acceis

Gwenn Feunteun

gwenn@acceis.fr

Architecture & sécurité réseau

Les composants d’un réseau sécurisé

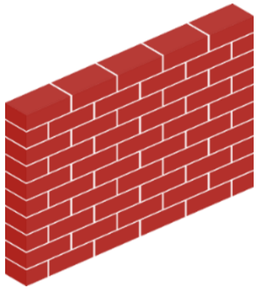
acceis

Les pare-feux

Se protéger par l'architecture

Pare-feu

Le pare-feu est une solution logicielle ou matérielle, placé en rupture des flux de communication, dont l'objectif et de contrôler le trafic entre différentes zones de confiance du système d'information (incluant éventuellement Internet). Historiquement, il effectue un filtrage en fonctions de critères relevant des **niveaux 3 et 4** du modèle OSI (adresses IP, nature des protocoles, ports sources et de destination, etc.).



Il est considéré comme la pierre angulaire de la sécurité d'une infrastructure réseau.

Pare-feu

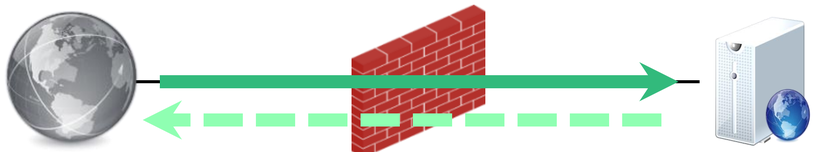
Il existe deux grande catégorie de pare-feu, liés à leur capacité de filtrage en fonction du contexte des communications :

- **Le pare-feu sans état** (*stateless*), le plus ancien, qui traite les datagrammes IP indépendamment les uns des autres en les comparant à une liste de règles préconfigurées.

- **Le pare-feu avec état** (*statefull*), qui tient compte du contexte des échanges et en particulier du sens des communications (*Stateful Packet Inspection* ou SPI). Il peut également inspecter certains échanges au niveau applicatif pour autoriser certains flux à la volée (pour le transfert de fichiers via FTP par exemple).

Pare-feu

Cas d'usage : exposition d'un service HTTP sur Internet, pare-feu **sans** état.



Comme le pare-feu sans état ne tient pas compte du contexte de la communication, il est nécessaire de définir deux règles de filtrage (une pour les datagrammes IP à destination du serveur et une autre pour les réponses du serveur).

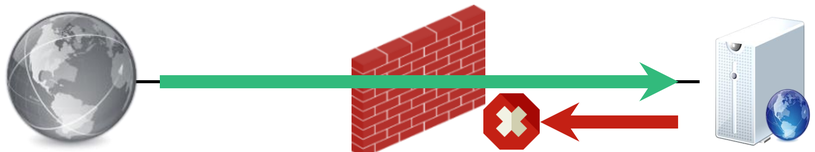
IP source	Port TCP source	IP destination	Port TCP destination	Statut
*	*	192.168.0.1	80	✓
192.168.0.1	80	*	*	✓

Risque

Cela implique que le serveur est autorisé à accéder à n'importe quel service sur Internet, à la seule condition que le port TCP source soit le 80. En cas de compromission, un attaquant peut très facilement exfiltrer des données ou rebondir.

Pare-feu

Cas d'usage : exposition d'un service HTTP sur Internet, pare-feu **avec** état.



Le pare-feu avec état maintient en mémoire une table des connexions. En plus des règles définies, il dispose d'une règle implicite autorisant les datagrammes appartenant à un connexion en cours (dont l'établissement a fait l'objet d'un contrôle vis-à-vis des règles en vigueur). Il n'est plus nécessaire d'autoriser les flux sortants.

IP source	Port TCP source	IP destination	Port TCP destination	Statut
*	*	192.168.0.1	80	✓

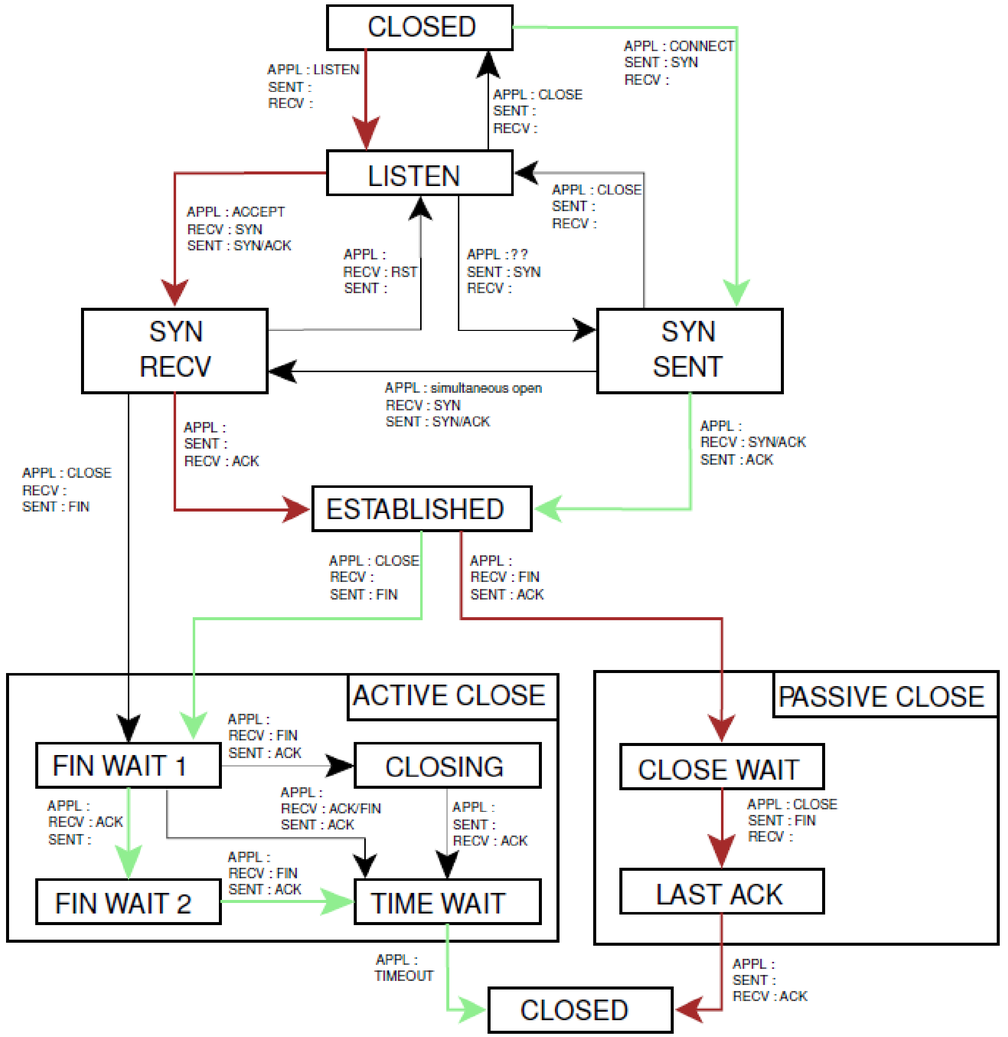
Risque

Cela implique que le pare-feu doit allouer des ressources internes (mémoire) pour assurer le suivi des communications dans le temps. En cas de (très) nombreuses communications simultanées, le pare-feu pourrait être saturé et être à l'origine d'un déni de service.

Filtrage TCP

Une connexion TCP est complétement caractérisée par :

- L’adresse IP source.
- L’adresse IP destination.
- Le port source.
- Le port destination.
- L’état de la connexion dans l’automate TCP.



Filtrage UDP

UDP n'est pas un protocole orienté connexion. Cependant il est souvent utilisé dans un mode client-serveur. Dans ce cas un client initie une communication avec un serveur par un premier paquet UDP, et le dialogue se poursuit dans les deux sens.

- Il est possible d'étendre la notion de connexion au cas du protocole UDP ;
- Le début d'une connexion peut être détectée par un paquet UDP circulant dans un sens défini avec des ports sources et/ou destination particulier ;
- La fin d'une connexion est plus difficile à définir. En général, on utilise un délai de garde : une connexion est considérée comme terminée quand plus aucun paquet n'a circulé (dans chacun des deux sens) depuis un délai défini arbitrairement (par exemple une minute).

Différents types de pare-feu

Matériels

...

Différents types de pare-feu

Matériels

- Performants (près de 100 Gbps), grâce notamment à l'utilisation de FPGA ou d'ASIC pour le décodage protocolaire ;
- Très complets avec des fonctionnalités additionnelles.

- Peut être cher, voire très chers (plus de 100 000 € pour un pare-feu de cœur de réseau).

Différents types de pare-feu

Logiciels

pfSense

netfilter
firewalling, NAT, and packet mangling for linux

+ pare-feux systèmes …

Différents types de pare-feu

Logiciels

- Gratuits pour certains ;

- Peu, voire pas de support en dehors de la communauté pour les gratuits ;
- Performance moins bonnes.

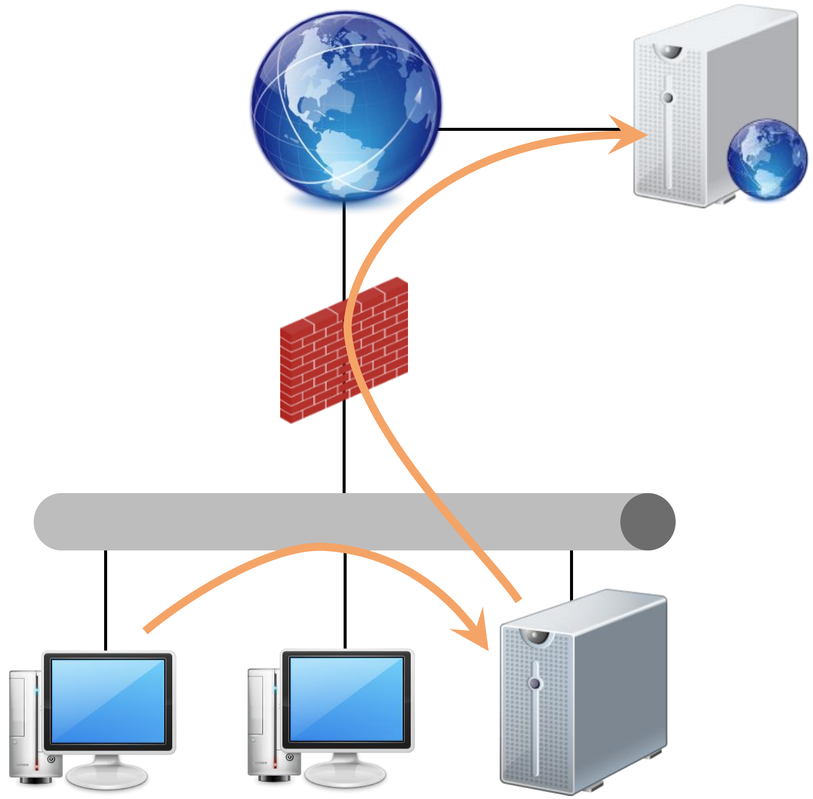
UTM / NGFW

Les pare-feu moderne ne se contentent plus d'assurer « simplement » le filtrage des communication au niveaux 3 et 4. Avec le temps, ils se sont vu adjoindre de nouveaux modules pour prendre en charge de nouvelles fonctions : concentrateur VPN, inspection du contenu applicatif, détection/prévention des intrusions (NIDS / NIPS), « antivirus de flux », proxy, etc.

Ces équipements proposant un approche globale du traitement de la sécurité des communication sont généralement regroupé sous le terme d'*Unified threat management* (UTM) ou de *Next Generation Firewall* (NGFW).

Le cas du proxy

Le proxy est un serveur mandataire destiné à effectuer des requêtes sur Internet pour le compte de tiers.



Le cas du proxy

- Il assure une rupture des flux ;
- Il est capable de limiter les protocoles utilisés ;
- Il peut filtrer les requêtes en fonction d'une politique définie (notion de listes blanches / noires).

➔ **C'est une forme de pare-feu.**

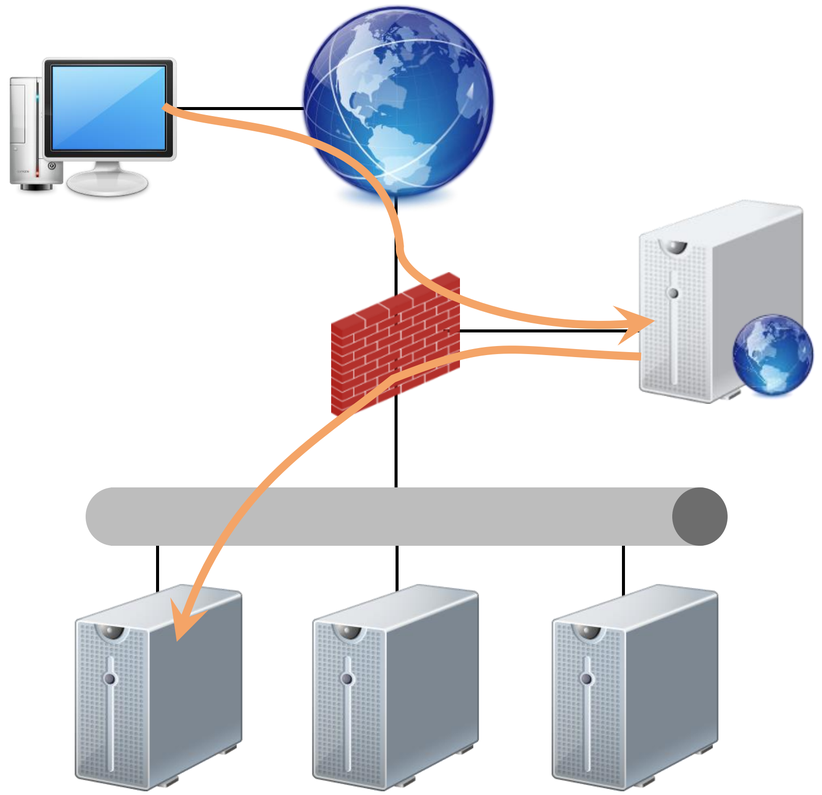
Le cas du proxy

- Initialement utilisé pour des questions de performances (fonctions de cache) ;
- Agit sur les couches au-delà du niveau 4 ;
- Supporte généralement un ensemble de protocoles restreints (HTTP/S, FTP/S…) ;
- Permet de mettre en place des mécanismes d'authentification des utilisateurs ;
- Assure une journalisation « fine » des transactions ;
- Masque « l'identité » des utilisateurs ;
- Peut mettre en place une rupture des flux HTTPS.

➔ **Ce n'est pas vraiment un pare-feu.**

Le cas du reverse-proxy

Le reverse-proxy est un serveur mandataire *inverse* destiné à recevoir des requêtes depuis Internet, à destination des serveurs du réseau interne.



