SYSTEME ET INFRASTRUCTURE

Firewalls et contrôle des flux

Principes, mise en oeuvre et outils open source

Maxime **BESSON**
Expert technique

Smile
OPEN SOURCE SOLUTIONS

www.smile.fr • +33 (0)1 41 40 11 00 • contact@smile.fr
www.smile-oss.com • blog.smile.fr • twitter: @GroupeSmile

PRÉAMBULE

Smile

Smile est une société d'ingénieurs experts dans la mise en œuvre de solutions open source et l'intégration de systèmes appuyés sur l'open source. Smile est membre de l'APRIL, l'association pour la promotion et la défense du logiciel libre, de Alliance Libre, PLOSS, et PLOSS RA, des associations clusters régionaux d'entreprises du logiciel libre.

Smile compte 480 collaborateurs en France, 600 dans le monde, ce qui en fait la première société en France spécialisée dans l'open source.

Depuis 2000, environ, Smile mène une action active de veille technologique qui lui permet de découvrir les produits les plus prometteurs de l'open source, de les qualifier et de les évaluer, de manière à proposer à ses clients les produits les plus aboutis, les plus robustes et les plus pérennes.

Cette démarche a donné lieu à toute une gamme de livres blancs couvrant différents domaines d'application. La gestion de contenus (2004), les portails (2005), la business intelligence (2006), les frameworks PHP (2007), la virtualisation (2007), et la gestion électronique de documents (2008), ainsi que les PGIs/ERPs (2008). Parmi les ouvrages publiés en 2009, citons également « Les VPN open source », et « Firewall est Contrôle de flux open source », et « Middleware », dans le cadre de la collection « Système et Infrastructure ».

Chacun de ces ouvrages présente une sélection des meilleures solutions open source dans le domaine considéré, leurs qualités respectives, ainsi que des retours d'expérience opérationnels.

Au fur et à mesure que des solutions open source solides gagnent de nouveaux domaines, Smile sera présent pour proposer à ses clients d'en bénéficier sans risque. Smile apparaît dans le paysage informatique français comme le prestataire intégrateur de choix pour accompagner les plus grandes entreprises dans l'adoption des meilleures solutions open source.

Ces dernières années, Smile a également étendu la gamme des services proposés. Depuis 2005, un département consulting accompagne nos clients, tant dans les phases d'avant-projet, en recherche de solutions, qu'en accompagnement de projet. Depuis 2000, Smile dispose d'un studio graphique, devenu en 2007 Smile Digital – agence interactive, proposant outre la création graphique, une expertise e marketing, éditoriale et interfaces riches. Smile dispose aussi d'une agence spécialisée dans la TMA (support et l'exploitation des applications) et d'un centre de formation complet, Smile Training. Enfin, Smile est implanté à Paris, Lille, Lyon, Grenoble, Nantes, Bordeaux, Poitiers, Aix-en-Provence et Montpellier. Et présent également en Espagne, en Suisse, au Benelux, en Ukraine et au Maroc.

Quelques références

Intranets et Extranets

Société Générale - Caisse d'Épargne - Bureau Veritas - Commissariat à l'Energie Atomique - Visual - CIRAD - Camif - Lynxial - RATP - Sonacotra - Faceo - CNRS - AmecSpie - INRA - CTIFL - Château de Versailles - Banque PSA Finance - Groupe Moniteur - Vega Finance - