**Pare-feu (informatique)**

[](http://commons.wikimedia.org/wiki/File:Firewall_(networking).png)

Un pare-feu, représenté par un mur de briques, pour cloisonner le réseau privé

Un **pare-feu**1, ou **firewall** (en anglais), est un logiciel et/ou un matériel, permettant de faire respecter la politique de sécurité du réseau, celle-ci définissant quels sont les types de communication autorisés sur ce réseau informatique. Il mesure la prévention des applications et des paquets.

**Terminologie**

Un pare-feu est parfois appelé coupe-feu, garde-barrière, barrière de sécurité, ou encore firewall.

Dans un environnement Unix BSD (Berkeley Software Distribution), un pare-feu est aussi appelé *packet filter*.

**Origine du terme**

Le terme peut avoir plusieurs origines :

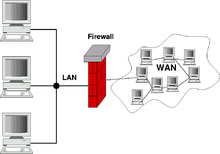
* au théâtre le « *pare-feu* » ou « *coupe-feu* » est un mécanisme qui permet, une fois déclenché, d'éviter au feu de se propager de la salle vers la scène.
* dans le domaine de la lutte contre les incendies, le mot fait référence aux allées pare-feux sont destinées à bloquer les incendies de forêt, et dans le domaine de l'architecture il fait référence aux portes coupe-feux.



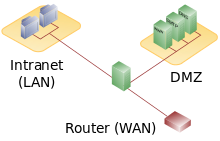
Autocollant « Porte coupe-feu »

En informatique l'usage du terme « pare-feu » est donc métaphorique : une porte empêchant les flammes de l'Internet d'entrer chez soi et/ou de « *contaminer* » un réseau informatique.

**Fonctionnement général**

[](http://commons.wikimedia.org/wiki/File:Gateway_firewall.png)

Pare-feu passerelle entre LAN et WAN

[](http://commons.wikimedia.org/wiki/File:DMZ_network_diagram_1_firewall.svg)

Pare-feu routeur, avec une zone DMZ

Le pare-feu était jusqu'à ces dernières années considéré comme une des pierres angulaires de la sécurité d'un réseau informatique (il perd en importance au fur et à mesure que les communications basculent vers le HTTP sur SSL, court-circuitant tout filtrage). Il permet d'appliquer une politique d'accès aux ressources réseau (serveurs).

Il a pour principale tâche de contrôler le trafic entre différentes zones de confiance, en filtrant les flux de données qui y transitent. Généralement, les zones de confiance incluent Internet (une zone dont la confiance est nulle) et au moins un réseau interne (une zone dont la confiance est plus importante).

Le but est de fournir une connectivité contrôlée et maîtrisée entre des zones de différents niveaux de confiance, grâce à l'application de la politique de sécurité et d'un modèle de connexion basé sur le principe du moindre privilège.

Le filtrage se fait selon divers critères. Les plus courants sont :

* l'origine ou la destination des paquets (adresse IP, ports TCP ou UDP, interface réseau, etc.) ;
* les options contenues dans les données (fragmentation, validité, etc.) ;
* les données elles-mêmes (taille, correspondance à un motif, etc.) ;
* les utilisateurs pour les plus récents.

Un pare-feu fait souvent office de routeur et permet ainsi d'isoler le réseau en plusieurs zones de sécurité appelées zones démilitarisées ou DMZ. Ces zones sont séparées suivant le niveau de confiance qu'on leur porte.

Enfin, le pare-feu est également souvent extrémité de tunnel IPsec ou SSL. L'intégration du filtrage de flux et de la gestion du tunnel est en effet nécessaire pour pouvoir à la fois protéger le trafic en confidentialité et intégrité et filtrer ce qui passe dans le tunnel. C'est le cas notamment de plusieurs produits du commerce nommés dans la liste ci-dessous.

**Catégories de pare-feu**

Les pare-feu sont un des plus vieux équipements de sécurité informatique et, en tant que tels, ils ont été soumis à de nombreuses évolutions. Suivant la génération du pare-feu ou son rôle précis, on peut les classer en différentes catégories.

