

Parefeux

Guillaume Hiet

guillaume.hiet@supelec.fr

D'après les supports de Frédéric Tronel et Laurent Heye

Equipe CIDRE, SUPELEC

Septembre 2017

Notes

- 1 Parefeux
 - 2 Guide de survie pour NetFilter
 - 3 Technologies
 - 4 Architecture
 - 5 Politiques de filtrage

Notes

- Filtrage de paquets basé sur une politique de sécurité exprimant des contraintes en terme de :
 - type de protocoles (UDP, TCP, ICMP, IGMP, SCTP, etc) ;
 - adresses source et destination ;
 - ports source et destination ;
 - contenu de la *payload*.
 - Rôles supplémentaires :
 - Traduction d'adresses (*NAT*).
 - Relais applicatif (*proxy*).
 - Authentification d'utilisateurs.
 - Recherche de virus.
 - Détection/prévention d'intrusions.

Notes

- Politique ouverte
 - Par défaut tout est permis.
 - La politique exprime les flux réseau interdits.
 - Tout ce qui n'est pas explicitement interdit est autorisé
 - Politique fermée
 - Par défaut tout est interdit.
 - La politique exprime les flux réseau autorisés.
 - Tout ce qui n'est pas explicitement autorisé est interdit.
 - Il est évidemment conseillé de travailler avec une politique **fermée**.

Notes

Les flux interdits ou autorisés sont exprimés en terme de :

- Protocoles.
 - Adresses IP source et destination.
 - Ports source et destination.
 - Sens de circulation du paquet.

Le sort d'un paquet est décidé indépendamment de tous les paquets déjà vus (pas d'état dans le parefeu).

Notes

Il faut une règle pour chaque sens de circulation. Par exemple pour autoriser les clients d'un réseau local à atteindre les serveurs HTTP, il faut :

- Une règle acceptant les paquets sortants du réseau TCP à destination du port 80.
 - Une règle acceptant les paquets TCP entrant dans le réseau provenant du port 80.

Attaque possible

Un attaquant peut forger des paquets contenant une charge malveillante et les faire entrer dans le réseau, en leur attribuant comme port source le port 80.

Notes

- Pour pallier le problème précédent il est nécessaire de pouvoir suivre les connexions TCP.
 - Un parefeu capable de suivre l'état des connexions réseau est dit **à états** (*stateful*).
 - Pour reprendre l'exemple précédent, il faut les règles suivantes :
 - Accepter les paquets TCP sortants du réseau à destination du port 80 qui créent une nouvelle connexion (paquet SYN).
 - Accepter les paquets TCP sortants du réseau à destination du port 80 qui appartiennent à une connexion existante.
 - Accepter les paquets TCP entrant dans le réseau provenant du port 80 et qui appartiennent à une connexion existante ou établissant une connexion (paquet SYN/ACK).
 - Pour cela il faut maintenir une table résumant les connexions existantes.

Notes

Une connexion TCP est complètement 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.

Conséquence

Un parefeu à états doit maintenir une table avec toutes les connexions TCP en cours.

Notes

Constatation

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).

Notes

Exemple du suivi de connexion sous Linux

IPTState - IPTables State Top					
Version: 2.2.5	Sort: DstIP	b: change sorting	h: help	Prt	State
Source		Destination		TTL	
192.168.0.134:55661		54.240.172.61:80		tcp	TIME_WAIT 0:00:38
192.168.0.134:51530		74.125.230.205:80		tcp	TIME_WAIT 0:00:37
192.168.0.134:51532		74.125.230.205:80		tcp	TIME_WAIT 0:00:37
192.168.0.134:55097		74.125.230.206:80		tcp	TIME_WAIT 0:00:38
192.168.0.134:44999		74.125.230.219:80		tcp	TIME_WAIT 0:00:36
192.168.0.134:55260		74.125.230.220:80		tcp	TIME_WAIT 0:00:37
192.168.0.134:53763		91.103.142.194:80		tcp	TIME_WAIT 0:00:38
127.0.0.1:45980		127.0.0.1:143		tcp	ESTABLISHED 119:59:50
127.0.0.1:45452		127.0.0.1:143		tcp	ESTABLISHED 119:59:34
127.0.0.1:45979		127.0.0.1:143		tcp	ESTABLISHED 119:59:59
192.168.0.134:41310		192.70.40.1:993		tcp	ESTABLISHED 119:59:59
192.168.0.134:34103		193.51.193.146:993		tcp	ESTABLISHED 119:59:50
192.168.0.134:59132		193.51.224.48:80		tcp	TIME_WAIT 0:00:36
192.168.0.134:53472		217.109.67.162:80		tcp	TIME_WAIT 0:00:38
192.168.0.134:53470		217.109.67.162:80		tcp	TIME_WAIT 0:00:38
192.168.0.134:53471		217.109.67.162:80		tcp	TIME_WAIT 0:00:38
192.168.0.117:5353		224.0.0.251:5553		udp	0:00:11

Copie d'écran de la sortie de l'outil `iptstate` montrant le suivi de connexions pour les protocoles TCP et UDP

Notes

- 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
 - ① d'accéder à des services situés sur Internet (connexions sortantes) ;
 - ② 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 sécurité du parefeu.

Note

Attention ceci est en contradiction avec le paradigme de TCP/IP qui se veut être une connexion de bout en bout (le client et le serveur étant les deux équipements intelligents). Ici la connexion est séparée en deux parties.

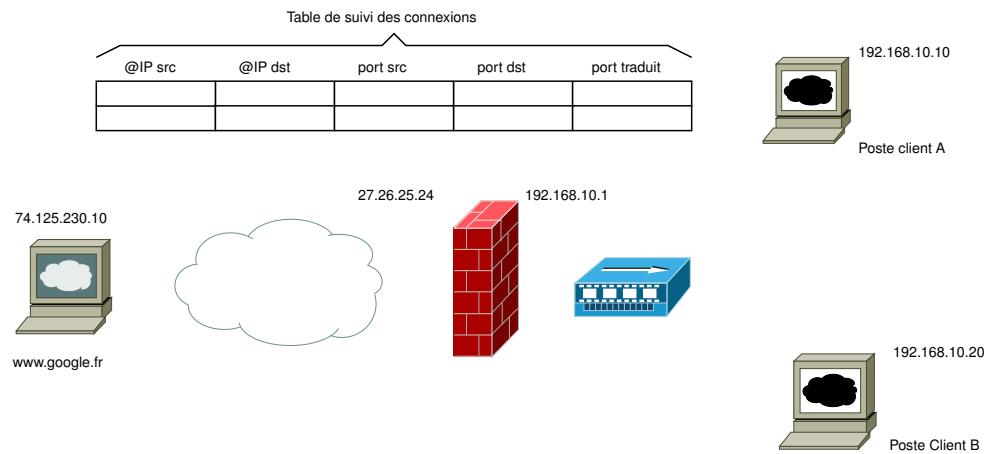
Notes

- 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.

