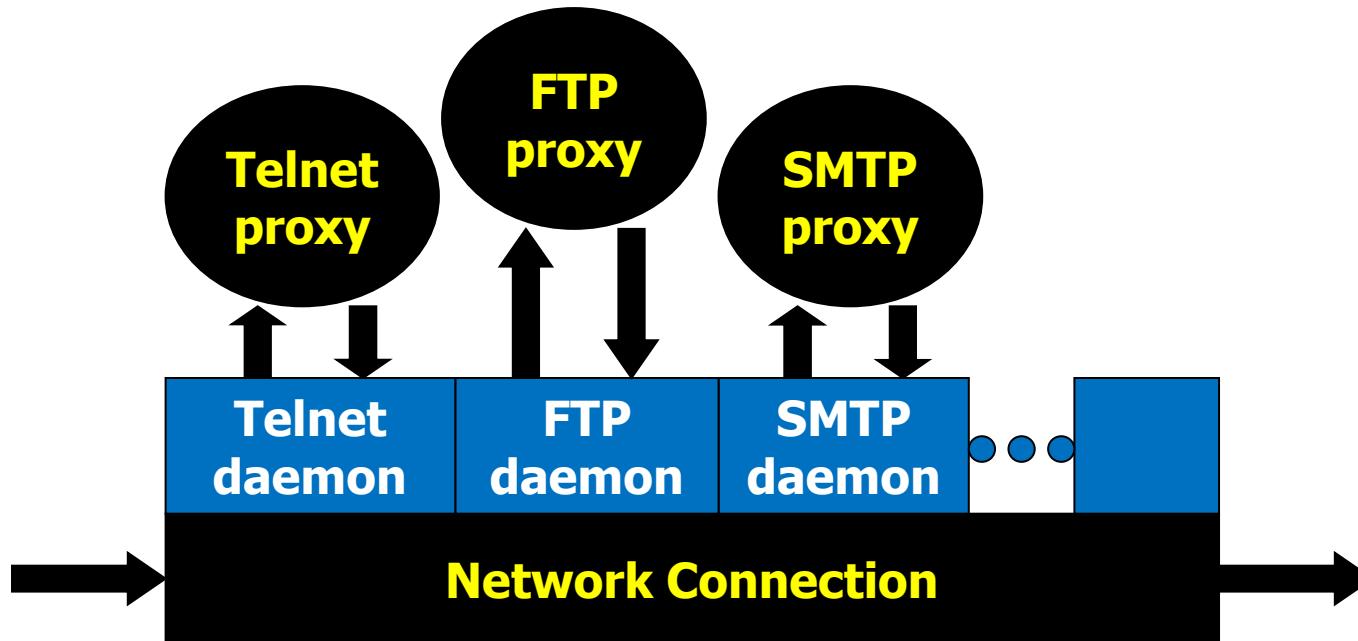
- Le proxy est une implémentation particulière de pare-feu couche 7
- Le proxy peut s'interposer pour faire de l'inspection et du blocage
- Le proxy camoufle l'adressage interne
  - L'extérieur voit uniquement le proxy





# Proxy

- Besoin de définir un proxy par protocole analysé
  - Le démon associé au proxy s'active lorsque la communication est détectée
  - Le proxy peut faire du DPI (Deep Packet Inspection)
    - Intéressant pour la sécurité mais coûteux en performance





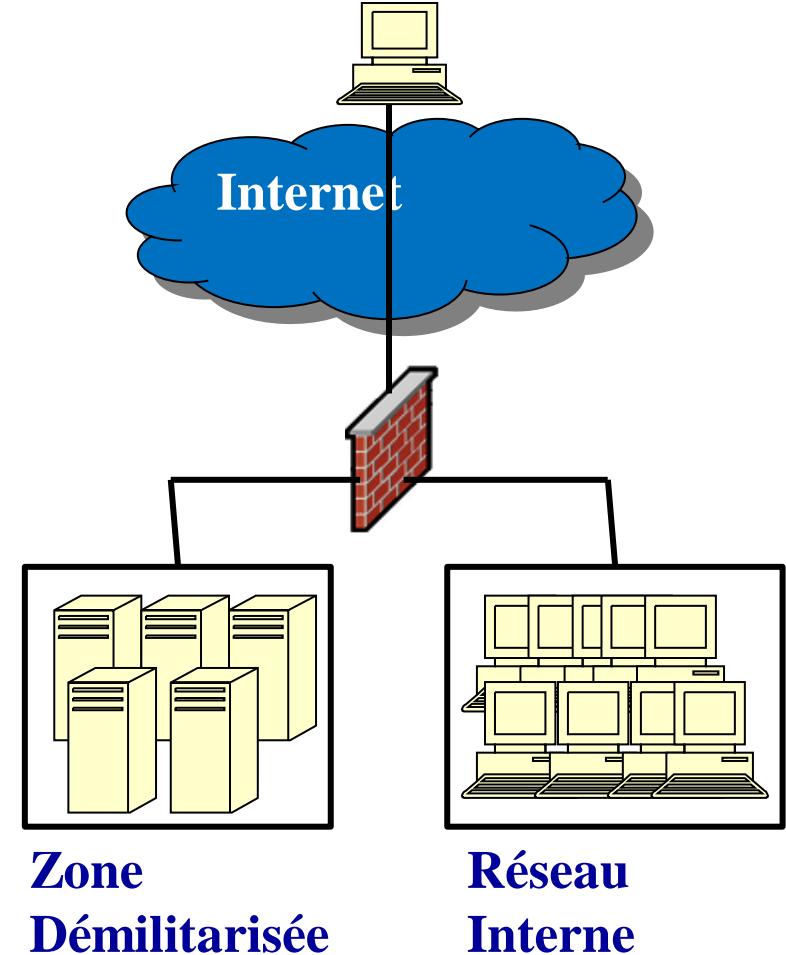
# Configuration sécuritaire du proxy

- Principes de bastionnage
  1. Exécuter une version sécurisée du système d'exploitation
  2. Installer uniquement les services nécessaires pour l'administration réseau
  3. Configurer chaque proxy pour assurer un sous-ensemble nécessaire des commandes du standard de l'application
  4. Concevoir chaque module de proxy de façon minimale et sécurisée
  5. Chaque proxy doit journaliser le trafic, chaque connexion et la durée de chaque connexion
  6. Chaque proxy est indépendant des autres proxies
  7. Chaque proxy s'exécute comme un usager non privilégié dans un répertoire privé et sécurisé



# DMZ

- Pour augmenter la flexibilité d'un pare-feu, on segmente le réseau
- Un attaquant qui compromet une machine dans un segment doit travailler aussi fort pour compromettre une machine dans un autre segment
- Une zone spécialisée bâtie pour exposer des services sur Internet est appelée zone démilitarisée (DMZ)





# Zone démilitarisée (DMZ)

- Objectif
  - Permettre de fournir des services de ou vers l'extérieur du réseau interne, tout en protégeant celui-ci
- Principe de base
  - Créer une zone intermédiaire où se trouve les services strictement nécessaires
  - Protégé par un pare-feu/passerelle
  - Isolé du réseau interne par un pare-feu/passerelle



# Zone démilitarisée (DMZ)

- Avec une DMZ, on peut raffiner les règles de filtrage
  - Accepter les connexions entrantes uniquement vers la DMZ
  - Refuser les connexions (indésirables) en provenance de l'intérieur
  - Refuser les connexions à partir des serveurs vers l'intérieur
- Permet de faire la séparation des zones en fonction des risques
- Le concept de segmentation peut être utilisé pour isoler d'autres types de zones à risques
  - Zone sans-fil
  - Zone partenaires
  - Serveurs de test



# Services typiques dans une DMZ

- Services fournis à l'extérieur
  - Serveur DNS (pour adresses DMZ seulement)
  - Serveur SMTP
  - Serveur Web
  - Passerelle VPN
- Services fournis à l'intérieur
  - Serveur mandataire Web
  - M-à-j des fichiers du serveur Web
  - Connexion avec serveur SMTP interne
  - Connexion entre serveur Web et serveur de BD



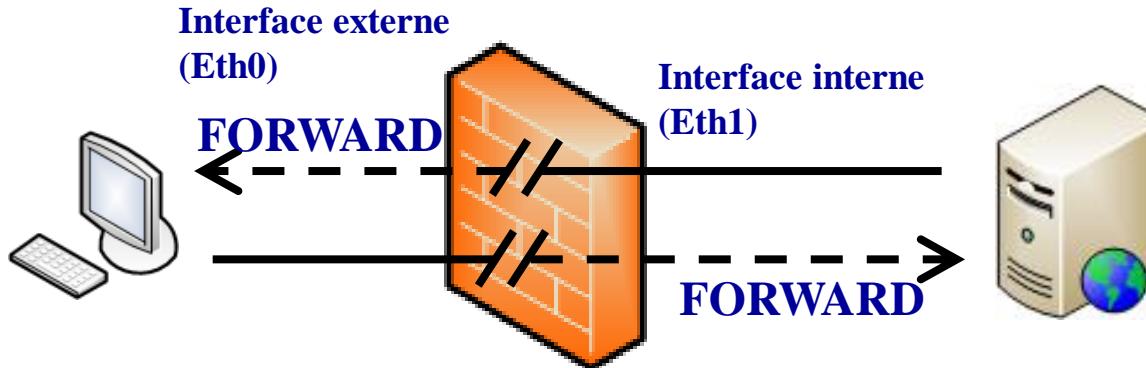
# NetFilter / IpTables

- NetFilter
  - Firewall stateful sous LINUX
  - Logiciel Open Source
  - Première version en 1998
- IpTables
  - Module de NetFilter qui permet d'écrire et de gérer les ACLs
  - Module réalisant le filtrage de paquets (noyaux LINUX  $\geq 2.4$ )



# NetFilter / IpTables

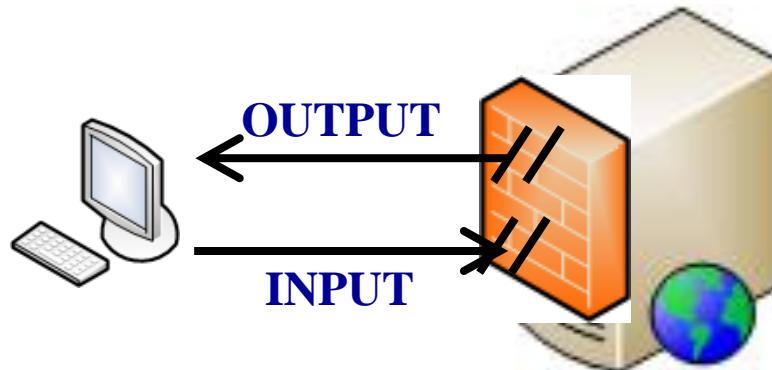
- Deux modes de fonctionnement (1/2)
- Pare-feu réseau : Les paquets qui arrivent sur le pare-feu sont filtrés et transmis à la destination s'il sont acceptés par le pare-feu





# NetFilter / IpTables

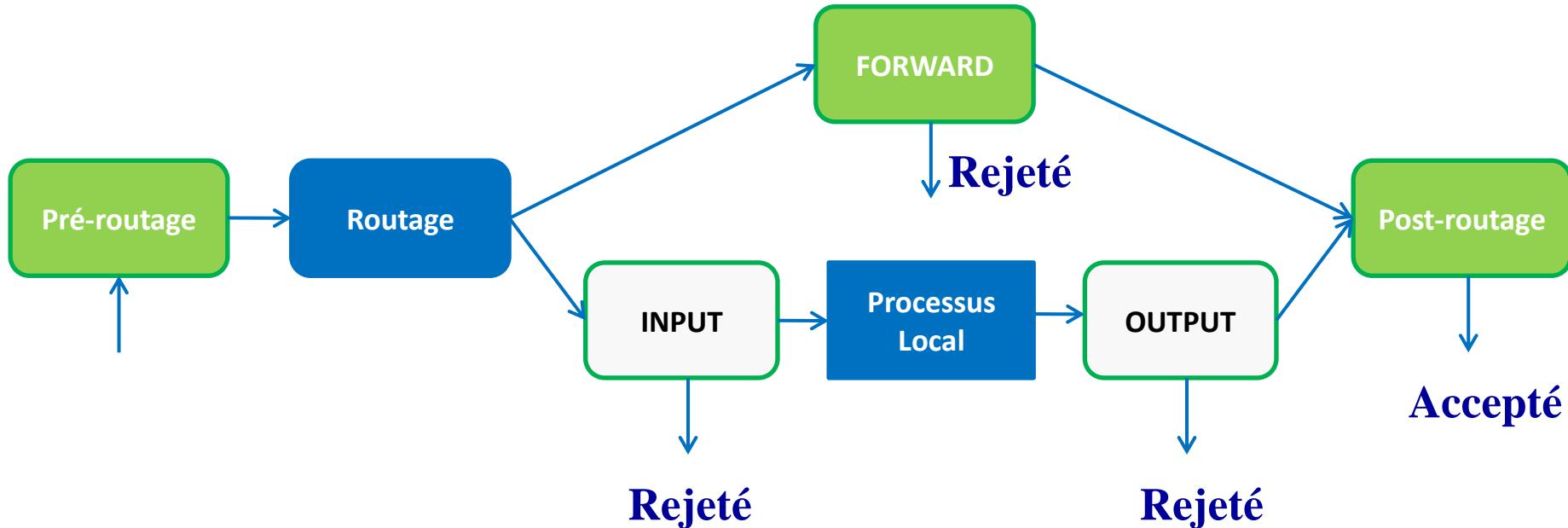
- Deux modes de fonctionnement (2/2)
- Pare-feu personnel : Le pare-feu est associé à un ordinateur hôte et filtre le trafic qui arrive et qui sort de cet ordinateur





# NetFilter / IpTables

- Les ACLs sont associées à des chaines
  - 5 chaines prédéfinies
  - Possibilité de définir de nouvelles chaines



# NetFilter / IpTables

- Filtrage des paquets IP, TCP, UDP ou ICMP
- Spécification de règles pour le rejet ou l'acceptation de paquets
- Règles traitées de manière séquentielle : Le paquet sort dès qu'il rencontre une règle qui peut lui être appliquée

Exemple :

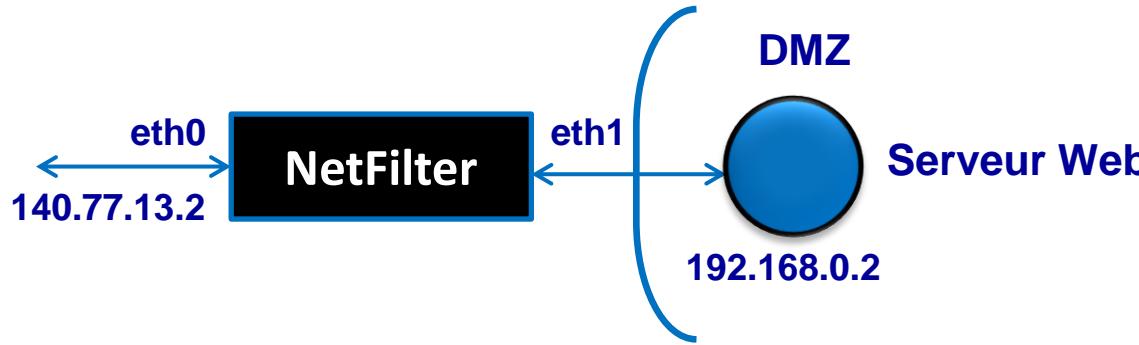
Accepter de router les paquets entrant sur eth0 tels que :

@source	@dest	P-source	P-dest
0/0	140.77.13.2	1024-65535	80

```
iptables -A FORWARD -s 0/0 -i eth0 --d 140.77.13.2 -o eth1 -p TCP  
--sport 1024:65535 -dport 80 -j ACCEPT
```

# NetFilter / IpTables

- Fonctionnalités NAT de NetFilter



- Modification de la destination du paquet avant le routage  
(paquet reçu de l'extérieur)

```
iptables -t nat -A PREROUTING -d 140.77.13.2 --dport 80 -i eth0 -j DNAT -to-destination 192.168.0.2:80
```

- 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 -i eth1 -j SNAT 140.77.13.2
```



# NetFilter / IpTables

- Suivi des connexions
- 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 le routeur à relayer tous les paquets reçus concernant de nouvelles connexions sur le port 22

```
iptables -A FORWARD -p tcp -i eth0 -dport 22 –sport 1024:65535 -m state -state NEW -j ACCEPT
```



# Focus sur le filtrage à états

- Tous les pare-feu à états doivent utiliser une table interne de sessions pour suivre l'état des paquets traversant le pare-feu
- Exemple pour une connexion TCP

TCP SYN Packet

Packet	Prot	Src-IP	Dst-IP	SP	DP	SYN	ACK
Packet#1	TCP	192.168.1.3	192.168.2.2	2235	80	1	0

Session Table entry after receiving the SYN packet

TCP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	TCP	192.168.1.3	192.168.2.2	2235	80	SYN_RCVD	Half Open connection, default 10s

Session Table entry after completing the three-way hand shaking

TCP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	TCP	192.168.1.3	192.168.2.2	2235	80	ESTABLISHED	Full connection, default 3600s

State  
NEW

State  
ESTABLISHED



# Focus sur le filtrage à états

- Avantage de la table de session
  - Lorqu'un pare-feu à état reçoit un paquet, il va seulement regarder s'il y a une session établie correspondant à ce paquet dans la table de session
  - Dans ce cas, le paquet sera accepté sans consulter les ACLs
  - Le filtrage est beaucoup plus rapide dans ce cas



# Focus sur le filtrage à états

- Les pare-feu à états permettent aussi le suivi des connexions UDP
  - Même si UDP est un protocole sans état
  - S'il y a une ACL qui autorise un paquet UDP, la pare-feu insère une nouvelle entrée dans la table de session
  - Tout paquet entre la source et la destination correspondant au ports spécifiés pourra traverser le pare-feu dans les deux sens tant que le timeout n'est pas atteint
  - Le Timeout est une option configurable (la valeur par défaut en général de 2mn)

State  
RELATED

UDP Packet					
Packet	Prot	Src-IP	Dst-IP	SP	DP
Packet#1	UDP	192.168.1.3	192.168.2.3	3454	53

Session Table entry after receiving the UDP request packet

UDP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	UDP	192.168.1.3	192.168.2.3	3454	53	Request RCVD	Default 40s

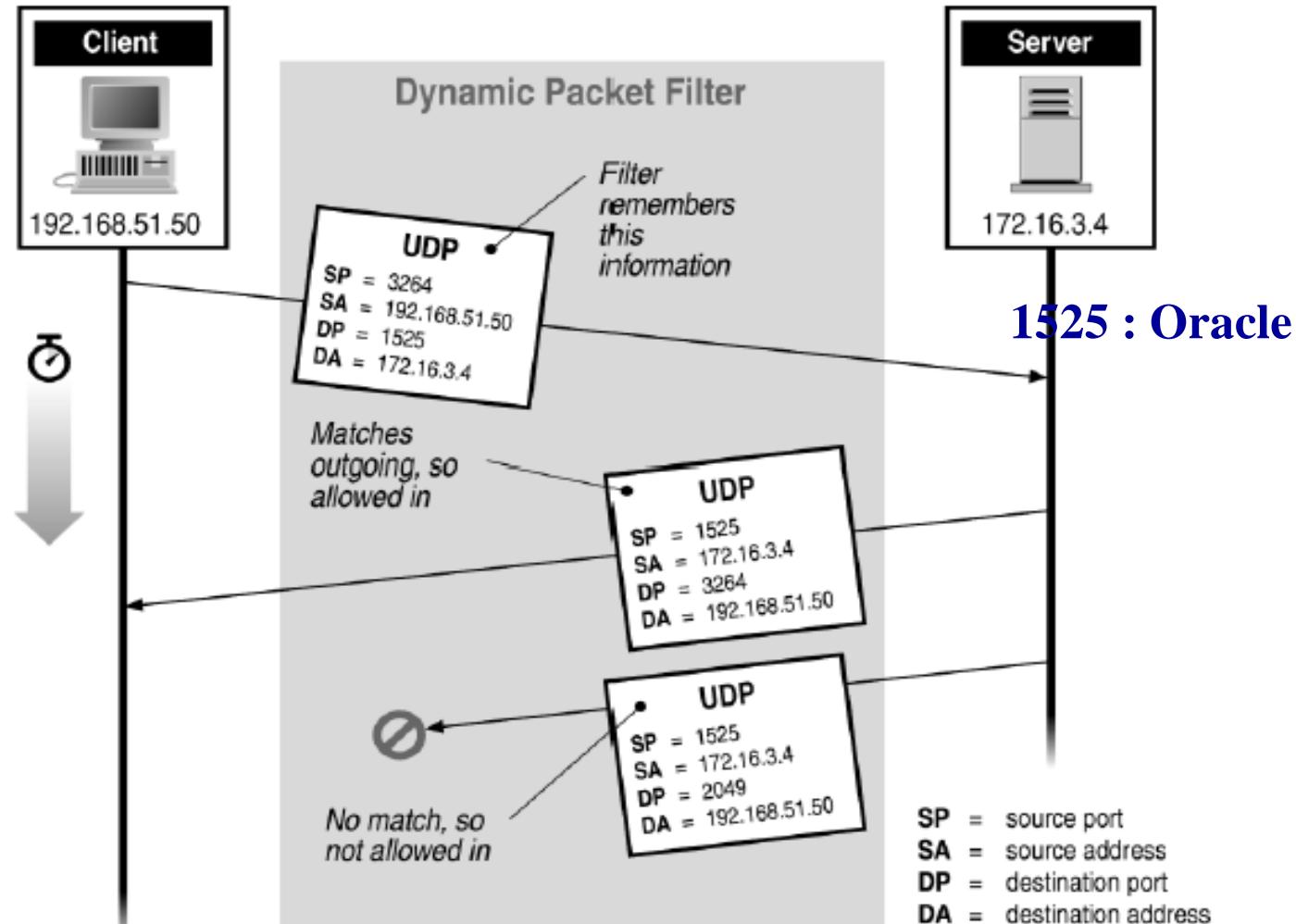
Session Table entry after receiving the UDP reply packet

UDP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	UDP	192.168.1.3	192.168.2.3	3454	53	Response RCVD. Connection is considered as ESTABLISHED	Default 2 mins



# Focus sur le filtrage à états

- Exemple : Inspection UDP à état