Le cas du reverse-proxy

- Il « masque » les ressources accédées depuis l'extérieur ;
- Il permet de réaliser de la répartition de charge ;
- Il assure généralement des fonctions de cache ;
- Il réalise un filtrage des requêtes (détection et prévention d'intrusion)
  ➔ il est point de terminaison TLS ;

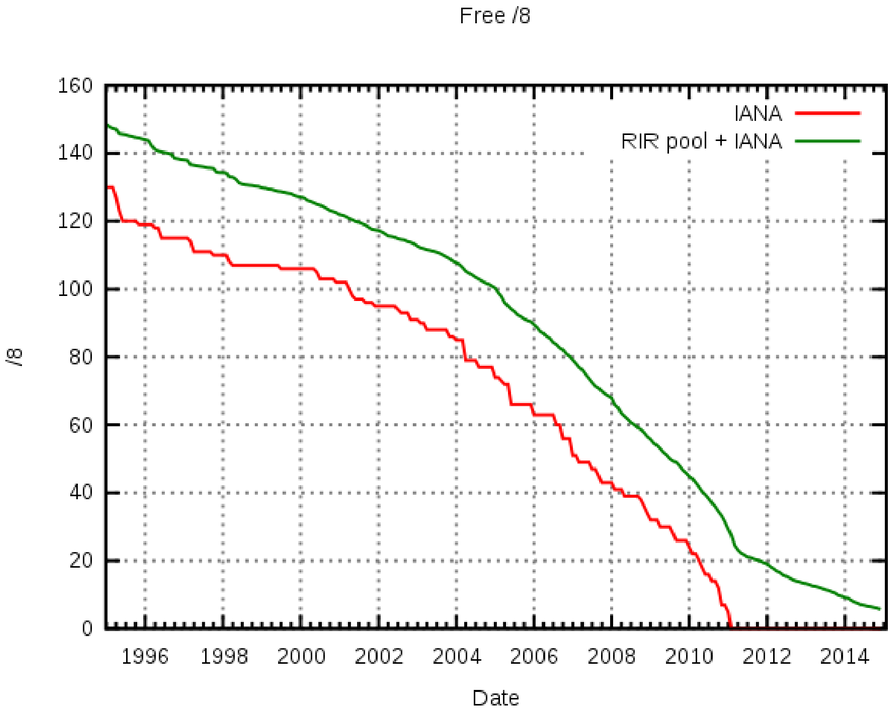
WAF

Le *web application firewall* (WAF) est un type particulier de pare-feu dédié à la sécurité de la couche applicative. Il sert à protéger les services web des attaques auxquelles ils peuvent être exposés (*Cross Site Scripting*, Injection SQL, exécution de commande, *directory traversal*…). Pour cela, il dispose de capacités de décodage, d'analyse et de filtrage du contenu HTTP, voire HTTPS (dans ce cas, le pare-feu est souvent le point de terminaison TLS).

Il est souvent assimilé à un reverse-proxy qui, du point de vue sécurité, fournit sensiblement les mêmes fonctions.

Pénurie d'IPv4

La croissance du nombre d'utilisateurs et de serveurs d'Internet s'accompagne d'un épuisement des adresses IPv4, c'est-à-dire de la diminution progressive de la quantité d'adresses IPv4 publiques disponibles. Cet épuisement menace la croissance du réseau internet. En février 2011, la réserve de blocs libres d'adresses publiques IPv4 de l'*Internet Assigned Numbers Authority* (IANA) est arrivée à épuisement.



Pénurie d'IPv4

Plusieurs techniques ont été proposées pour résoudre le problème ou repousser l'échéance de l'épuisement complet des adresses IP :

- les registres Internet régionaux (RIR) ont développé des politiques d'assignation d'adresses plus contraignantes, qui tiennent compte des besoins réels à court terme.
- Utilisation de système tel que SNI pour installer plusieurs certificats sur la même IP ;
- IPv6, la nouvelle version du Protocole Internet développée dans ce but pendant les années 1990, et dont la capacité d'adressage est considérable ;
- NAT, qui permet à de nombreux ordinateurs d'un réseau privé de partager une adresse publique, mais qui nuit au bon fonctionnement de certains protocoles ;
- la récupération des blocs assignés autrefois…

Traduction d’adresses (NAT)

Afin de faire face à la pénurie d’adresses IPv4, le mécanisme de traduction d’adresse (*Network Address Translation*) a été proposé pour des clients possédant une adresse IPv4 non routable sur Internet (adresses IPv4 privées) situés derrière un point d’accès possédant lui une adresse IPv4 routable, le NAT permet :

- d’accéder à des services situés sur Internet (connexions sortantes) ;
- d’être la destination pour certains services (ports) de connexions entrantes ;

Ceci permet par effet de bord d’accroître la sécurité du réseau interne.Il est en effet masqué du reste d’Internet, sauf pour certains services sélectionnés par la politique de sécurit é du pare-feu.

Traduction d'adresses (NAT)

NAT dynamique, pour les connexions sortant du réseau interne :

- Traduction de l'adresse de source par une adresse fixe (*SNAT*).
- Traduction de l'adresse de source par une adresse dynamique (*Masquerading*).

NAT statique, pour les connexions entrant vers le réseau interne :

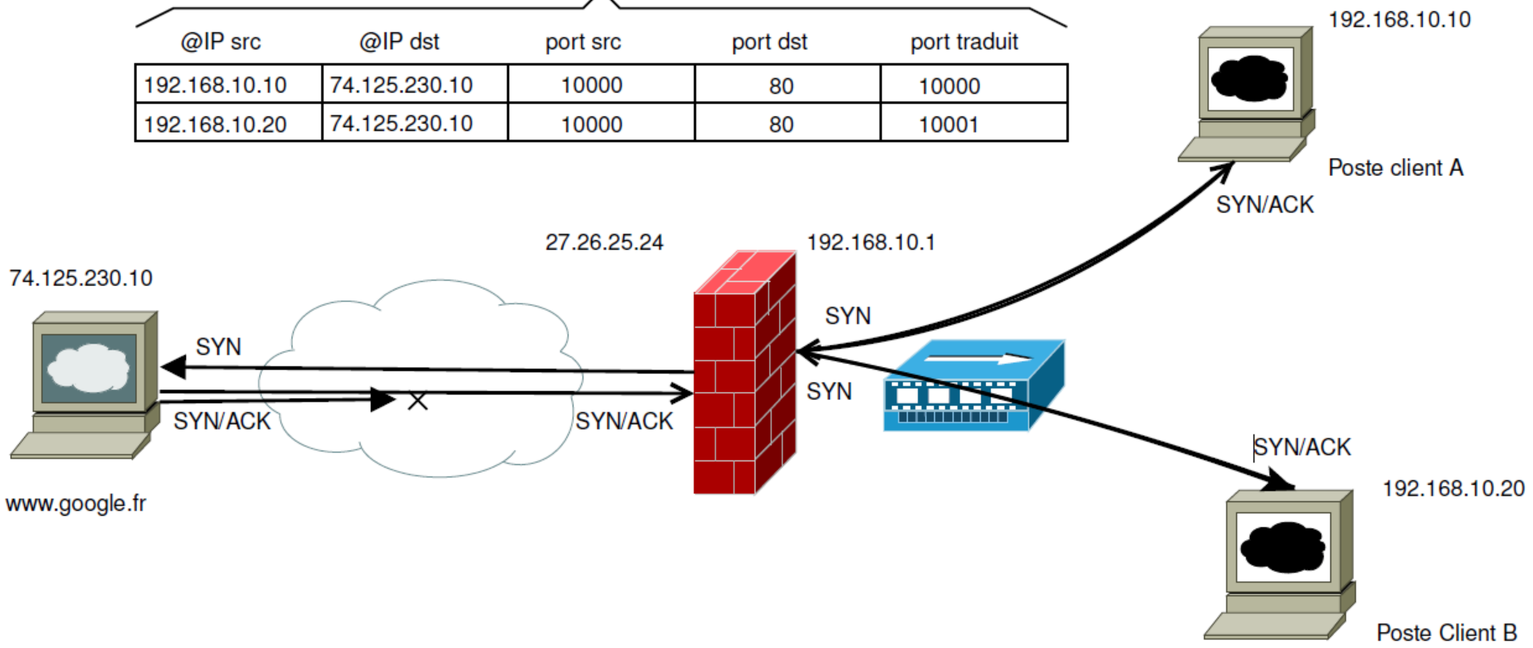
- Traduction de l'adresse de destination par une adresse fixe (*DNAT*).
- Traduction du port de destination par un port différent.