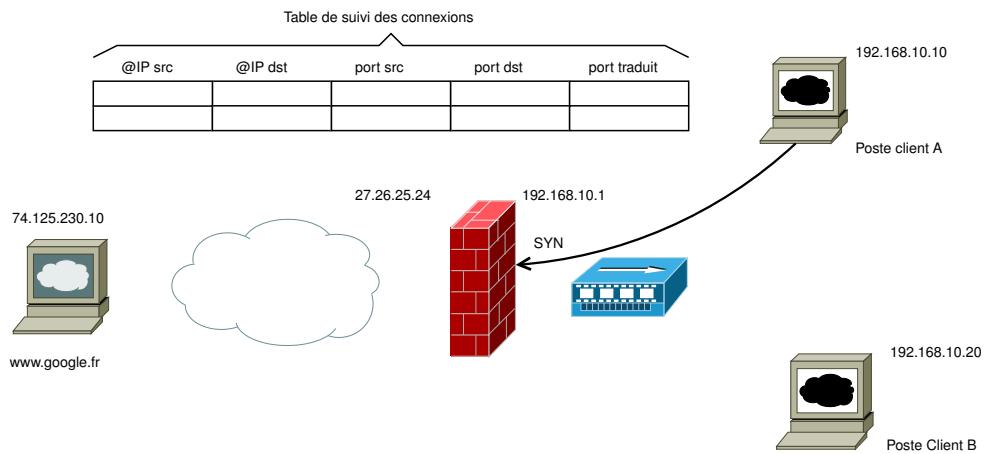
Remarque

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

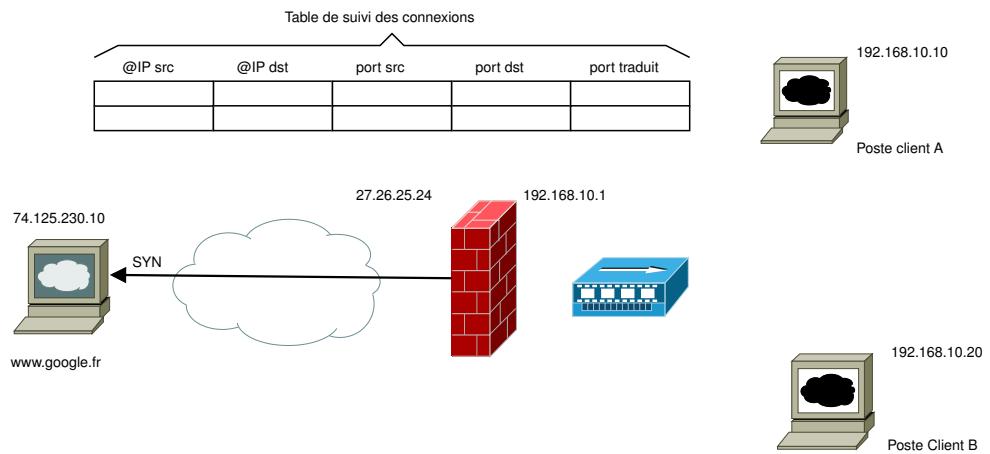
Notes



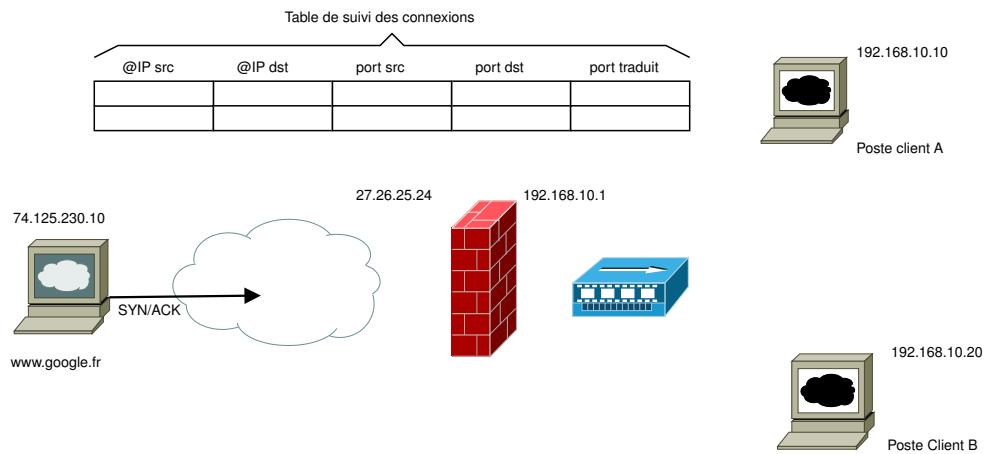
Notes



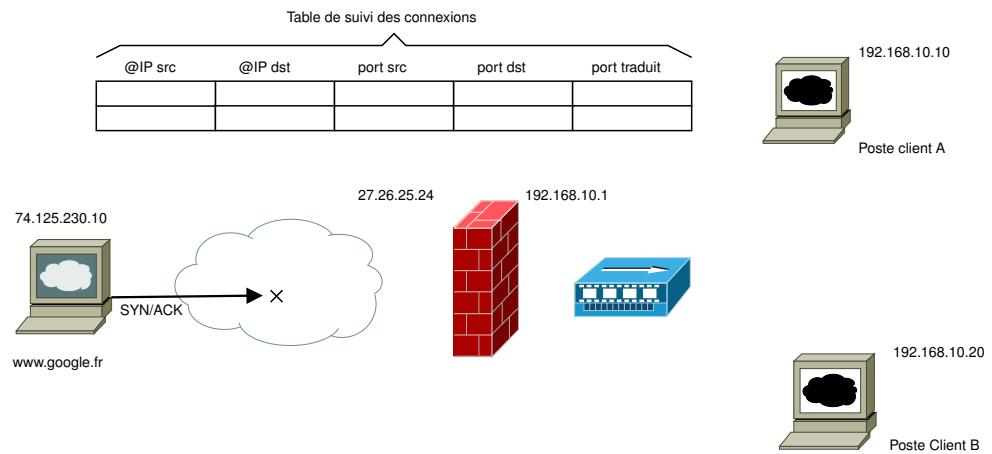
Notes



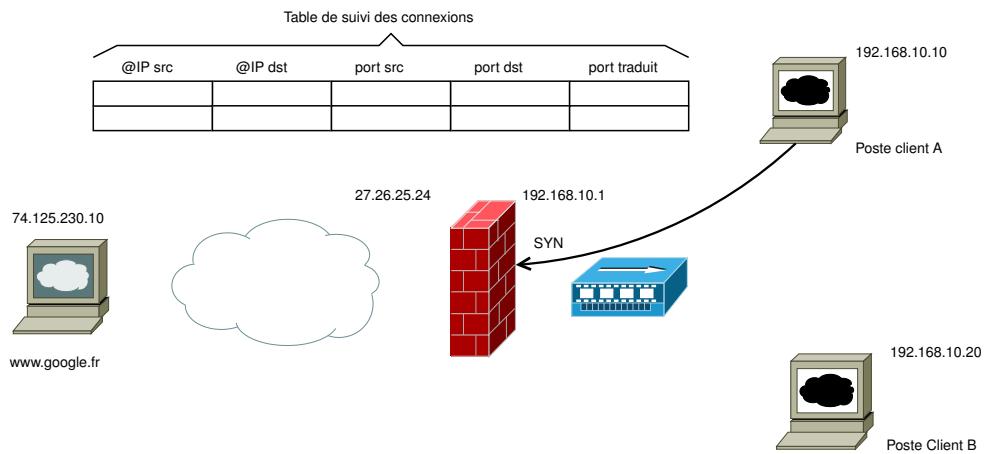
Notes



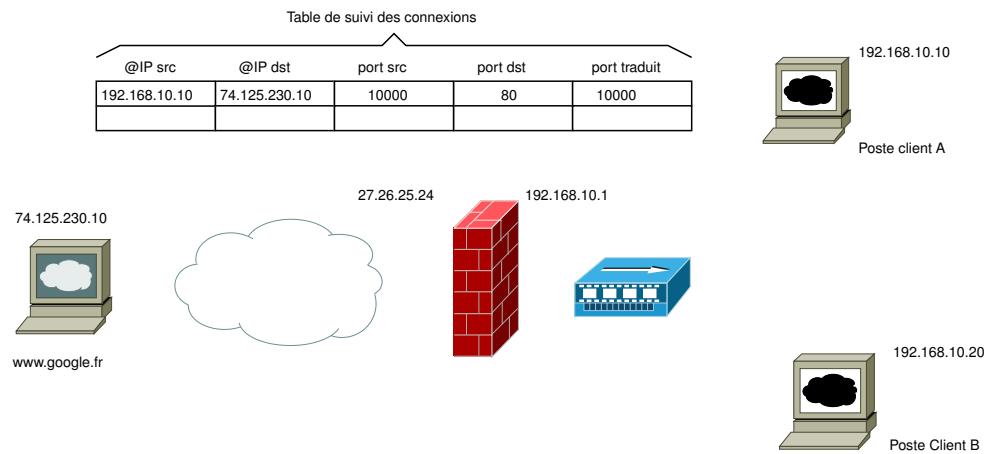
Notes



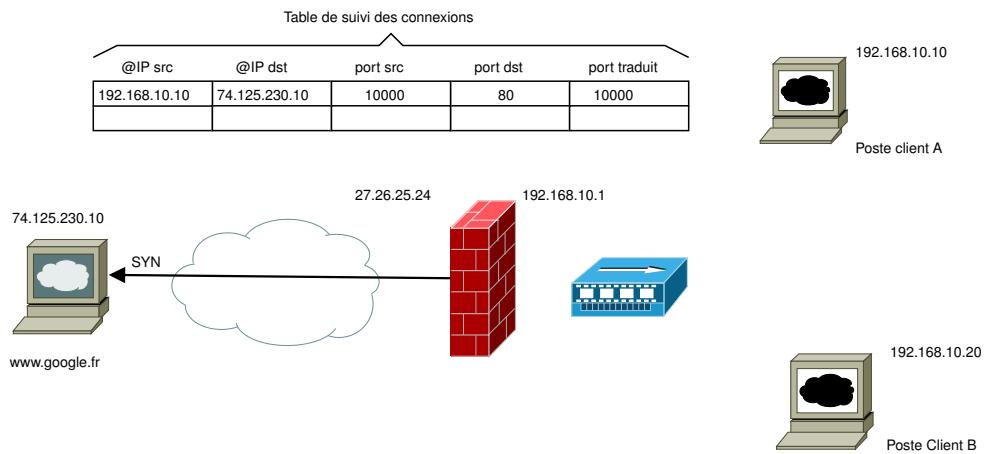
Notes



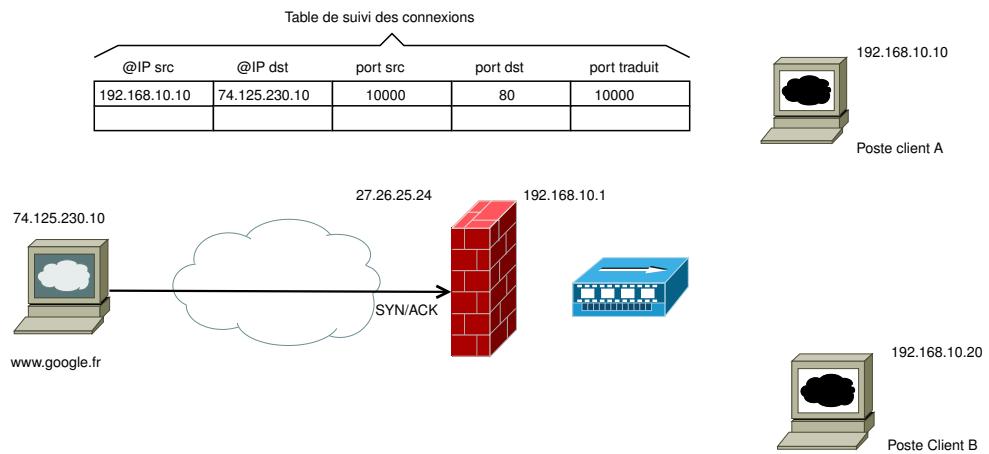
Notes