# Focus sur le filtrage à états

- Inspection ICMP à état
  - Exemple : ICMP Echo Request (ping)
  - Si ce ping est accepté par le pare-feu, le paquet ICMP Echo Reply sera géré comme du traffic “related”

ICMP echo request packet

Packet	Prot	Src-IP	Dst-IP	TYPE	CODE	Identifier	Sequence number
Packet#1	ICMP	192.168.1.4	192.168.2.6	8	0	100	1

Session Table entry after receiving the ICMP echo request packet

State	ICMP Connection	Prot	Src-IP	Dst-IP	TYPE	CODE	Identifier	Sequence number	Connection State	Timeout
RELATED	Connection#1	ICMP	192.168.1.4	192.168.2.6	8	0	100	1	Request RCVD	Default 2s



# Focus sur le filtrage à états

- Inspection ICMP à état
  - Suivi des messages d'erreur
  - Exemple : une paquet UDP est envoyé sur le port 53
  - Supposons que ce paquet est accepté par le pare-feu mais la destination n'est pas un serveur DNS
  - La destination va renvoyer un message ICMP port unreachable [Type: 3, Code: 3]

UDP Packet

Packet	Prot	Src-IP	Dst-IP	SP	DP
Packet#2	UDP	192.168.1.3	192.168.2.4	2200	53

Session Table entry after receiving the UDP request packet

UDP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#2	UDP	192.168.1.3	192.168.2.4	2200	53	Request RCVD	Default 40s

ICMP error message [TYPE:3, CODE:3] embedded to UDP connection #2

Packet	Prot	Src-IP	Dst-IP	TYPE	CODE	Payload Attributes			
						Src-IP	Dst-IP	SP	DP
Packet#2	ICMP	192.168.2.4	192.168.1.3	3	3	192.168.1.3	192.168.2.4	2200	53

Le pare-feu à état va accepter ce message d'erreur ICMP comme un trafic « related » à la connexion UDP

State  
RELATED



# Attaque DOF contre un pare-feu à état

- DOF = Denial of Firewall
- Premier objectif de l'attaque
  - Saturer la table de session
  - Lorsque la table de session est pleine, la pare-feu ne peut plus créer de nouvelles sessions et va “dropper les nouvelles demandes
  - Exemple : Innondation TCP, UDP ou ICMP contre les pare-feu à état
- Second objectif de l'attaque
  - Saturer les ressources de calcul du pare-feu
  - Exemple : attaque black-nurse
  - L'attaquant envoie une requête DNS sur le port 53 à un serveur qui n'est pas un serveur DNS
  - L'attaquant va ensuite envoyer au pare-feu des messages ICMP port unreachable spoofé avec l'adresse destination du serveur
  - Les expérimentations montrent que 7000 paquets par seconde suffisent à saturer les ressources du pare-feu