Toutes ces traductions nécessitent de pouvoir suivre les connexions et sont donc étroitement liées au système de suivi de connexion.

Traduction d’adresses (SNAT)

Table de suivi des connexions

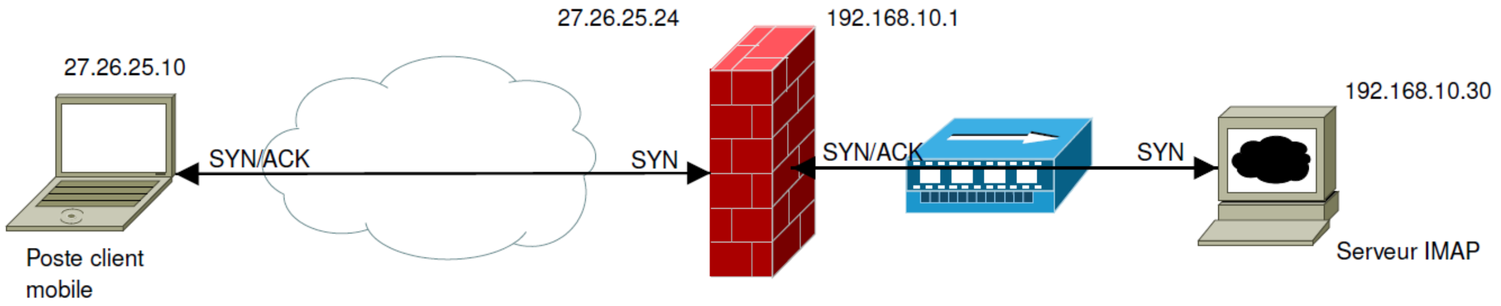
@IP src	@IP dst	port src	port dst	port traduit
192.168.10.10	74.125.230.10	10000	80	10000
192.168.10.20	74.125.230.10	10000	80	10001



Traduction d’adresses (DNAT)

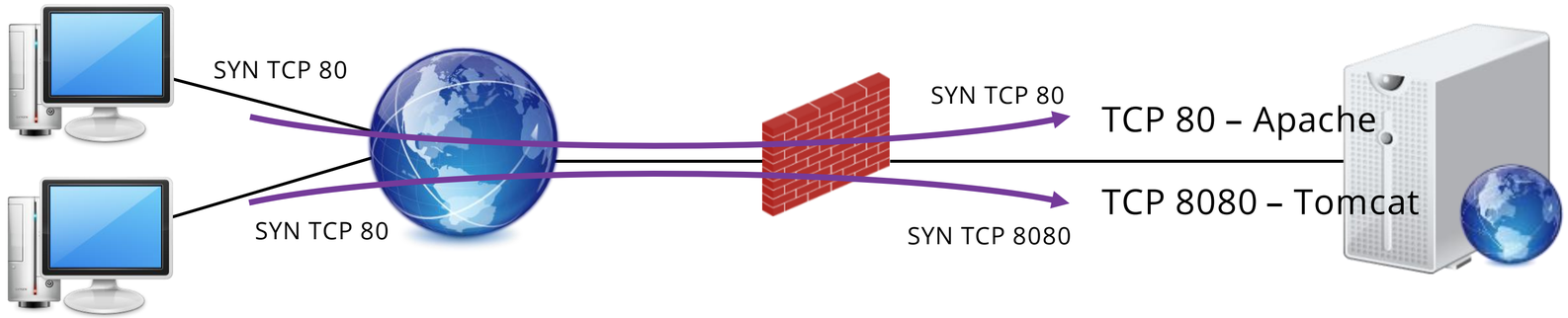
Table de suivi des connexions

@IP src	@IP dst	port src	port dst	@IP dst traduit	port traduit
27.26.25.10	27.26.25.24	10000	143	192.168.10.30	143



Traduction de port (PAT)

Il s'agit d'opérer une traduction au niveau du port (source ou destination). Cela permet par exemple d'exposer plusieurs services présents sur une même machine avec le même port apparent.



La redirection se fait selon les caractéristiques de la connexion (port de destination, IP source par exemple).

Organisation technique et politique de filtrage

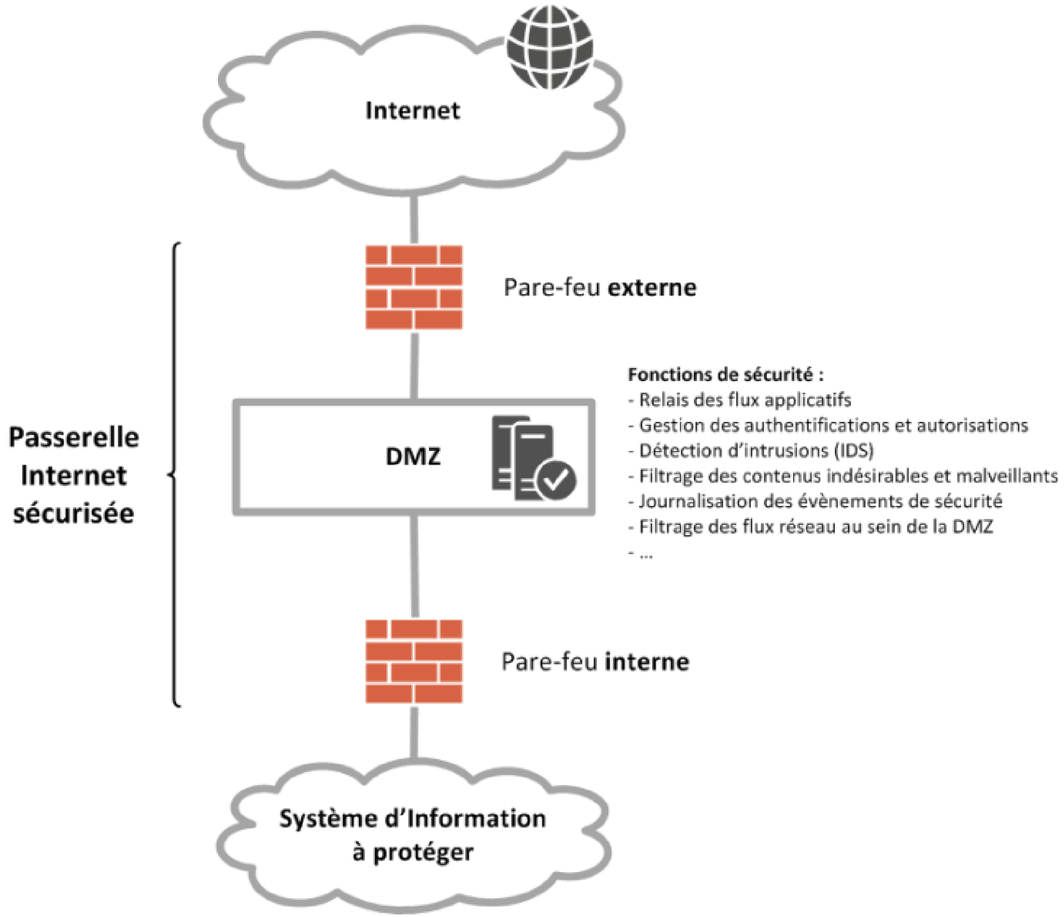
Double pare-feu

Les architectures ne mettant en œuvre qu'un seul ne présentent pas un niveau de sécurité satisfaisant (sauf pour les petites structures) :

- Une passerelle Internet sécurisée doit mettre en œuvre au moins deux pare-feux en cascade car une architecture basée sur un seul pare-feu, créé un point de défaillance dont la compromission expose l'ensemble des ressources du SI à protéger ;

- Il est recommandé de déployer des pare-feux présentant des différences technologiques à tous les niveaux : matériel, système d'exploitation, moteur de filtrage et IHM.
  ➔ Il faut garantir la maîtrise des différentes technologies par les équipes d'exploitation.