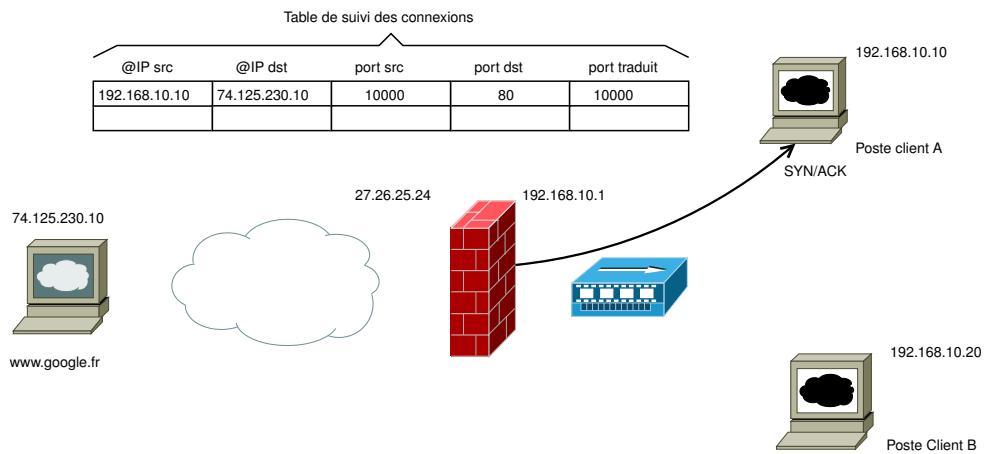
Notes



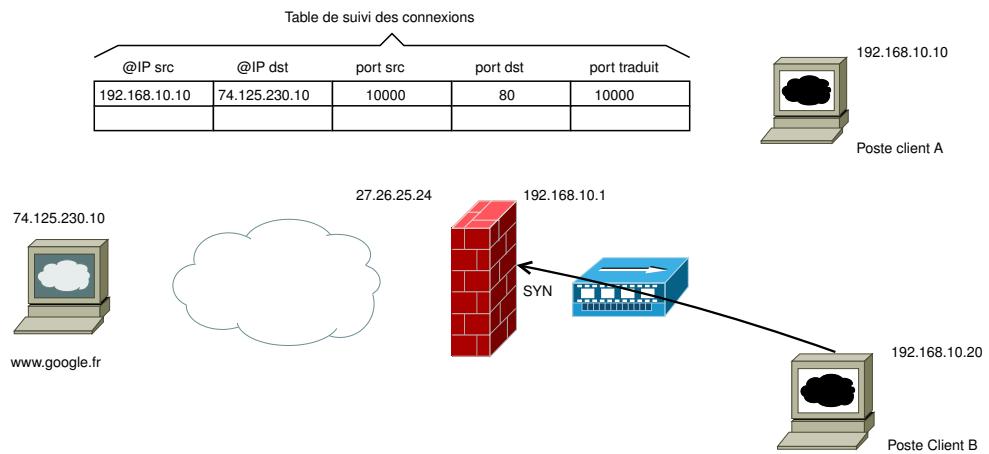
Notes



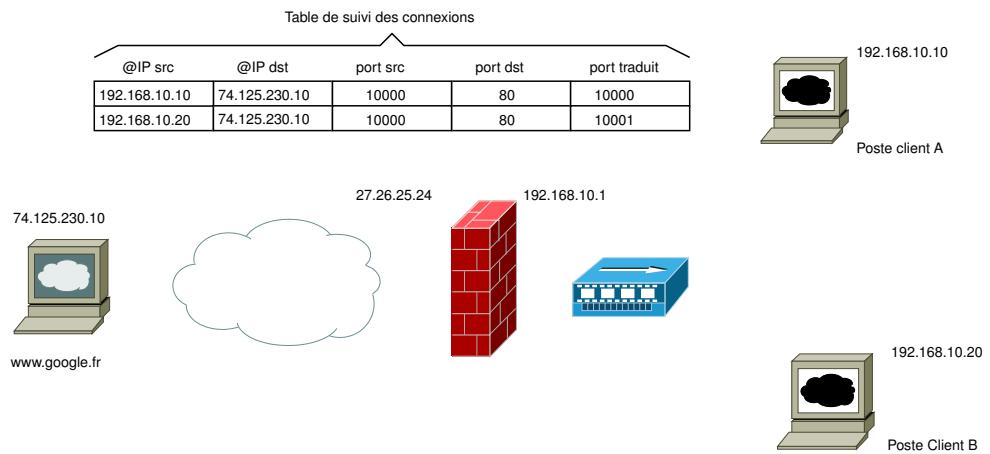
Notes



Notes



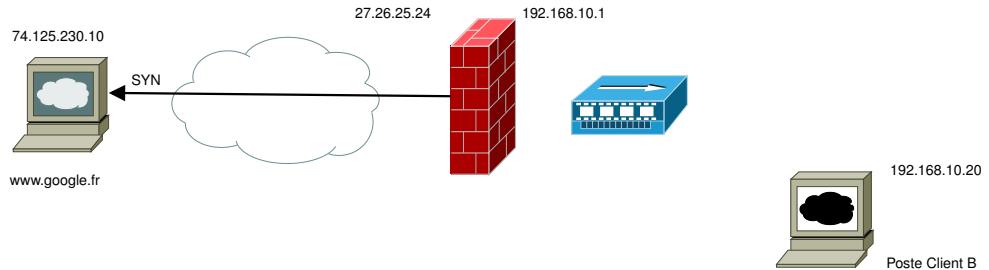
Notes



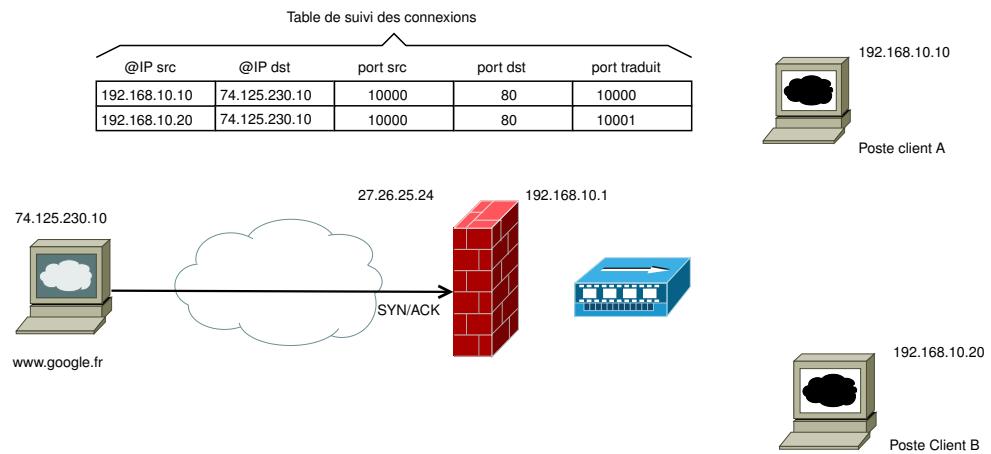
Notes

Table de suivi des connexions

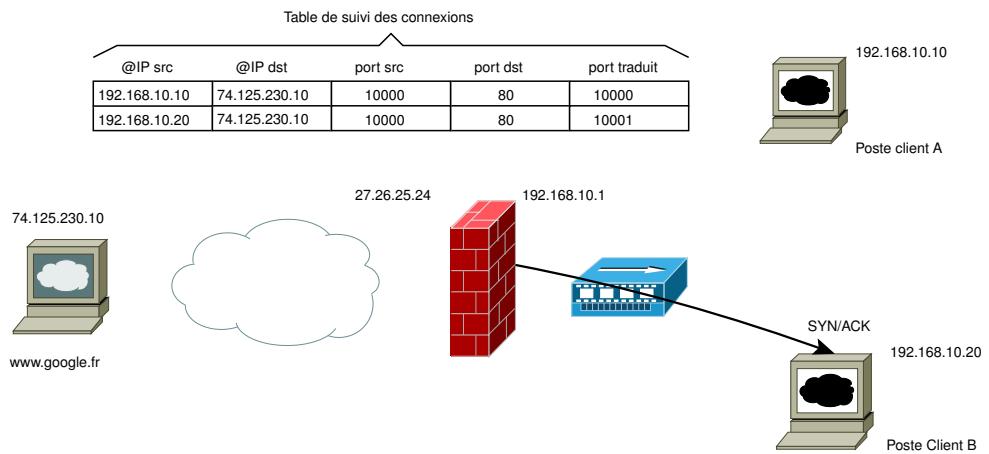
@IP src	@IP dst	port src	port dst	port traduit
192.168.10.10	74.125.230.10	10000	80	10000
192.168.10.20	74.125.230.10	10000	80	10001