**Pare-feu sans état (*stateless firewall*)**

C'est le plus vieux dispositif de filtrage réseau, introduit sur les routeurs. Il regarde chaque paquet indépendamment des autres et le compare à une liste de règles préconfigurées.  
Ces règles peuvent avoir des noms très différents en fonction du pare-feu :

* « ACL » pour *Access Control List* (certains pare-feu Cisco),
* politique ou *policy* (pare-feu Juniper/Netscreen),
* filtres,
* règles ou *rules*,
* etc.

La configuration de ces dispositifs est souvent complexe et l'absence de prise en compte des machines à états des protocoles réseaux ne permet pas d'obtenir une finesse du filtrage très évoluée. Ces pare-feu ont donc tendance à tomber en désuétude mais restent présents sur certains routeurs ou systèmes d'exploitation.

**Pare-feu à états (*stateful firewall*)**

Certains protocoles dits « à états » comme TCP introduisent une notion de connexion. Les pare-feu à états vérifient la conformité des paquets à une connexion en cours. C’est-à-dire qu'ils vérifient que chaque paquet d'une connexion est bien la suite du précédent paquet et la réponse à un paquet dans l'autre sens. Ils savent aussi filtrer intelligemment les paquets ICMP qui servent à la signalisation des flux IP.

Enfin, si les ACL autorisent un paquet UDP caractérisé par un quadruplet (ip\_src, port\_src, ip\_dst, port\_dst) à passer, un tel pare-feu autorisera la réponse caractérisée par un quadruplet inversé, sans avoir à écrire une ACL inverse. Ceci est fondamental pour le bon fonctionnement de tous les protocoles fondés sur l'UDP, comme DNS par exemple. Ce mécanisme apporte en fiabilité puisqu'il est plus sélectif quant à la nature du trafic autorisé. Cependant dans le cas d'UDP, cette caractéristique peut être utilisée pour établir des connexions directes (P2P) entre deux machines (comme le fait Skype par exemple).

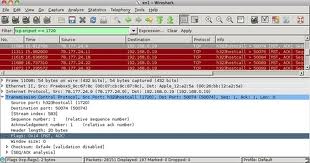
**Pare-feu applicatif**

Dernière mouture de pare-feu, ils vérifient la complète conformité du paquet à un protocole attendu. Par exemple, ce type de pare-feu permet de vérifier que seul du protocole HTTP passe par le port TCP 80. Ce traitement est très gourmand en temps de calcul dès que le débit devient très important. Il est justifié par le fait que de plus en plus de protocoles réseaux utilisent un tunnel TCP afin de contourner le filtrage par ports.

Une autre raison de l'inspection applicative est l'ouverture de ports dynamique. Certains protocoles comme le fameux FTP en mode passif échangent entre le client et le serveur des adresses IP ou des ports TCP/UDP. Ces protocoles sont dits « **à contenu sale** » ou « passant difficilement les pare-feu » car ils échangent au niveau applicatif (FTP) des informations du niveau IP (échange d'adresses) ou du niveau TCP (échange de ports). Ce qui transgresse le principe de la séparation des couches réseaux. Pour cette raison, les protocoles « à contenu sale » passent difficilement voire pas du tout les règles de NAT ...dynamiques, à moins qu'une inspection applicative ne soit faite sur ce protocole.

Chaque type de pare-feu sait inspecter un nombre limité d'applications. Chaque application est gérée par un module différent pour pouvoir les activer ou les désactiver. La terminologie pour le concept de module est différente pour chaque type de pare-feu : par exemple : Le protocole HTTP permet d'accéder en lecture sur un serveur par une commande GET, et en écriture par une commande PUT. Un pare-feu applicatif va être en mesure d'analyser une connexion HTTP et de n'autoriser les commandes PUT qu'à un nombre restreint de machines.

* **Pare-feu applicatif** sur DenyAll
* **Conntrack** (suivi de connexion) et **l7 Filter** (filtrage applicatif) sur Linux Netfilter
* **CBAC** sur Cisco IOS
* **Fixup** puis **inspect** sur Cisco PIX
* **ApplicationLayerGateway** sur **Proventia M**,
* **Predefined Services** sur Juniper ScreenOS
* **Stateful Inspection** sur Check Point FireWall-1
* **Deep Packet Inspection** sur Qosmos
* **Web Application Firewall** sur BinarySEC



Analyse de trafic d’un pare-feu avec Wireshark

**Pare-feu identifiant**

Un pare-feu réalise l’identification des connexions passant à travers le filtre IP. L'administrateur peut ainsi définir les règles de filtrage par utilisateur et non plus par adresse IP ou adresse MAC, et ainsi suivre l'activité réseau par utilisateur.

Plusieurs méthodes différentes existent qui reposent sur des associations entre IP et utilisateurs réalisées par des moyens variés. On peut par exemple citer authpf (sous OpenBSD) qui utilise ssh pour faire l'association. Une autre méthode est l'identification connexion par connexion (sans avoir cette association IP=utilisateur et donc sans compromis sur la sécurité), réalisée par exemple par la suite NuFW, qui permet d'identifier également sur des machines multi-utilisateurs.  
On pourra également citer Cyberoam qui fournit un pare-feu entièrement basé sur l'identité (en réalité en réalisant des associations adresse MAC = utilisateur) ou Check Point avec l'option NAC Blade qui permet de créer des règles dynamiques basée sur l'authentification Kerberos d'un utilisateur, l'identité de son poste ainsi que son niveau de sécurité (présence d'antivirus, de patchs particuliers).



Logo de Check Point

**Pare-feu personnel**

Les pare-feu personnels, généralement installés sur une machine de travail, agissent comme un pare-feu à états. Bien souvent, ils vérifient aussi quel programme est à l'origine des données. Le but est de lutter contre les virus informatiques et les logiciels espions.

**Portail captif**

Les portails captifs sont des pare-feux dont le but est d'intercepter les usagers d'un réseau de consultation afin de leur présenter une page web spéciale (par exemple : avertissement, charte d'utilisation, demande d'authentification, etc.) avant de les laisser accéder à Internet. Ils sont utilisés pour assurer la traçabilité des connexions et/ou limiter l'utilisation abusive des moyens d'accès. On les déploie essentiellement dans le cadre de réseaux de consultation Internet mutualisés filaires ou Wi-Fi.

**Technologies utilisées**

Les pare-feu récents embarquent de plus en plus de fonctionnalités, parmi lesquelles on peut citer :

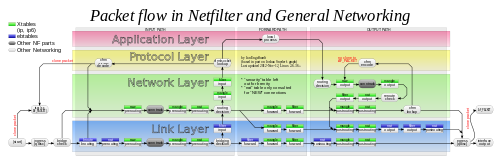
* Filtrage sur adresses IP / protocole,
* Inspection *stateful*2 et applicative,
* Intelligence artificielle pour détecter le trafic anormal,
* Filtrage applicatif :
  + HTTP (restriction des URL accessibles),
  + Courriel (Anti-pourriel),
  + Logiciel antivirus, anti-logiciel malveillant
* Traduction d'adresse réseau,
* Tunnels IPsec, PPTP, L2TP,
* Identification des connexions,
* Serveurs de protocoles de connexion (telnet, SSH), de protocoles de transfert de fichier (SCP),
* Clients de protocoles de transfert de fichier (TFTP),
* Serveur Web pour offrir une interface de configuration agréable,
* Serveur mandataire (« *proxy* » en anglais),
* Système de détection d'intrusion (« IDS » en anglais)
* Système de prévention d'intrusion (« IPS » en anglais)

**Etudions Netfilter**

**Netfilter** est un framework implémentant un pare-feu au sein du noyau Linux à partir de la version 2.4 de ce dernier. Il prévoit des accroches (*hooks*) dans le noyau pour l'interception et la manipulation des paquets réseau lors des appels des routines de réception ou d'émission des paquets des interfaces réseau.

La version 1.4.2 a reçu un Certificat de Sécurité de Premier Niveau (CSPN) par l'Agence nationale de la sécurité des systèmes d'information1.

## Mode de fonctionnement

[](http://commons.wikimedia.org/wiki/File:Netfilter-packet-flow.svg)

Flow of network packets through Netfilter

### Programmes utilitaire

#### iptables

Les modules du noyau nommés *ip\_tables*, *ip6\_tables*, *arp\_tables* (les soulignements font partie du nom) et *ebtables* sont les systèmes des accroches de Netfilter. Ils fournissent un système basé sur des tableaux pour définir des règles de pare-feu qui filtrent les paquets ou les transforment. Les tableaux peuvent être administrés par les outils utilisateur **iptables**, **ip6tables**, **arptables** et **ebtables**, respectivement.

Chaque tableau est en fait sa propre accroche, et chaque tableau a été créé pour servir un but précis. En ce qui concerne Netfilter, généralement l'exécution de ces tableaux dans un ordre précis par rapport aux autres tableaux. Toutefois, tous les tableaux vont exécuter la même fonction de traitement de tableau pour parcourir, et exécuter des règles.

Les chaînes, à cet égard, correspondent à l'endroit où la pile de Netfilter a été invoquée, comme la réception de paquets (PREROUTING), rendu sur place (INPUT), transmise (FORWARD), généré localement (OUTPUT) et envoyer/envoyant des paquets (POSTROUTING). Les modules de Netfilter qui ne prévoient pas de tableaux (voir ci-dessous) peuvent également vérifier l'origine des paquets pour choisir leur mode de fonctionnement.

* le module *iptable\_raw*, lorsqu'il est chargé, peut enregistrer une accroche qui sera appelée avant toutes les autres accroches. Il fournit un tableau appelé *raw* que l'on peut utiliser pour filtrer des paquets avant qu'ils n'atteignent les opérations nécessitant plus de mémoire, comme Connection Tracking.
* le module *iptable\_mangle* enregistre une accroche et le tableau *mangle*, qui est consulté après Connection Tracking (mais toujours avant les autres tableaux), pour apporter des modifications au paquet, qui peuvent influencer d'autres règles, telles que le NAT ou le filtrage.
* le module *iptable\_nat* enregistre deux accroches : Les transformations de DNAT sont appliquées avant l'accroche de filtrage ; les transformations de SNAT sont appliquées ensuite. Le tableau *nat* qui est mis à la disposition de iptables est simplement une “base de données de configuration” pour les mappages NAT, et n'est destiné à aucun filtrage d'aucune sorte.
* enfin, le module *iptable\_filter* enregistre le tableau *filter* utilisé pour le filtrage général (firewall).

### Connection Tracking

Une des caractéristiques importantes construites sur le framework Netfilter est Connection Tracking. CT permet au noyau de garder la trace de toutes les connexions réseau logiques ou de sessions, et, par conséquent, porte tous les paquets qui composent cette connexion. NAT s'appuie sur cette information pour traduire tous les paquets de la même manière, et iptables peut utiliser cette information pour agir comme un pare-feu "stateful".

L'état de connexion est cependant complètement indépendant de tout état haut niveau, tel que l'état TCP ou SCTP. Une partie de la raison en est que, lorsque les paquets ne font que transiter (pas de livraison locale), le moteur de TCP ne doit pas nécessairement être invoqué. Même les modes sans-connexion tels que les transmissions UDP, IPsec (AH/ESP), le GRE et autres protocoles de tunneling ont au moins un pseudo-état de connexion. L'heuristique de ces protocoles est souvent basée sur une valeur de délai de l'inactivité préréglée, après expiration de laquelle une connexion Netfilter est abandonnée.