Double pare-feu



Physique vs. virtuel

Le filtrage réseau étant une fonction importante pour la sécurité d'un SI, il est recommandé de ne pas cumuler sur la même instance de pare-feu les fonctions de filtrage réseau avec d'autres fonctions de sécurité, afin de diminuer la surface d'attaque. De même, la virtualisation des systèmes augmente le nombre des risques pesant sur ceux-ci :

- La fuite d'information par manque de cloisonnement ou du fait d'une vulnérabilité inhérente au mécanisme de cloisonnement mis en œuvre par l'hyperviseur ;
- La compromission du pare-feu virtualisé ou de l'hyperviseur sous-jacent ;
- L'atteinte plus aisée à la disponibilité en cas de compromission, mais aussi du fait d'une erreur d'administration inhérente à la complexité accrue des tâches d'administration.

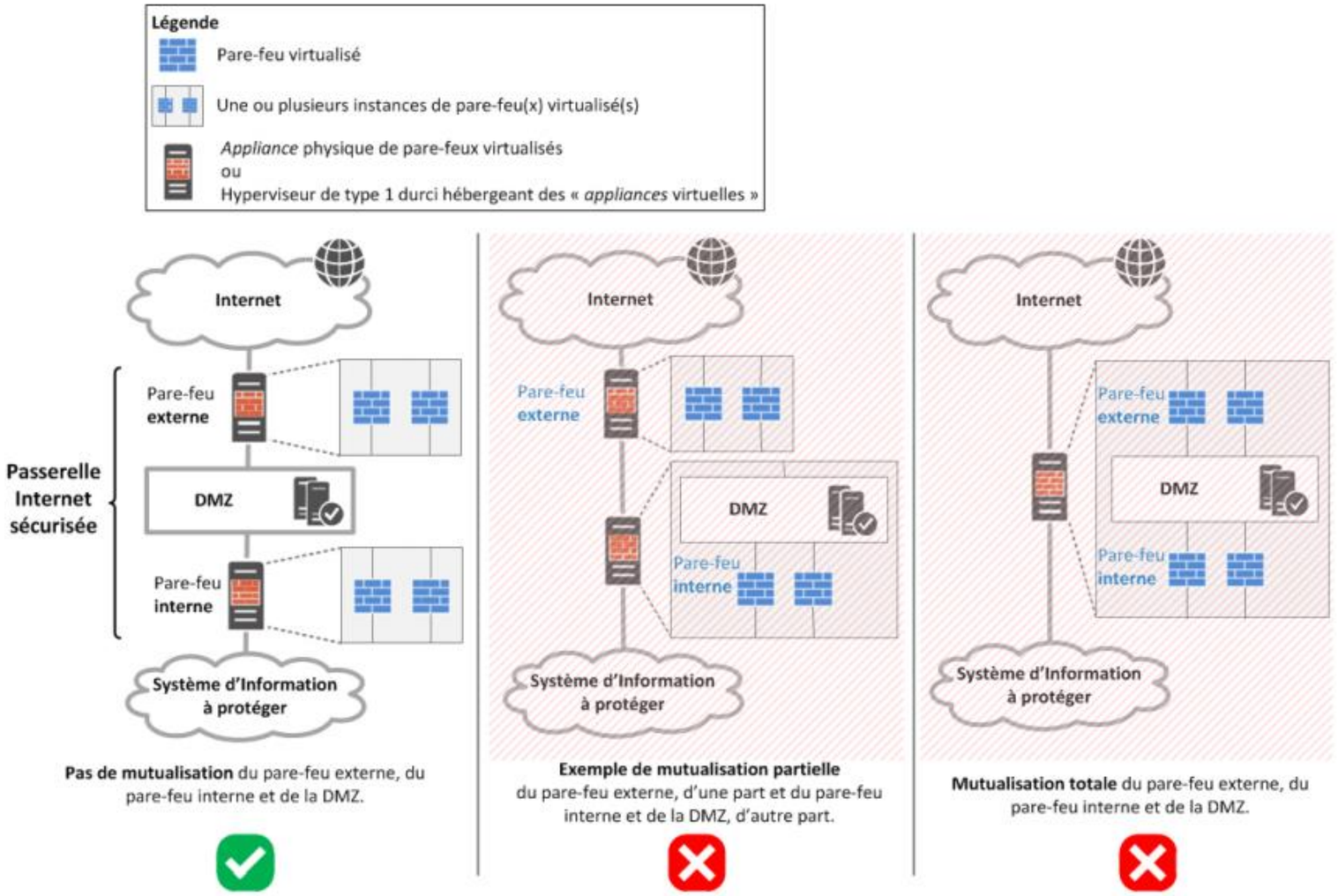
Physique vs. virtuel

La confiance pouvant être accordée aux mécanismes de cloisonnement est très difficile à démontrer car, en pratique, l'isolation entre des systèmes virtualisés n'est jamais complète, particulièrement du fait des ressources partagées de bas niveau.

De façon générale, la virtualisation des pare-feux des passerelles Internet sécurisées est déconseillée.

Séparer physiquement le pare-feu externe et le pare-feu interne si des solutions de virtualisation sont utilisées pour les pare-feux externe et/ou interne.

Physique vs. virtuel



Netfilter

Caractéristiques

- NetFilter est un pare-feu à état.
  - NetFilter assure 4 missions au sein du noyau :
  - politique de filtrage des flux entrants, sortants et rout és par un pare-feu ;
  - politique de traduction d'adresses en IPv4 ;
  - modification et marquage de paquets permettant de mettre en place la différenciation des flux dans l'optique d'une politique de routage intelligent (policy routing) ou d'assurer de la qualité de service (QoS).
  - marquage de flux en associant avec un contrôle d'accès obligatoire (Mandatory Control Access) de type SELinux permettant d'autoriser ou de refuser un flux dans le cadre d'une politique de sécurit é plus globale (en fonction des utilisateurs et des applications par exemple).
- NetFilter supportait initialement seulement la pile de protocoles IPv4.
- Son architecture a été reprise et étendue à la pile de protocole IPv6, aux couches ARP et MAC (Ethernet).