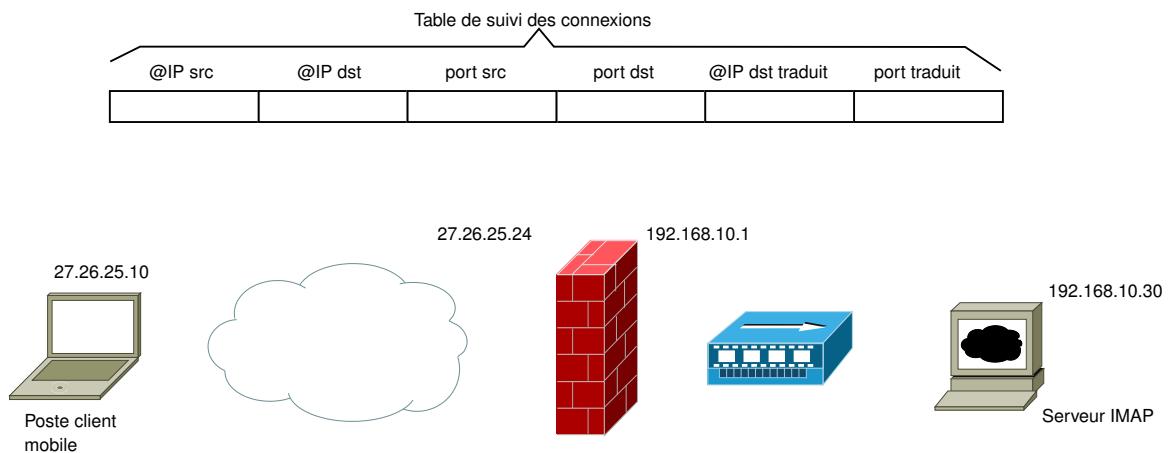
Notes



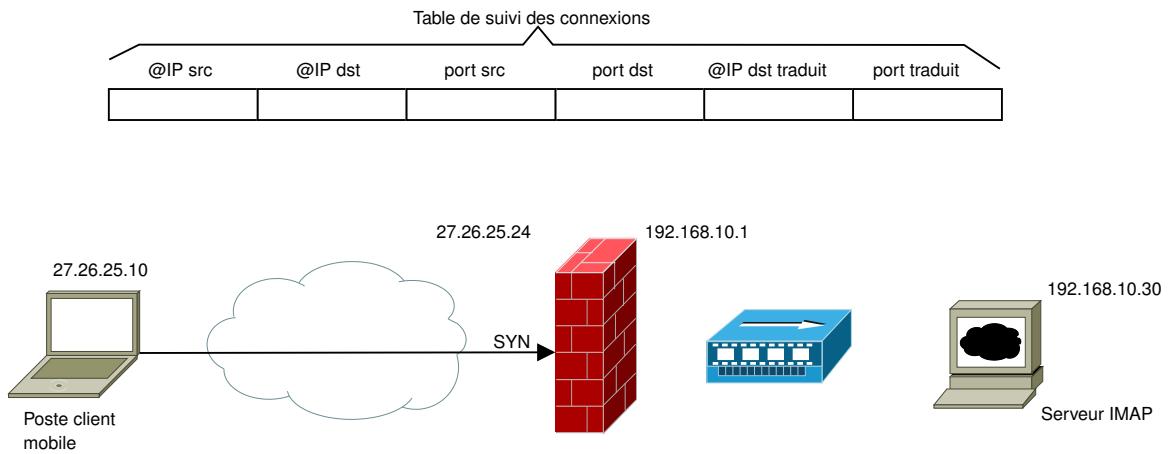
Notes



Notes

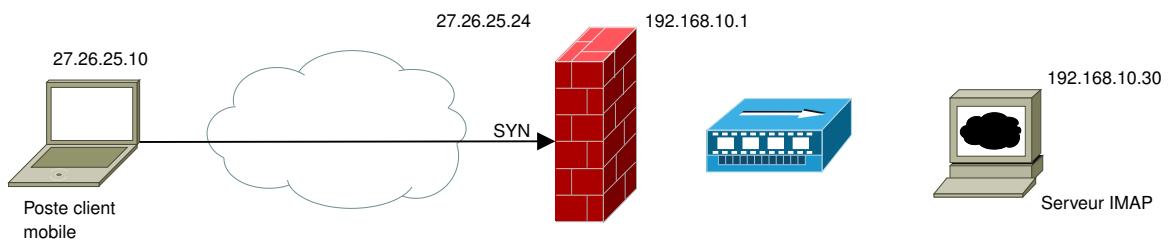


Notes



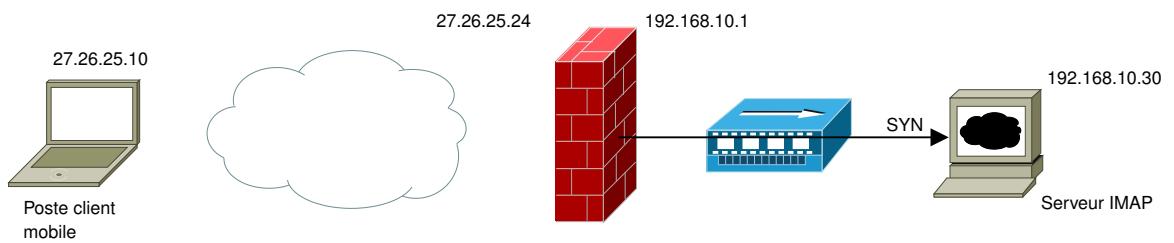
Notes

Table de suivi des connexions



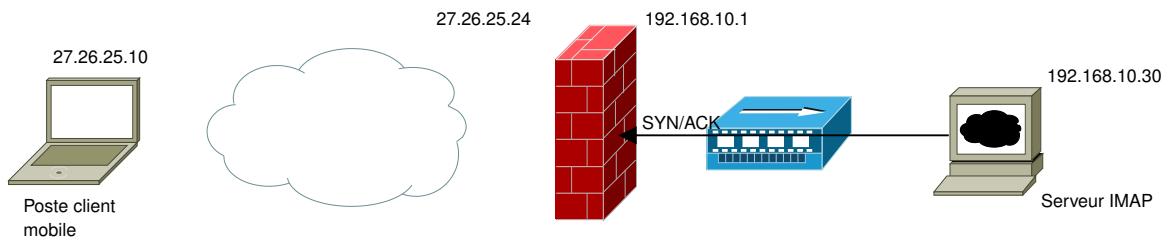
Notes

Table de suivi des connexions



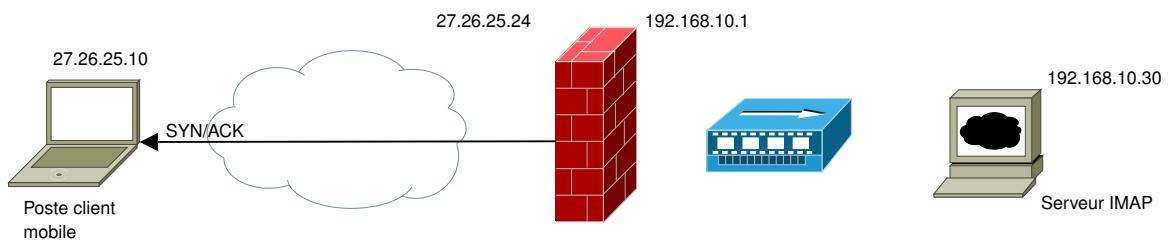
Notes

Table de suivi des connexions



Notes

Table de suivi des connexions



Notes

- NetFilter est le parefeu par défaut du noyau Linux depuis le noyau 2.4.x
 - Il fait suite aux projets *ipchains* (noyau 2.2.x) et *ipfwadm* (noyau 2.0.x)
 - Le projet *ipfwadm* était un dérivé de *ipfw* qui est une implémentation de référence d'un parefeu pour le projet BSD.
 - La version de NetFilter pour un noyau Linux 2.6.27 ainsi que l'utilitaire d'administration du parefeu *iptables* en version 1.4.2 sont certifiés au premier niveau du CSPN (ANSSI) depuis le 31 août 2009.

Notes

- NetFilter est un parefeu à état.
 - NetFilter assure 4 missions au sein du noyau :
 - politique de filtrage des flux entrants, sortants et routés par un parefeu.
 - politique de traduction d'adresses en IPv4 ;
 - modification et marquage de paquets permettant de mettre en place la différenciation des flux (*Diff Serv*) 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 sécurité 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).
 - Dans ce cours nous n'aborderons que ce qui concerne le filtrage de paquets et la traduction d'adresses pour IPv4.

Notes

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.

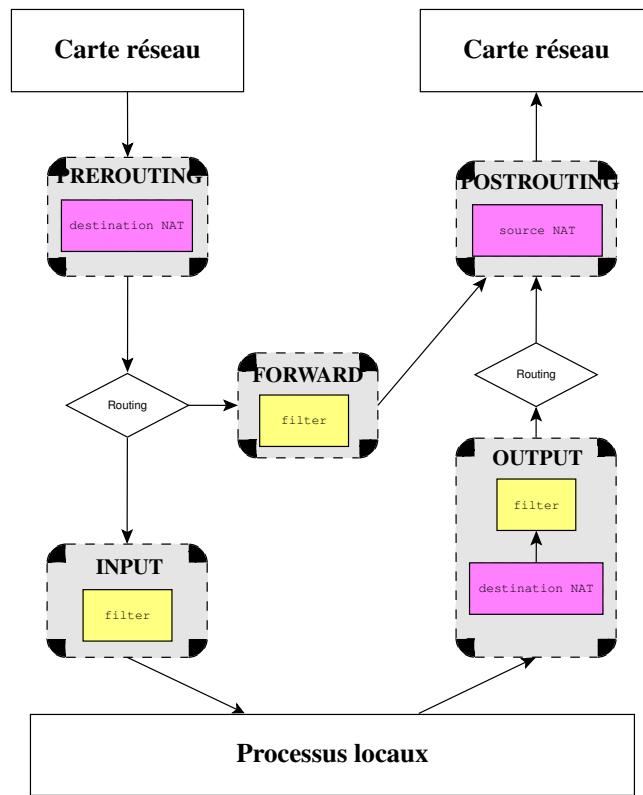
Il existe une dernière table appelée *raw* qui permet des manipulations avant toutes celles permises par les autres tables.

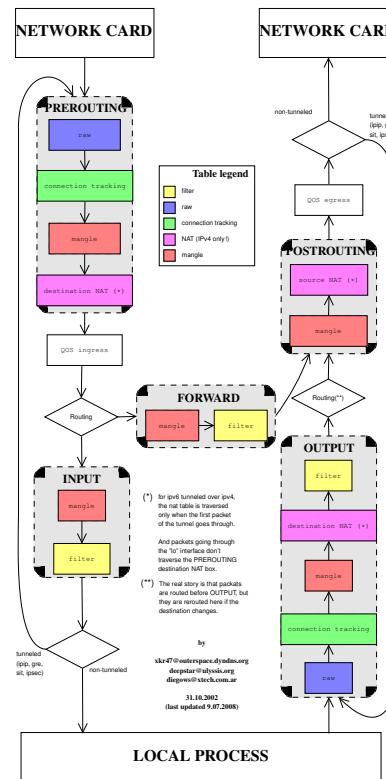
Notes

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

Notes





http://xkr47.outerspace.dyndns.org/netfilter/packet_flow/

Notes

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

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

Note

iptables n'est pas le seul outil pour configurer le parefeu. 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>)

Notes

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 précédent s'applique.

Notes

- 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)

Notes

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

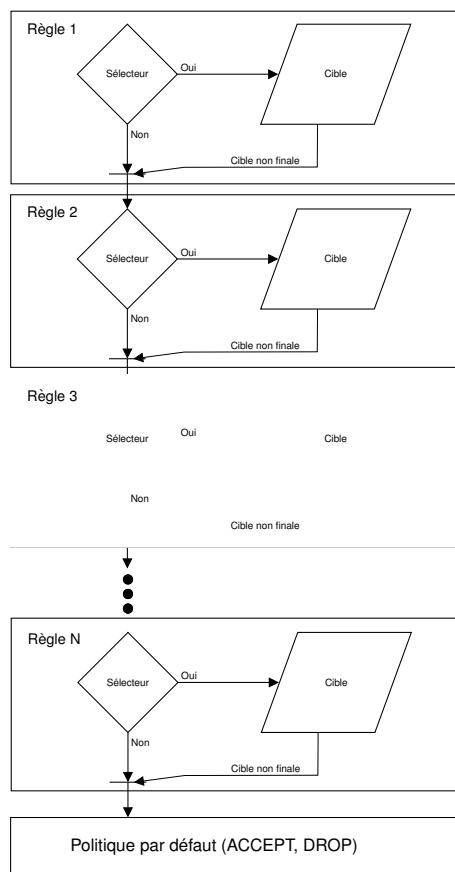
Notes

- 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]

- 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 être en attente des paquets et 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

Notes

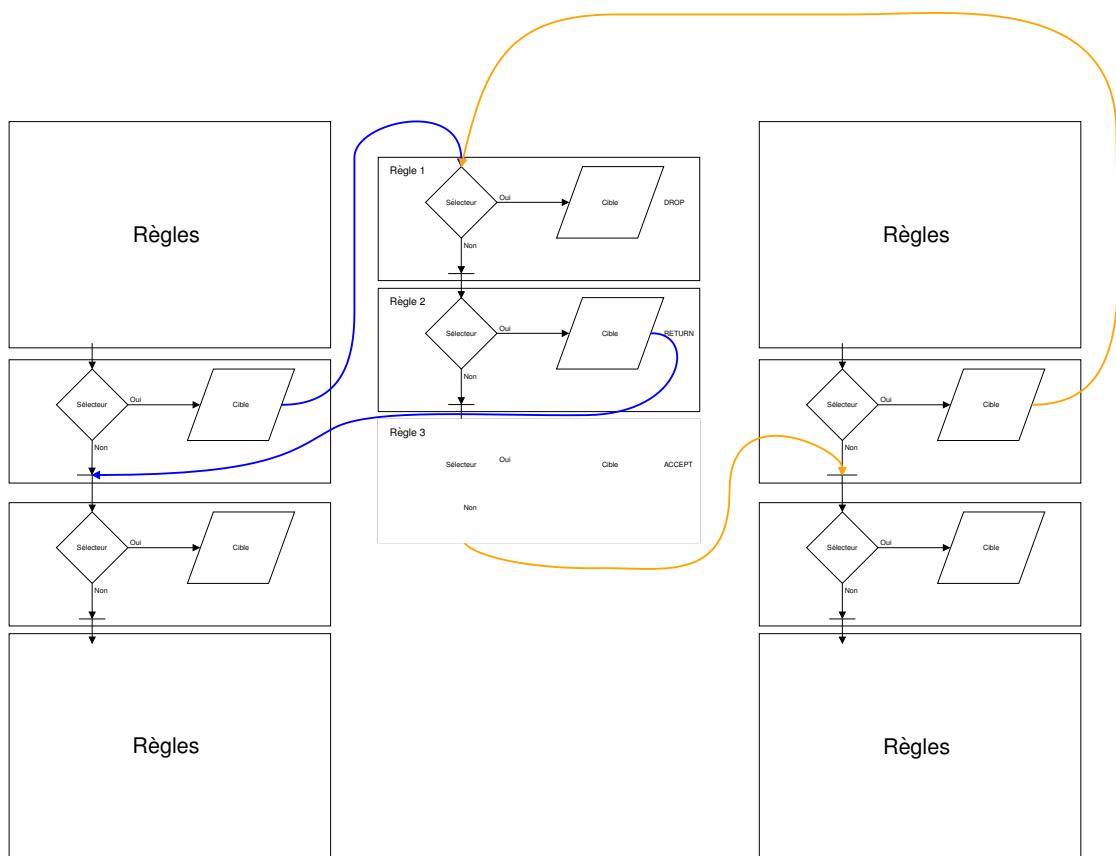


- Il est possible de regrouper un ensemble de règles dans une chaîne dédiée.
 - Une telle chaîne est appelée chaîne utilisateur.
 - Chacune de ces chaînes possède un nom qui lui est propre.
 - Il est possible d'utiliser le nom d'une telle chaîne comme cible :
-j chaine-utilisateur
 - Ceci est équivalent à un appel de fonction.

Notes

- 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

Notes



Ajout d'une règle en fin de chaîne

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

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

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

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

iptables [-t table] -D chaîne 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 chaîne [ACCEPT | DROP]
```

Notes

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]

Notes

NetFilter possède une architecture extensible. Une extension peut concerter :

- 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/iptables/libxt`).

Le module noyau doit évidemment avoir été chargé 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]])]***

Notes

NetFilter est un parefeu 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 \geq 1.4.16). États 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*.

Notes

Connexions entrants sur un serveur Web hébergé par le parefeu 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
```

Notes

Constatation

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 négociés 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 négocié dynamiquement.
 - SIP canal de contrôle en TCP sur le port 5060, puis conversation en UDP sur des ports négociés dynamiquement.

Notes

- Le système de suivi de connexion supporte l'insertion d'une connexion dont on s'attend à ce qu'elle soit 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

Notes

Avantages

Ceci 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 [1].

[1] "Attaque contre les systèmes de suivi de connexions", Éric Leblond, SSTIC 2012, https://www.sstic.org/2012/presentation/utilisation_malveillante_des_suivis_de_connexions/.

Notes

Différentes technologies

- Filtrage simple (*packet filtering*)
 - Filtrage à état (*statefull inspection*)
 - Filtrage applicatif (*application firewalls*)
 - Passerelle proxy (*application-proxy gateways*)
 - Proxy dédié
 - Virtual Private Network
 - Network Access Control
 - Web Application Firewalls
 - Universal Threat Management

Différents usages

- Pare-feu individuel (*host-based firewall*) → serveurs
 - Pare-feu personnel → postes client
 - Pare-feu d'interconnexion

Notes

- Analyse protocolaire (*Deep Packet Inspection*)
 - Analyse de la couche applicative (cf IDS)
 - Profil de comportement de référence par protocole applicatif
 - Filtrage de protocoles sur port non standard (encapsulation)
 - Restrictions sur l'utilisation d'une application (exemple : FTP put)
 - Interdire certains types de contenus (pièce jointes exécutables, applet Java, etc.)
 - Déetecter les comportement suspicieux (répétition de commandes)
 - Vérification sommaire des entrées (taille des champs)
 - Vérification de la conformité protocolaire (RFC, automates)

Notes

- Pare-feu (L3 + L4) + proxy (L5)
 - Un agent proxy par type de protocole applicatif
 - Double connexion (client/proxy + proxy/serveur)
 - Proxy filtrant + authentification
 - Déchiffrement possible
 - + Niveau sécurité (capacité filtrage)
 - Consommation ressources
 - Nombre limité de protocoles supportés

Notes

- Généralement machine dédiée sans fonctionnalité de filtrage de paquet
 - Utilisation conjointe avec un pare-feu (redirection de trafic)
 - Alternative aux passerelles proxy
 - Spécifique à un type de flux (HTTP)
 - Filtrage des requêtes vers l'extérieur (outbound)
 - Filtrage depuis l'extérieur (inbound), reverse-proxy

Notes

- Fonctionnalité additionnelle de certains pare-feu
 - Pare-feu situé à la « frontière » du SI
 - Intégration permet le filtrage des flux en clair
 - Sécuriser les communications sur les réseaux non sûrs
 - SSL ou IPSEC
 - Deux types d'utilisation :
 - Connexion individuelle distante au SI (*host-to-gateway*)
 - Interconnexion multi-sites (*gateway-to-gateway*)
 - Contrôle d'accès + authentification (LDAP, RADIUS)
 - Consommation ressources (accélération matérielle)

Notes

- Contrôler l'accès des équipements (poste clients) aux ressources du SI
 - Authentification (802.1x) + vérification de « l'état de santé »
 - Vérifie que le poste respecte des règles de sécurité
 - Mises-à-jour de sécurité installées
 - Anti-malware installé et à jour
 - Configuration de sécurité (contrôle accès, etc.)
 - Présence ou absence de certains logiciels, etc.
 - Accès plus ou moins limité selon les vérifications effectuées
 - Nécessite l'installation d'un agent qui puisse dialoguer avec le NAC
 - Utile pour gérer le BYOD

Notes

Web Application Firewall

- Pare-feu applicatif dédié aux flux Web
 - Fonctionnalités proches d'un IPS dédié au Web (signature spécifiques + détection d'anomalies)
 - Placement en frontal du serveur Web

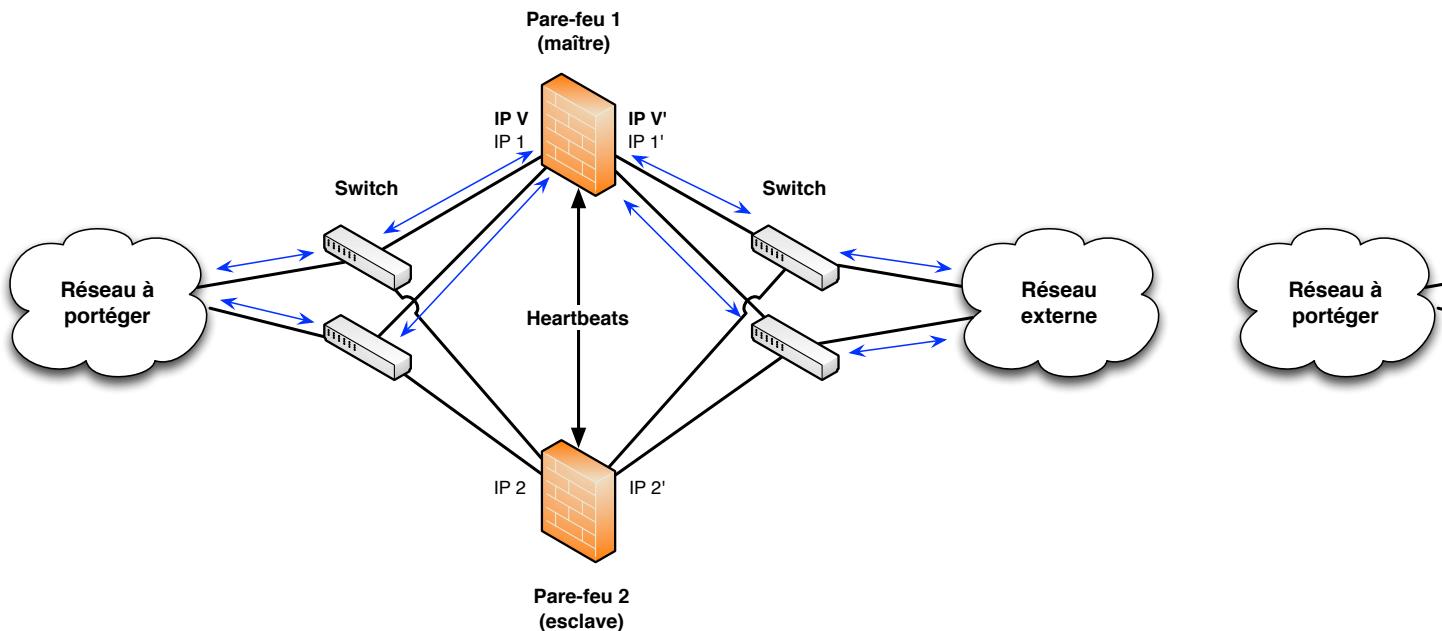
Unified Threat Management

- Solution « toute-en-un »
 - Intégration de différentes fonctions de sécurité : pare-feu, VPN, anti-malware, IPS, etc.
 - Vise typiquement les PME
 - + Coût, déploiement et administration simplifiée
 - Ressources partagées, fonctionnalités imposées

Notes

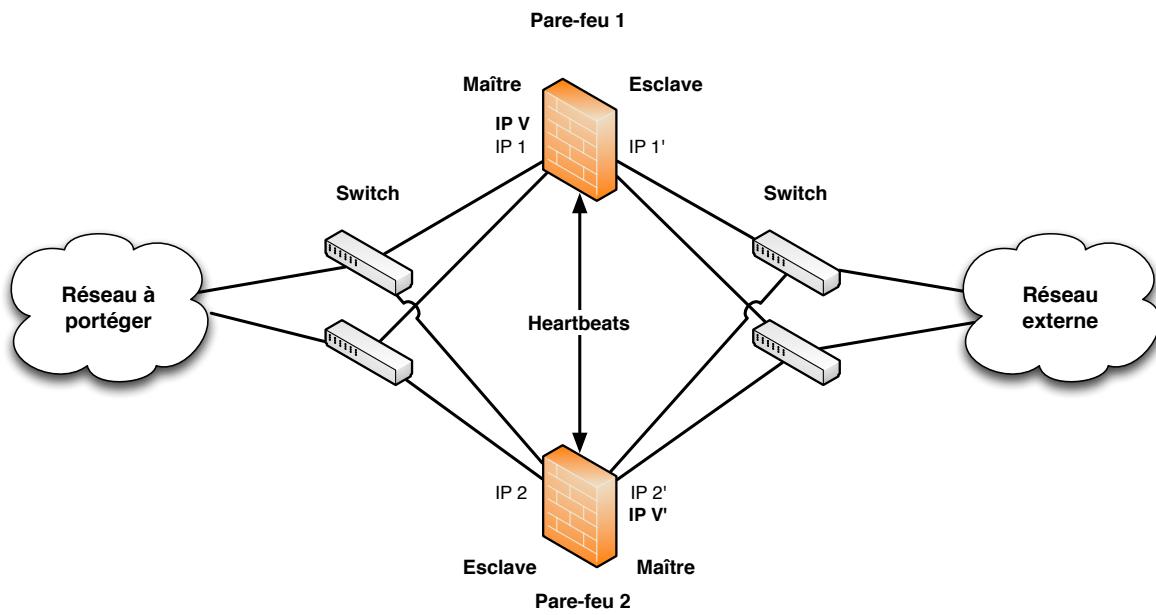
- Le pare-feu est un point de passage obligatoire critique (SPOF)
 - Tolérer les pannes → multiplier les pare-feux (cluster) pour chaque noeud de filtrage
 - Différents modes de réPLICATION :
 - Actif/passif (maître/esclave)
 - Actif/actif (permet l'équilibrage de charge)
 - Difficulté pour le filtrage à état (*statefull*) : il faut synchroniser l'état du suivi de connexion (sauvegardé en mémoire)
 - Suppose aussi redondance des équipements réseaux (802.3ad, LACP)
 - Augmente la complexité (synchronisation des paramétrages et des journaux)

Notes



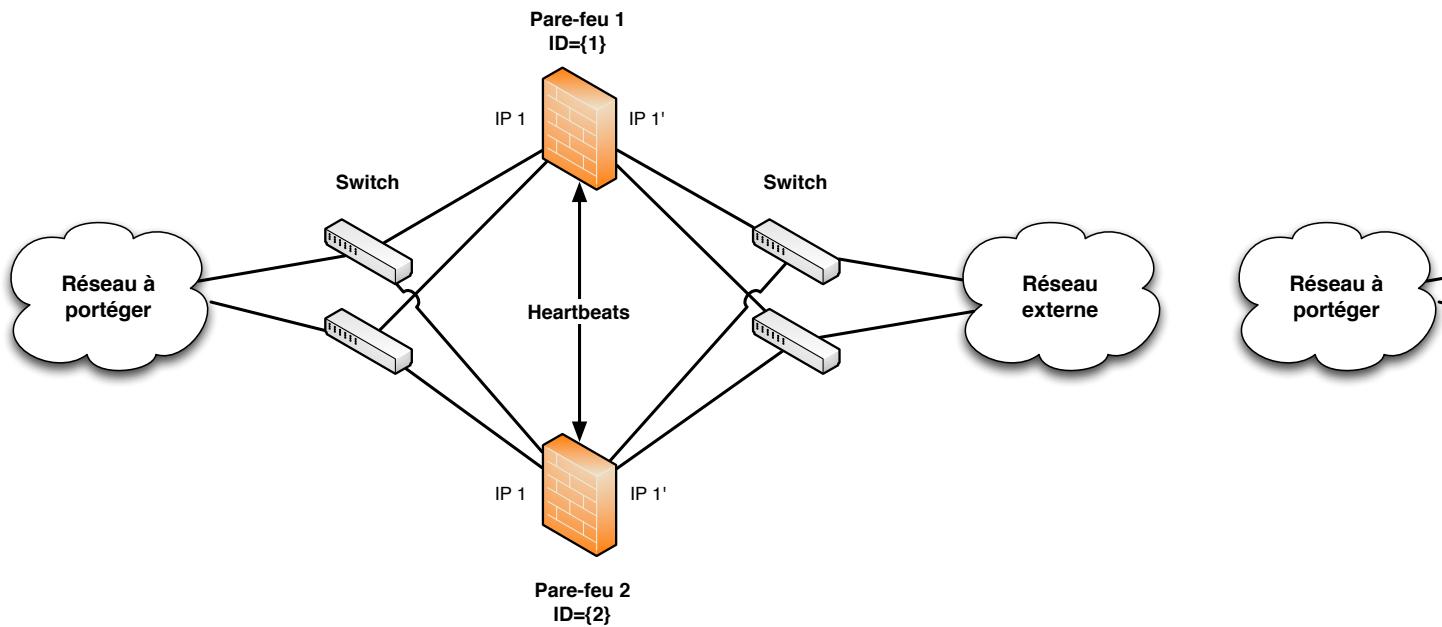
- Protocole de réPLICATION : CARP, VRRP, HSRP
 - Protocole de synchronisation : pfsync