Chaque connexion Netfilter est identifiée de façon unique par un tuple (protocole de couche 3, adresse source, adresse de destination, protocole de couche 4, clé de couche 4). La clé de couche 4 dépend du protocole de transport : pour les protocoles TCP/UDP c'est le numéro de port; pour des tunnels, c'est leur tunnel ID. Dans le cas contraire, la clé prend la valeur zéro comme si elle ne faisait pas partie du tuple. Pour être capable d'inspecter le port TCP, les paquets seront obligatoirement défragmentés.

Les connexions Netfilter peuvent être manipulées avec l'outil **conntrack**.

Iptables peut inspecter les informations de connexion tels que les états, les statuts etc. pour rendre les règles de filtrage de paquets plus puissantes et plus faciles à gérer. Le plus souvent, les états sont les suivants :

* “NEW” (nouveau) : le paquet essaie de créer une nouvelle connexion
* "ESTABLISHED" (établi) : le paquet fait partie d'une connexion déjà existante
* "RELATED" (liée) : cet état est attribué à un paquet qui est l'ouverture d'une nouvelle connexion, et qui a été "attendu". Les mini-ALGs susmentionnés mettent ces attentes en place, par exemple, lorsque le module *nf\_conntrack\_ftp* voit une commande FTP "PASV".
* "INVALID" (invalide) : le paquet a été jugé invalide. La cause la plus fréquente est qu'il ne respecterait pas le diagramme d'état du protocole TCP.
* "UNTRACKED" (non-suivant) : état spécial qui peut être attribué par l'administrateur pour contourner Connection Tracking (voir tableau *raw* ci-dessus)

En pratique, le premier paquet que le sous-système conntrack perçoit est donc classé "nouveau". Le paquet de réponse est classé "établi". À l'inverse, une erreur ICMP est "liée", et un paquet d'erreur ICMP qui ne correspond à aucune connexion connue est catégorisé "invalide".

#### Connection Tracking aides

Grâce à l'utilisation de modules de plugin, CT peut prendre connaissance des protocoles de la couche application, et donc comprendre que deux connexions distinctes ou plus sont "liées". Considérons, par exemple, le protocole FTP. Une connexion contrôle est établie, mais chaque fois que des données sont transférées, une connexion séparée est établie pour les transférer. Lorsque le module *nf\_conntrack\_ftp* est chargé, le premier paquet d'une connexion de données FTP sera classé comme "lié" à la place de "nouveau", comme il fait logiquement partie d'une connexion existante.

Les aides n'inspectent qu'un paquet à la fois. Si des informations vitales pour CT sont divisées en deux paquets — en raison de la fragmentation IP ou de la segmentation TCP — l'aide ne reconnaitra pas nécessairement les modèles et ne saura donc pas s'acquitter de son fonctionnement. La fragmentation IPv4 est traitée avec le sous-système CT exigeant la défragmentation, même si la segmentation TCP n'est pas traitée. En cas de FTP, les paquets sont réputés ne pas être segmentés "près de" la commande PASV avec des tailles de secteur (MSS) et ne sont donc pas traités par Netfilter.

C:\Documents and Settings\Administrateur\Bureau\images.jpg

Logo projet Iptables/Netfilter

## Histoire

Le projet netfilter/iptables a été lancé en 1998 par Rusty Russell (en), qui était aussi l'auteur du programme précédent, ipchains. Bien que le projet ait grandi, il a fondé le *Netfilter Core Team* (ou simplement *coreteam*, équipe de développement principale) en 1999. Le logiciel qu'ils produisent (appelé netfilter à partir de maintenant) est sous la licence GNU General Public License (GPL), et a été intégré dans Linux 2.3 en mars 2000. En août 2003, Harald Welte a été fait président de la coreteam et, en avril 2004, suivant des recherches intensives du projet Netfilter dans des produits commerciaux qui ont distribué le logiciel sans respecter les termes de la licence, Harald Welte a réussi à obtenir une injonction historique contre Sitecom Allemagne qui a refusé de suivre les termes de la licence GPL. En septembre 2007, Patrick McHardy, qui a dirigé le développement de ces dernières années, a été élu nouveau président de la coreteam.