Tables

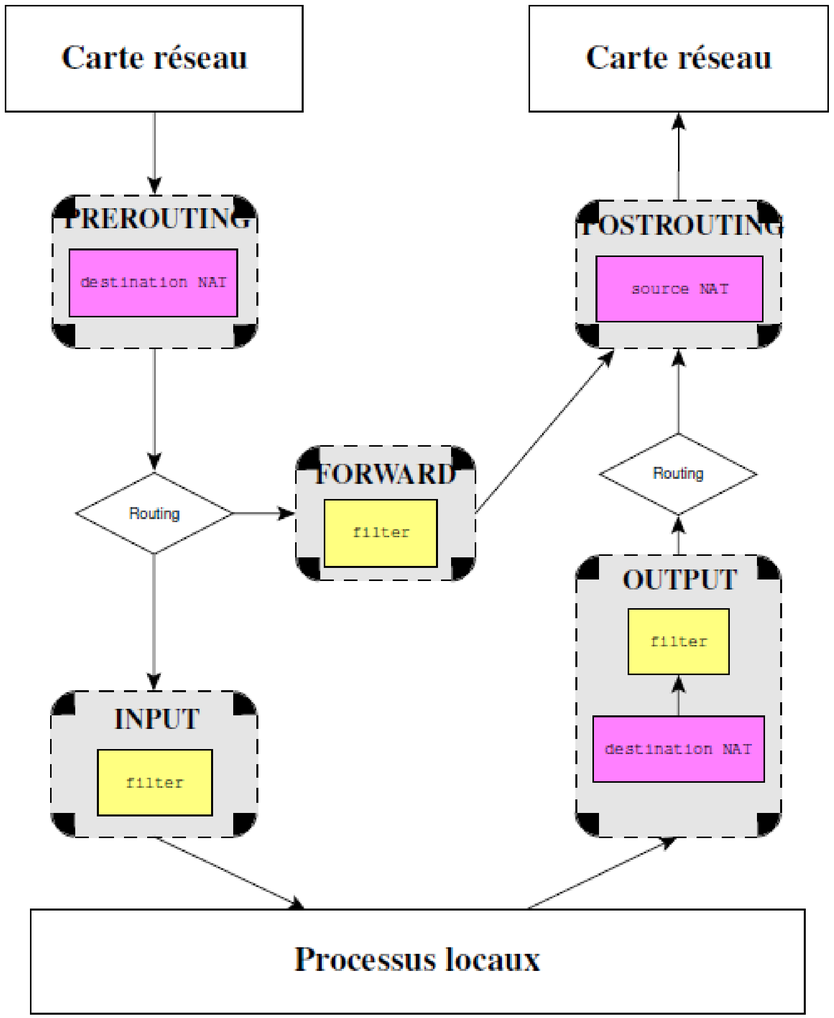
- Pour chacune des 4 missions énoncées précédemment, NetFilter utilise une table afin de stocker la politique associée :
    - **filter** : politique de filtrage ;
    - **nat** : politique de traduction d'adresses ;
    - **mangle** : modification et marquage des paquets ;
    - **security** : couplage avec une politique de contrôle d'accès obligatoire.
- Il existe une dernière table appelée *raw* qui permet des manipulations avant toutes celles permises par les autres tables.

Chaînes

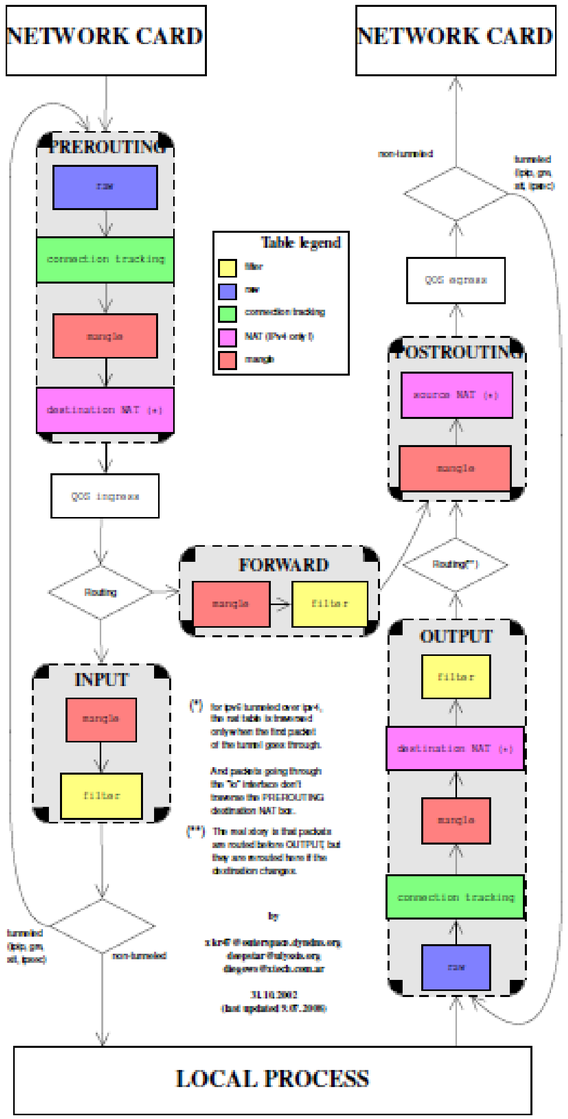
Afin d'intercepter les paquets lors de leur transit dans la pile IPv4, NetFilter définit un certain nombre de crochets (hook) au sein de la pile. Ces crochets sont appelés durant le parcours d'un paquet dans la pile. A chaque crochet on peut associer des règles qui peuvent filtrer ou accepter un paquet.

- **INPUT** : crochet associé aux paquets à destination de processus locaux au pare-feu (i.e. applications s'exécutant sur le pare-feu).
- **OUTPUT** : crochet associe aux paquets ´émis par des processus locaux au pare-feu (i.e. ´émis par des applications s'exécutant sur le pare-feu).
- **FORWARD** : crochet associe aux paquets transmis par le pare-feu (agissant comme un routeur).
- **PREROUTING** : crochet associe aux paquets entrant sur le pare-feu (avant consultation de la table de routage).
- **POSTROUTING** : crochet associé aux paquets transmis sortant du pare-feu (après consultation de la table de routage).

Cheminement d'un paquet dans NetFilter



Cheminement d’un paquet dans NetFilter (avec QoS)



Configuration

Pour configurer le pare-feu NetFilter, on peut utiliser un outil en ligne de commande appelé ***iptables***. Celui-ci permet de :

- ajouter des règles dans le pare-feu ;
- supprimer des règles existantes ;
- contrôler la politique par défaut (ouverte ou fermée) pour le filtrage ;
- configurer la politique de traduction d'adresses.

iptables n'est pas le seul outil pour configurer le pare-feu. C'est le plus complet en ligne de commandes. Mais il existe des outils graphiques qui permettent de le faire (par exemple Firewall Builder http://www.fwbuilder.org)

Règles

Chaque chaîne est composée d'un ensemble de règles (numérotées). Chaque règle est composée au minimum de :

- un filtre permettant de préciser les paquets auxquels devra s'appliquer la règle en question.
- une cible (**target**) précisant le sort des paquets pour lesquels le filtre précèdent s'applique.

Syntaxe des sélecteurs de paquets

- Protocole : `-p tcp, -p udp`
- Adresse source : `-s A.B.C.D[/masque]`
- Adresse destination : `-d A.B.C.D[/masque]`
- Port source : `--sport port`
- Port destination : `--dport port`
- Interface d'entrée : `-i ethX`
- Interface de sortie : `-o ethX`

Exemple : `-p tcp -d 172.16.1.10 --dport 80` (paquets TCP à destination de l'hôte 172.16.1.10 vers le port 80)

Syntaxe de la cible

- La cible est spécifiée à l'aide du mot-clé `-j` ou `- -jump`. Par exemple

```
-j cible [options de la cible]
```

La cible peut être :

- **ACCEPT** : Acceptation du paquet. Les règles suivantes ne sont pas examinées si le sélecteur est activé.
- **DROP** : Destruction silencieuse du paquet. Les règles suivantes ne sont pas examinées si le sélecteur est activé.
- **REJECT** : Destruction du paquet avec émission d'un paquet ICMP vers l'émetteur. Les règles suivantes ne sont pas examinées si le sélecteur est activé. Cette cible accepte une option permettant de préciser le type de paquet ICMP à retourner :

```
--reject-with type
```

Syntaxe de la cible

- **SNAT** : Traduction d'adresse source dont l'option est :

```
--to-source IP1[-IP2][:port1[-port2]]
```

- **MASQUERADE** : Idem mais avec détermination dynamique de l'adresse de traduction (celle de l'interface par laquelle le paquet va sortir).
- **DNAT** : Traduction de l'adresse de destination dont l'option est :

```
--to-destination IP1[-IP2][:port1[-port2]]
```