Notes



- Equipement actif fonction du sens de la connexion
 - RéPLICATION synchrone nécessaire mais inenvisageable

Notes



- Filtrage des paquets si $h(\text{IP_SRC}) \bmod n = \text{ID}$
- Synchronisation du suivi de connexion

Pare-feu d'interconnexion

- Machines dédiées (PC avec logiciel ou *appliance*)
 - Filtrage du trafic inter-zones (LAN, DMZ, WAN, etc.)
 - Défense périmétrique

Pare-feu individuels

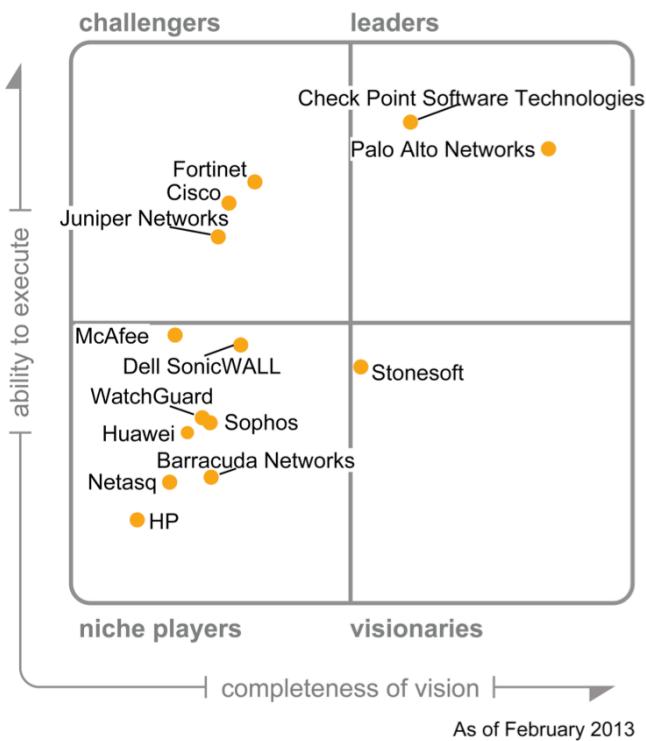
- Filtrage des flux entrant/sortant de chaque machine (serveur, poste client)
 - Protection de la machine + confinement
 - Fonctionnalité intégrée à l'OS (cf. Netfilter/IPTABLES)
 - Indispensable pour protéger les machines qui ne sont pas situées dans une zone protégée (par exemple poste client en itinérance)
 - + Défense en profondeur, granularité du filtrage
 - Désactivation possible en cas de compromission de la machine si l'attaquant obtient les priviléges nécessaires (administrateurs)

Notes

- Pare-feu individuel pour poste client
 - Interface administration simplifiée
 - Filtrage des flux par application
 - Nécessite l'utilisation d'un agent en espace utilisateur
 - Limite la propagation des malware
 - Profil de filtrage en fonction du type connexion/localisation

Notes

Figure 1. Magic Quadrant for Enterprise Network Firewalls



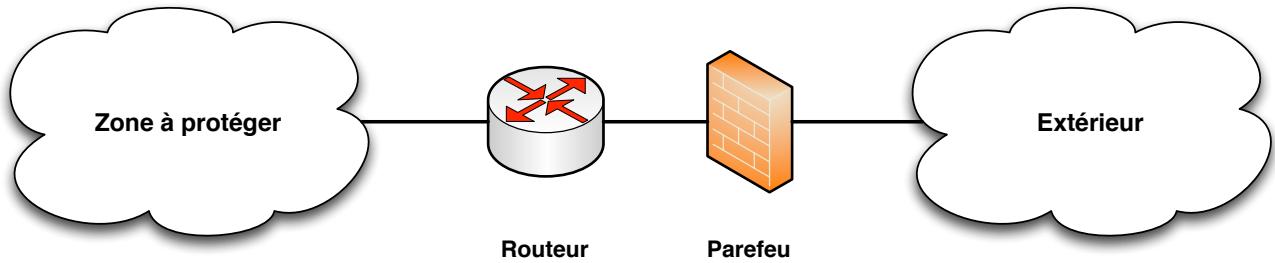
Source: Gartner (February 2013)

Notes

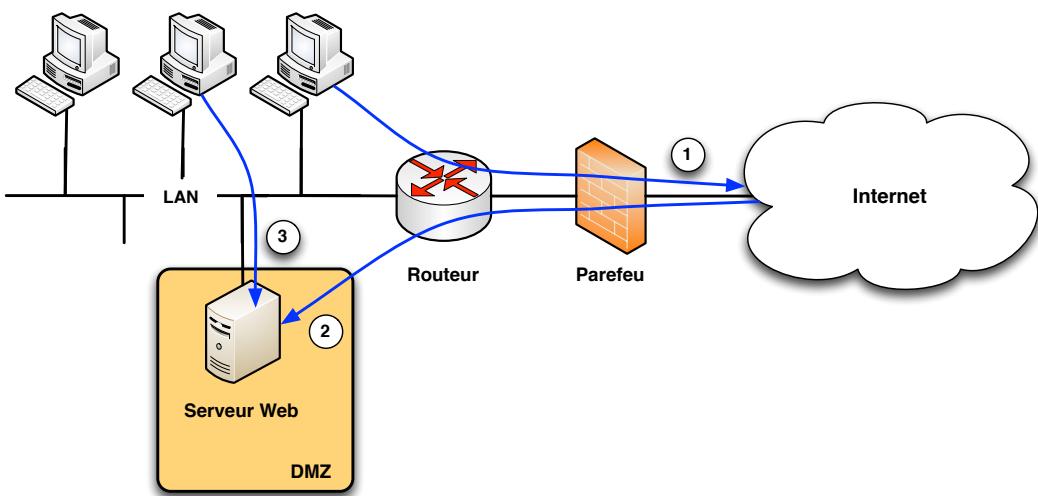
Figure 1. Magic Quadrant for Unified Threat Management



Source: Gartner (March 2012)



- Protection d'une zone (machine) vis-à-vis de l'extérieur
- Filtrage des flux entrant
 - Protection de la zone contre les attaques externes
- Filtrage des flux sortant
 - Empêcher fuite d'information
 - Limiter usage des utilisateurs
 - Défense en profondeur (propagation malware, porte dérobée)
- Placé en amont ou intégré au routeur

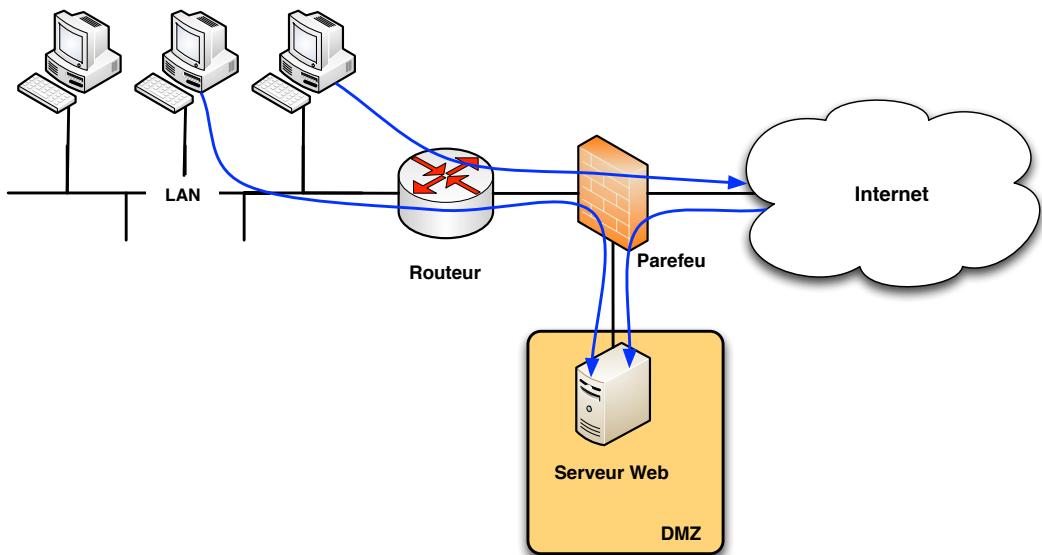


- 3 types de flux :
 - ① Consultation site externe depuis LAN
 - ② Consultation site web depuis Internet
 - ③ Mise à jour site web depuis LAN

Notes

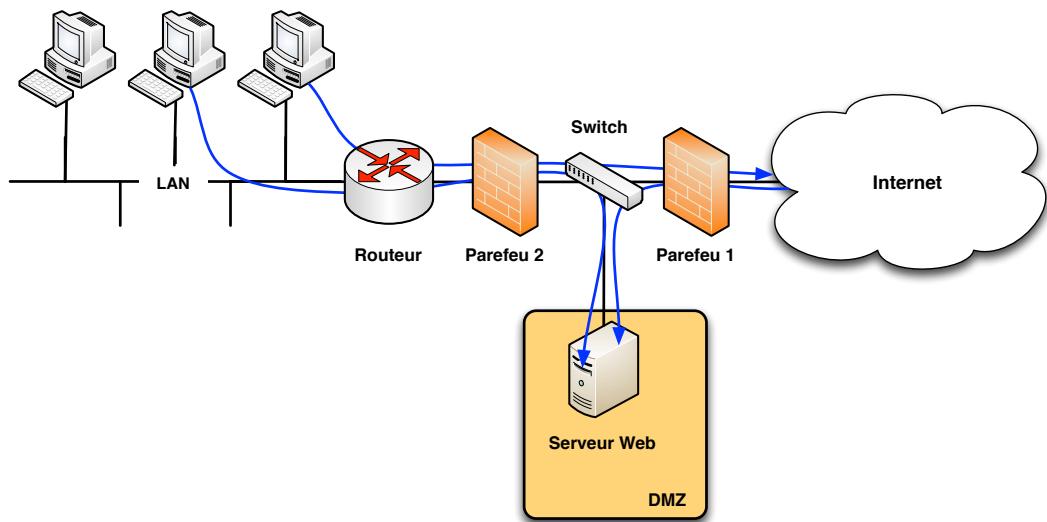
- Pas de cloisonnement (rebond vers LAN)
 - Sécurité repose sur un seul composant
 - Pas de filtrage des flux applicatifs