Avant iptables, les principaux logiciels de création de pare-feu sur Linux étaient ipchains (noyau linux 2.2) et ipfwadm (noyau linux 2.0), basé sur ipfw, un programme initialement conçu sous BSDs. ipchains et ipfwadm a modifié le code réseau directement, afin de leur permettre de manipuler les paquets, comme il n'y avait pas de paquet-cadre de contrôle général jusqu'au Netfilter.

Considérant que ipchains et ipfwadm combinaient le filtrage de paquets et NAT (en particulier les trois types de NAT, appelés le masquage, la transmission de port et la redirection), Netfilter sépare les opérations de paquets en plusieurs parties, décrites ci-dessous. Chacune se connecte à différents points d'accès dans les accroches Netfilter pour inspecter les paquets. Les sous-systèmes de connexion suivr (Connection Tracking) et de NAT sont plus généraux et plus puissants que les versions inférieures dans ipchains et ipfwadm.

**Le serveur Amon**

**Amon** est une méta distribution GNU/Linux, l'un des modules proposés par le projet EOLE, basée sur les versions LTS d'Ubuntu et qui répond au besoin d'automatisation de l'installation de serveurs pare-feu.

L'installation et la configuration du module Amon sont simplifiées, notamment grâce à l'installateur sans interaction et à l'Interface de configuration du module fournie par le projet. Cette interface permet également une intégration et une interaction avec les autres modules EOLE.



Page d’accueil du serveur Amon

## Présentation

Le module Amon permet de partager en toute sécurité un accès Internet entre les sous-réseaux d'un réseau local. Installé sur un serveur dédié, équipé de deux, trois, quatre ou cinq interfaces réseau, il permet d'organiser au mieux l'architecture réseau d'un établissement. Des modèles de règles de pare-feu sont disponibles pour chaque architecture. Vous pouvez les utiliser tels quels ou bien les modifier à votre convenance. Un outil spécifique, Era1, est à votre disposition pour effectuer ce travail.

Il est également possible de créer un réseau virtuel privé (RVP) entre l'établissement (une structure administrative) et un concentrateur académique (par exemple le module Sphynx). Ce réseau virtuel privé permet de sécuriser les flux sensibles au travers d'Internet.

Entièrement basé sur des logiciels libres, le module Amon est diffusé gratuitement par le Ministère de l'Éducation Nationale Français.

La méta distribution Amon est disponible sur le site du Pôle de Compétences EOLE2

## Historique

### Le projet initial

Le projet répondait initialement à une demande locale de l'Académie de Dijon pour répondre à un besoin identifié. Amon reposait alors sur ipchains. En 2001 il devient un projet National à la demande du Ministère National de l'Enseignement Supérieur et de la Recherche (MENESR) et donne naissance à une version 1.0 avec le passage à iptables. Les buts étaient :

* protéger les élèves ;
* protéger les données administratives ;

Une version 1.5 du module voit le jour en 2003. Plutôt destiné aux collèges et aux lycées, le module Amon commence à être utilisé dans les écoles primaires en 2005.



Logo du MENESR

### Amon Nouvelle Génération

La technologie change, EOLE repose à partir de 2006 sur les versions LTS d'Ubuntu. Une ré-écriture des outils de configuration est réalisée.

La méta-distribution Amon tourne désormais sur les architectures i386 et AMD64.

En 2008 le module Amon intègre le pare-feu authentifiant NuFW3

### Amon 2.3

## Principales fonctionnalités d'Amon

* routage ;
* support Vlan;
* authentification des utilisateurs ;
* filtrage réseau
* filtrage de site amélioré  ;
* réseau virtuel privé ;
* suivi détaillé de la navigation web ;
* mises à jour automatiques ;
* journalisation des fichiers logs ;
* détection d'intrusions ;
* service de cache web ;
* administration simplifiée ;
* statistiques sur l'état du système ;
* statistiques d'utilisation.

## Les avantages du module Amon

La configuration est aisée car centralisée dans l'Interface de configuration du module.

L'intégration avec la suite de modules EOLE proposés par le projet est gérée.