- **REDIRECT** : Redirection vers un processus local écoutant sur un port particulier. L'option est :

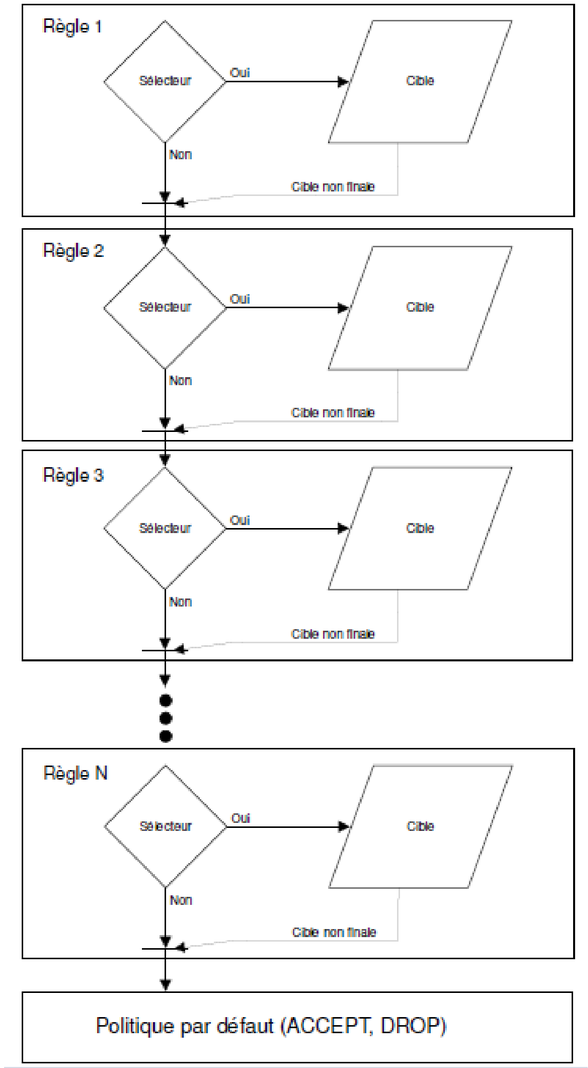
```
--to-ports port1[-port2]
```

Syntaxe de la cible

- **LOG** : permet d'enregistrer certaines informations à propos du paquet dans les journaux du système. Ceci ne présume pas du sort du paquet car les règles suivantes sont examinées. De nombreuses options sont prévues (voir la page de manuel).
- **QUEUE** : Permet de transférer le paquet vers l'espace utilisateur où un processus doit ˆ etre en attente des paquets et d´décider de leur sort.
- **TEE** : Permet de faire une copie du paquet et de la diriger vers une autre machine. L'option est :

```
--gateway IP
```

Fonctionnement d'une chaîne



Chaîne utilisateur

Il est possible de regrouper un ensemble de règles dans une chaîne dédiée.

- Une telle chaîne est appelée chaîne utilisateur.
- Chacune de ces chaînes possède un nom qui lui est propre.
- Il est possible d'utiliser le nom d'une telle chaîne comme cible :

```
-j chaine-utilisateur
```

- Ceci est ´équivalent à un appel de fonction.

Chaîne utilisateur

Les règles de la chaîne utilisateur sont inspectées jusqu'à ce que soit :

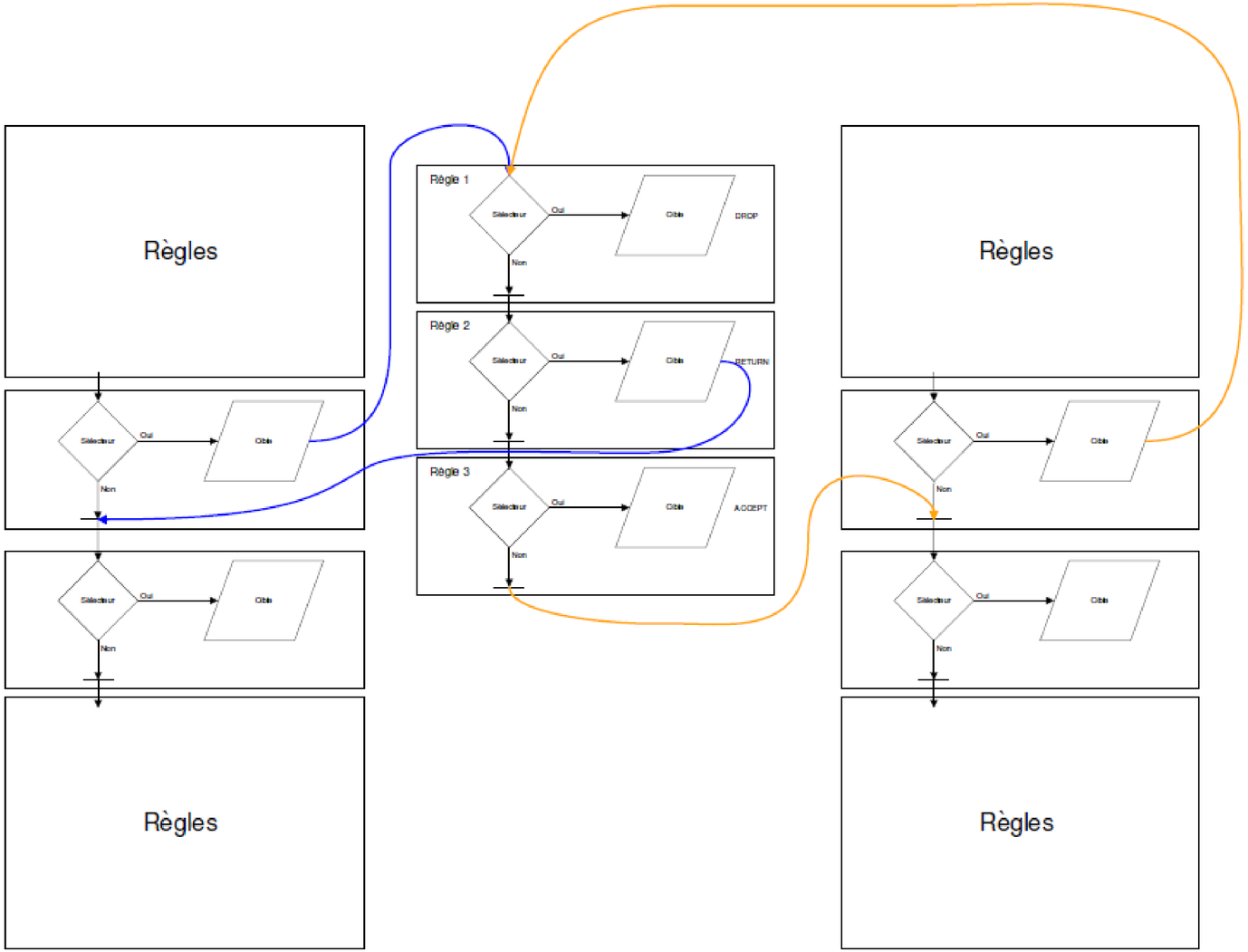
- une règle détermine le sort du paquet (**DROP** ou **ACCEPT**) ;
- une règle appelle la cible **RETURN**. Dans ce cas le parcours des règles est terminé dans la chaîne utilisateur et reprend dans la chaîne appelante, après la règle qui a appelé la chaîne utilisateur ;
- la fin de la chaîne utilisateur est atteinte. Tout se passe comme s'il y avait une règle implicite :

```
-j RETURN
```

Il est possible de simuler le comportement d'un *goto* plutôt qu'un appel de fonction par :

```
-g chaîne-utilisateur
```

Chaîne utilisateur



Syntaxe des commandes

Ajout d'une règle en fin de chaîne

```
iptables [-t table] -A chaine [ selecteur ] [-j cible [options de cible]]
```

Insertion d'une règle à une position précise dans une chaîne

```
iptables [-t table] -I chaine position [ selecteur ] [-j cible [options de cible]]
```

Suppression d'une règle à une position précise dans une chaîne

```
iptables [-t table] -D chaine position
```

Listage des règles d'une table/chaîne

```
iptables [-t table] -L [chaine] [-n] [-v]
```

Réglage de la politique de sécurité par défaut d'une chaîne

```
iptables [-t table] -P chaine [ACCEPT j DROP]
```

Syntaxe des commandes

Ajout d'une chaîne utilisateur dans une table

```
iptables [-t table] -N chaine
```

Suppression des règles dans une chaîne/table

```
iptables [-t table] -F [chaine]
```

Suppression d'une chaîne utilisateur (qui doit être vide)

```
iptables [-t table] -X [chaine]
```

Extensions

NetFilter possède une architecture extensible. Une extension peut concerner :

- la sélection de paquet (pouvoir sélectionner des paquets selon des critères plus complexes) ;
- les cibles.

Toute extension est divisée en deux parties modulaires :

- un module noyau qui implémente la fonctionnalité dans le noyau dont le nom de fichier est `xt_extension.ko` ou `xt_CIBLE.ko` ;
- une librairie partagée qui permet de configurer l'extension depuis *iptables* (située en général sous `/lib/xtables/libxt`).

Le module noyau doit ´évidemment avoir été charge au préalable dans le noyau afin de pouvoir l'utiliser.

Syntaxe complète de la sélection de paquet :

```
[[-p protocole] [-d IP[/masque]] [-s IP[/masque]] [--sport port] [--dport port]]
                                ([-m extension [options]])?
```

Suivi des connexions

NetFilter est un pare-feu avec état (*stateful*). Cette fonctionnalité est activée par le biais d'un module supplémentaire. Ce module est appelé `state` (*iptables* version < 1.4.16), ou `conntrack` (iptables version >= 1.4.16). Etats possibles pour une connexion :

- **NEW** : le paquet ´établi une nouvelle connexion.
- **ESTABLISHED** : le paquet appartient à une connexion existante.
- **RELATED** : le paquet démarre une nouvelle connexion en relation avec une connexion existante (protocoles complexes tels que FTP, SIP, etc).
- **INVALID** : le paquet n'est associé à aucune connexion connue.
- d'autres états sont possibles avec le module `conntrack`.

Suivi des connexions

Connexions entrants sur un serveur Web hébergé par le pare-feu lui-même.

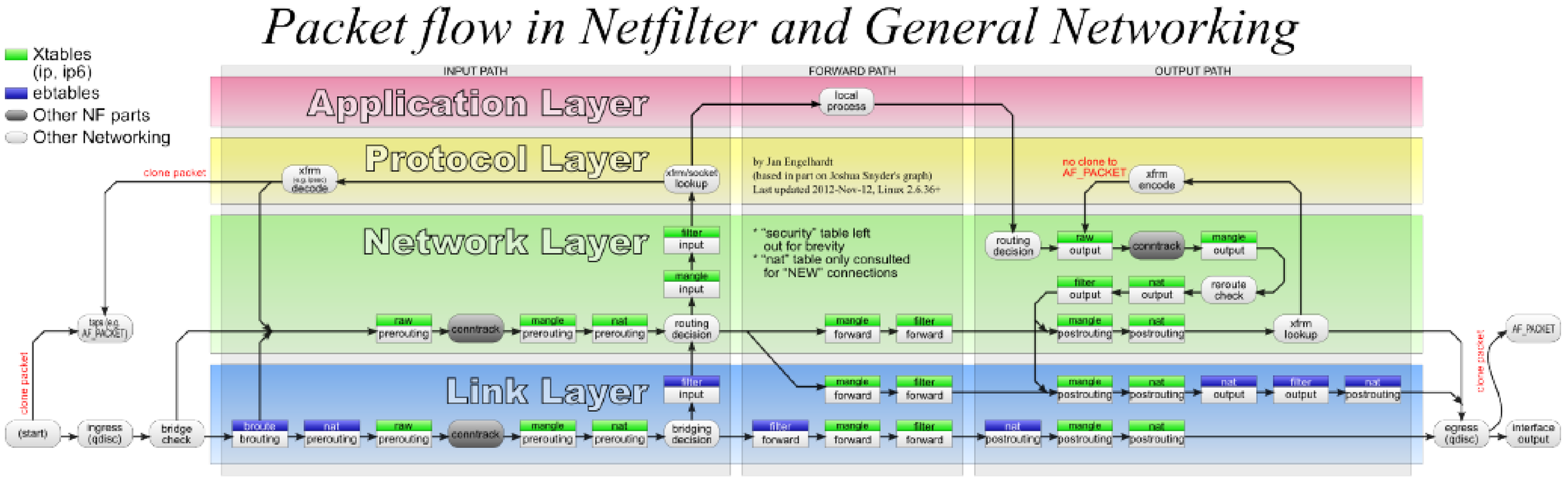
Exemple (module state)

```
iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

Exemple (module conntrack)

```
iptables -A INPUT -p tcp --dport 80 -m conntrack --ctstate NEW,ESTABLISHED -j
ACCEPT

iptables -A OUTPUT -p tcp --sport 80 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

The big picture



Protocoles complexes

Certains protocoles complexes utilisent à la fois un canal de commandes (habituellement utilisant le protocole TCP sur un port bien défini), ainsi qu'un ou plusieurs canaux auxiliaires (en TCP/UDP) en utilisant des ports n´négocies dynamiquement dans le canal de contrôle.

Exemples

- FTP : canal de contrôle sur le port 20 et transfert de données entre le client et le serveur par TCP sur un port n´négocie dynamiquement.
- SIP canal de contrôle en TCP sur le port 5060, puis conversation en UDP sur des ports n´négocies dynamiquement.

Protocoles complexes

Ce que permet de faire NetFilter :

- Le système de suivi de connexion supporte l'insertion d'une connexion dont on s'attend à ce qu'elle soient ouverte dans un futur proche (état `expected`).
- Un module auxiliaire capable de comprendre les ´échanges d'un protocole complexe peut donc ajouter à la volée les connexions attendues.
- De tels modules auxiliaires existent pour les protocoles : FTP, H323, IRC, PPTP, SANE, SIP, SNMP, TFTP…

Avantages : Cela permet d'écrire une politique de sécurité pour des protocoles complexes.

Inconvénients : Le noyau n'est pas le meilleur endroit où parser des protocoles complexes. Par ailleurs ceci autorise l'ouverture de connexions implicites et peut ´éventuellement être détourné par un attaquant.

Protocoles complexes

Ce que permet de faire NetFilter :

- Le système de suivi de connexion supporte l'insertion d'une connexion dont on s'attend à ce qu'elle soient ouverte dans un futur proche (état `expected`).
- Un module auxiliaire capable de comprendre les ´échanges d'un protocole complexe peut donc ajouter à la volée les connexions attendues.
- De tels modules auxiliaires existent pour les protocoles : FTP, H323, IRC, PPTP, SANE, SIP, SNMP, TFTP...

Avantages : Cela permet d'écrire une politique de sécurité pour des protocoles complexes.

Inconvénients : Le noyau n'est pas le meilleur endroit où parser des protocoles complexes. Par ailleurs ceci autorise l'ouverture de connexions implicites et peut ´éventuellement être détourné par un attaquant.

NetFilter est partout

Boeing 777-300R de la Singapore AirLines.

(Blog de Harald Welte) http://gnumonks.org/~laforge/photos/linux_netfilter_singapore_entertainment.jpg

acceis

Merci de votre attention.

Gwenn Feunteun

gwenn@acceis.fr