Notes



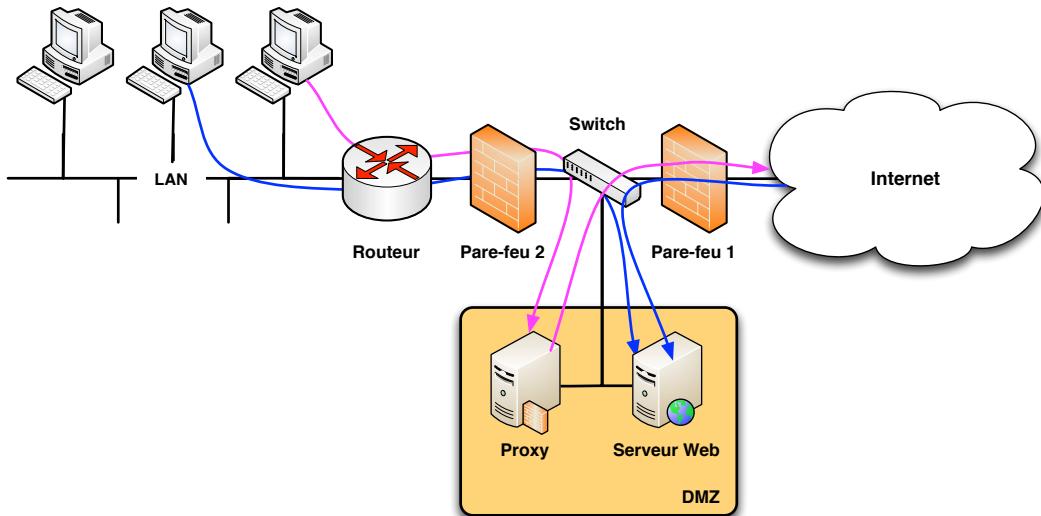
- DMZ cloisonnée

Notes



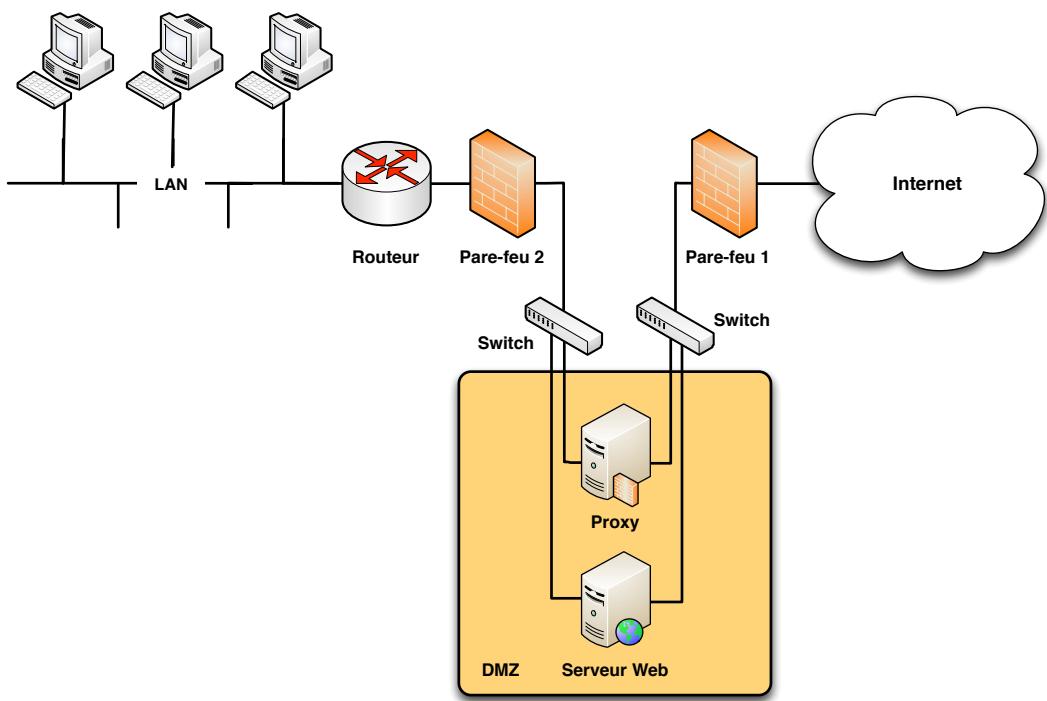
- DMZ cloisonnée + diversification des pare-feux

Notes



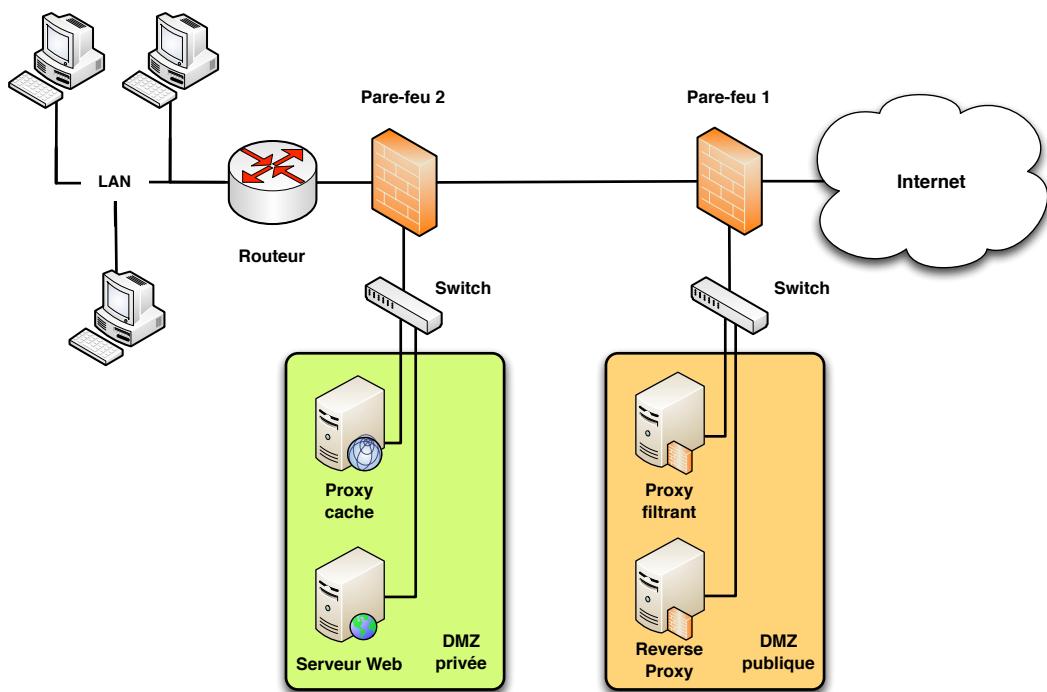
- DMZ cloisonnée + diversification des pare-feux + filtrage applicatif

Notes



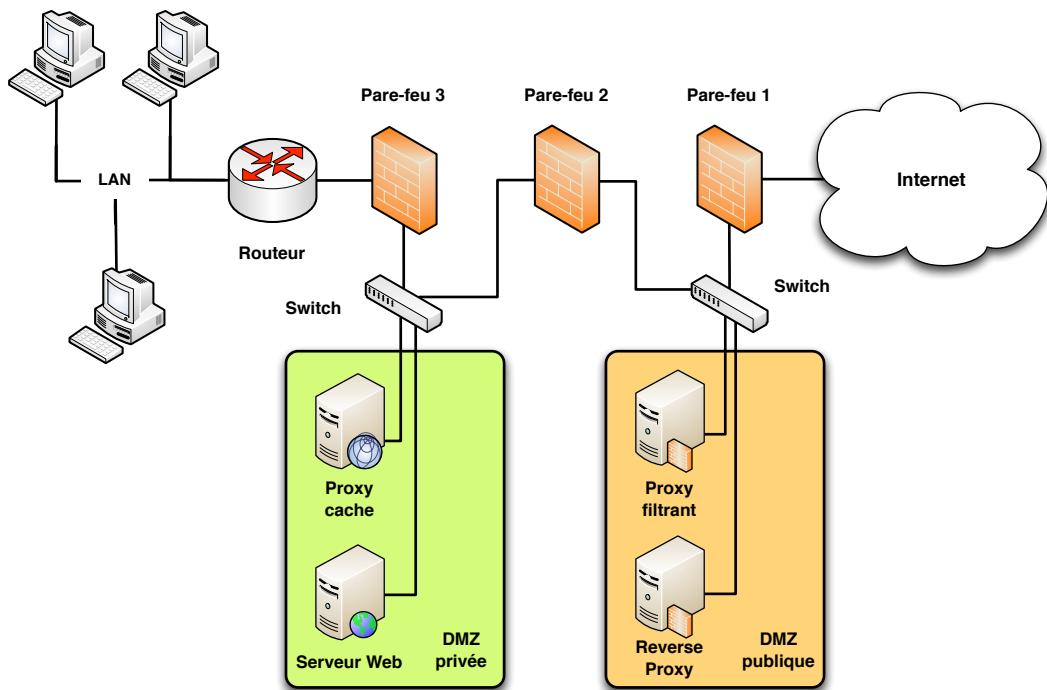
- DMZ cloisonnée + diversification des pare-feux + filtrage applicatif

Notes



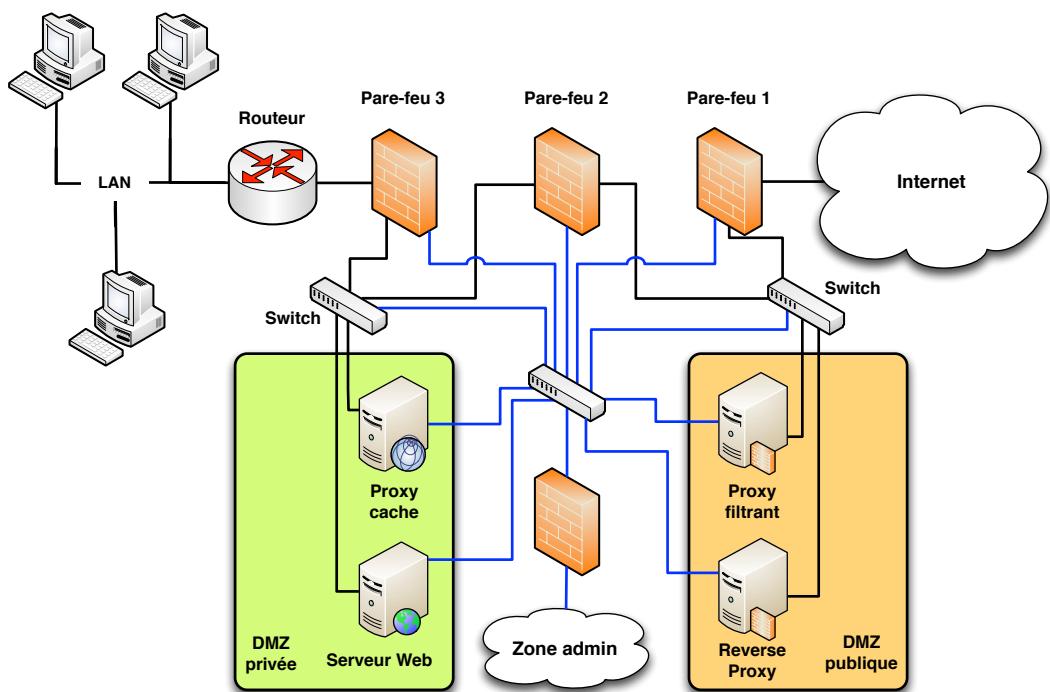
- DMZ publique (services) + DMZ privée (fonctions de sécurité)

Notes



- Cloisonnement vertical possible (par type de flux)

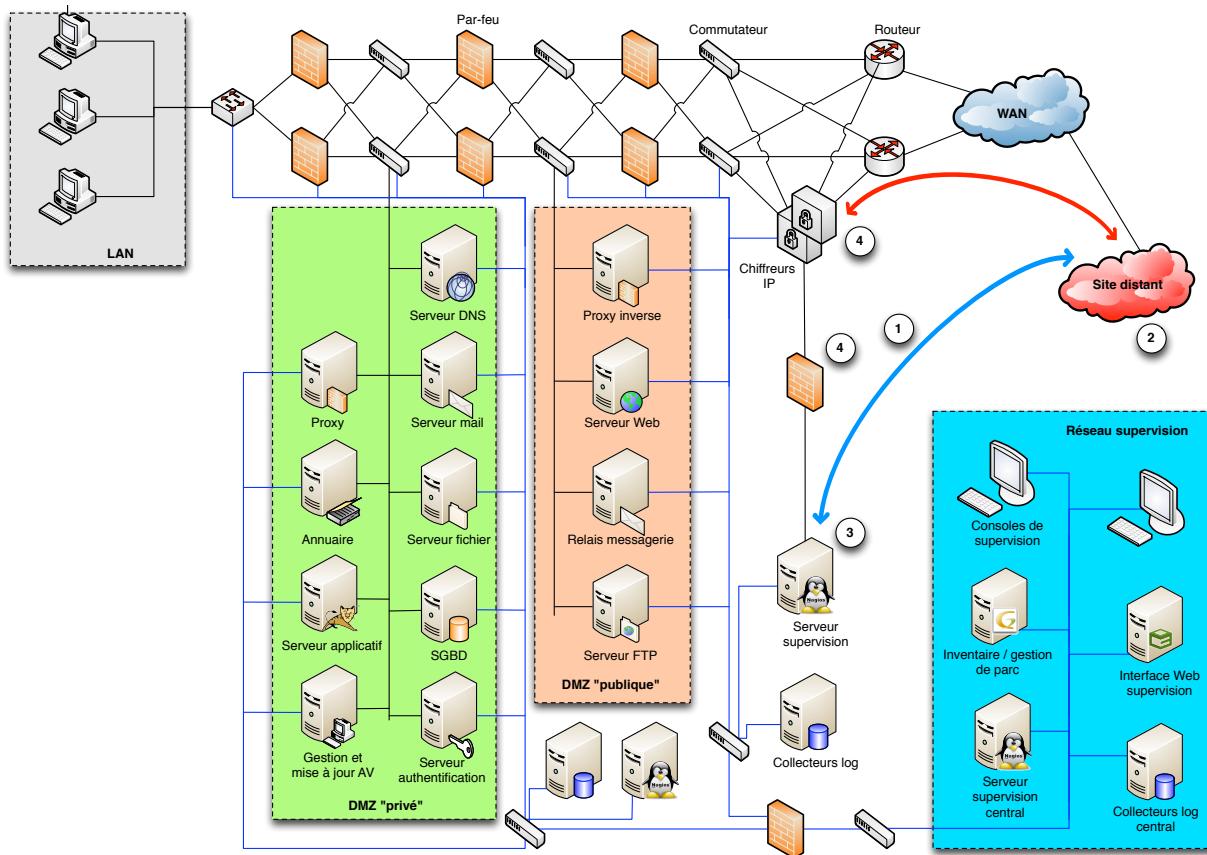
Notes



- Interfaces et réseau dédiés

Notes

Exemple d'architecture



- Pare-feu = protection périphérique
 - Différentes solutions possibles → choix fonction besoins (sécurité)
 - Cloisonnement horizontal des services selon niveau confiance
 - Cloisonnement vertical selon types de flux
 - Ne pas utiliser le NAT pour cloisonner
 - perturbe le fonctionnement de certains protocoles
 - problème de traçabilité
 - Multiplication des pare-feux sur le chemin
 - Ne pas faire de supposition sur les politiques
 - + Défense en profondeur
 - Coût, complexité (administration)
 - Mise au point en cas de problème
 - Diversification vs. maintenabilité
 - Réseau dédié pour l'administration/supervision

- Etape préliminaire : analyse de risque
 - Quels sont les flux nécessaires ?
 - Quels sont les risques (vulnérabilités) ?
 - Quels sont les mécanismes de sécurité utilisés ?
 - Interdire par défaut
 - Filtrer selon :
 - Caractéristiques IP (adresse source/destination)
 - Protocoles
 - Protocoles applicatifs (port)
 - Usage
 - Identité

Notes

- Toujours interdire adresses invalides (*martian packets*) : 127.0.0.0/8, 240.0.0.0/4, etc.
 - Associer plages adresses (réseau) à chaque zone (interface)
 - Limiter usurpation d'adresse (*spoofing*)
 - Interdire flux sortants avec adresse source n'appartenant pas au réseau de la zone
 - Interdire flux entrants avec adresse destination n'appartenant pas au réseau de la zone
 - Les adresses privées (ex : 192.168.0.0/16) ne doivent pas "sortir" de la zone (NAT)
 - Filtrer les paquets utilisant le *source routing*
 - Filtrer paquets invalides (entête incorrect, hors séquence, etc.)
 - Filtrer broadcast et multicast (suivant usage)
 - Filtrage au moins à la périphérie (*bogon filtering*)
 - Implémenter la matrice de flux (dépend des besoins)
 - Nombre de règles : sécurité vs. complexité / performances

TCP et UDP

- Filtrage des ports destination
 - Limites : utilisation port non standard, *tunneling*
 - Utilisation du filtrage à état pour autoriser les réponses

ICMP

- Protocole utile (diagnostique, signalisation) mais utilisé par les attaquants (découverte, manipulation des flux réseau)
 - Filtrage à la périphérie seulement
 - Filtrer selon type de message
 - Autoriser type 3 (destination unreachable)
 - Filtrer éventuellement le ping (type 8) en entrée
 - Adapter le filtrage pour IPV6 : cf RFC4890

Notes

- Utilisation d'un proxy dédié ou d'une passerelle proxy
 - Restreindre usages ou contenus qui représentent un risque
 - Filtrer en amont les flux pour alléger la charge du proxy
 - Détection encapsulation (*tunneling*)
 - Détection d'anomalies, log
 - Filtrage en entrée (*reverse-proxy*)
 - Protéger les serveurs (pas d'accès direct)
 - Alléger la charge du serveur + diversification
 - Limites : performances + fonctionnalités
 - Filtrage en sortie
 - Limiter usages des utilisateurs (filtrage requêtes)
 - Protéger utilisateurs (filtrage réponses)

Notes

Identité

- Notion d'identité absente des couches basses (L3-L4)
 - Utilisation d'un VPN avec authentification
 - Identité fournie par couche applicative
 - Utilisation d'un agent sur poste client (exemple NuFw/UFWI)
 - Identité associée à une IP ou une connexion
 - Utilisation d'un NAC

Aspects temporels

- Filtrer les connexions inactives
 - Filtrer, rediriger, réguler (QoS) si le débit est trop important (DoS)
 - Supposons de fixer des seuils

Notes

1 Planification

- Recueil du besoin, analyse de risque
 - Architecture
 - Choix du type de solution
 - Planification de l'administration (définition des rôles)
 - Spécification de la politique (matrice de flux)

② Configuration

- Implémentation de la politique
 - Configuration des fonctions annexes (administration, journalisation, etc.)
 - Documentation

③ Mise au point

- Idéalement, déploiement dans un environnement de test
 - A défaut, déploiement en mode dégradé : surveillance (log)
 - Vérification de la conformité : interopérabilité, niveau de sécurité, performances, etc.

4 Déploiement

5 Administration et supervision

Notes

- Niveau fonctionnel
 - Quel type de zones/hôtes à protéger ?
 - Quelles fonctionnalités de filtrage ?
 - Quelles fonctionnalités additionnelles (VPN, IPS, etc.) ?
 - Robustesse
 - Evaluation de sécurité, certification (CSPN, CC)
 - Analyse des vulnérabilités publiques (CVE)
 - Performances
 - Bande-passante, latency, # connexions en parallèle, # connexions/s
 - Haute-disponibilité (actif/actif, actif/passif)
 - Intégration
 - Compatibilité logicielle et matérielle
 - Intégration avec d'autres mécanismes de sécurité (VPN, proxy, anti-malware, etc.)
 - Contrainte liées à l'architecture

Notes

- Administration
 - Type d'interfaces : locale vs. à distance, protocoles (SSH, HTTPS, etc.)
 - Interface physique dédiée, VLAN
 - Administration centralisée (outil propriétaire, compatibilité avec des outils tierces)
 - Gestion rôles, contrôle d'accès
 - Supervision
 - Protocoles (SSH, HTTPS, SNMP, SysLog, etc.)
 - Interface physique dédiée, VLAN
 - Supervision centralisée (outil/format propriétaire, compatibilité avec des outils/formats tierces)
 - Contraintes physiques
 - Besoins en alimentation (redondance), climatisation
 - Encombrement, poids
 - Contrôle d'accès physique
 - Evolutivité : évolution des besoins en bande-passante, nouveaux usages (protocoles), nouvelles menaces

Notes

- Installer et configurer le matériel
 - Configurer le système et les dépendances (pare-feu logiciel)
 - Protéger le pare-feu (restreindre l'accès d'administration)
 - Mettre à jour, configurer les mises à jour
 - Implémenter la politique de filtrage
 - Ordre des règles (ordre d'évaluation, performances)
 - Commenter les règles
 - Structurer les règles (utilisation de « fonctions »)
 - Sauvegarder et gérer les règles (gestionnaire de version)
 - Mutualiser les règles si nécessaire (référentiel)
 - Configurer la journalisation/supervision
 - Journalisation locale et centralisée
 - Supervision fonctionnelle (panne, dysfonctionnement) vs. supervision de sécurité (accès refusé, détection comportement suspect)
 - Traçabilité vs. performance/volumétrie
 - Configurer les autres paramètres (administration)

Notes

- Phase la plus longue du cycle de vie
 - Mise à jour du produit
 - Mise à jour de la politique
 - Nouveaux usages (applications, protocoles)
 - Evolution de l'architecture (machines, réseaux, etc.)
 - Nouvelles menaces
 - Problèmes d'interopérabilité non détecté au préalable
 - Audit global régulier de la politique (harmonisation, suppression des règles obsolètes, simplification)
 - Suivi et vérification des performances
 - Identification des attaques (lien avec la supervision de sécurité)
 - Gestion des sauvegardes
 - Evaluation de la robustesse (tests d'intrusions)

Notes

-  Pablo Neira Ayuso, Rafael M. Gasca, and Laurent Lefèvre, *Demystifying cluster-based fault-tolerant firewalls*, IEEE Internet Computing **13** (2009), no. 6, 31–38.
 -  _____, *Ft-fw : A cluster-based fault-tolerant architecture for stateful firewalls*, Computers & Security **31** (2012), no. 4, 524–539.
 -  ANSSI, *Guide de définition d'une architecture de passerelle d'interconnexion sécurisée*, Tech. Report 3248, SGDSN/ANSSI, December 2011.
 -  Karen A. Scarfone and Paul Hoffman, *Guidelines on firewalls and firewall policy*, Tech. Report SP 800-41 Rev. 1., Gaithersburg, MD, United States, 2009.

Notes

Boeing 777-300R de la Singapore AirLines.



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

Notes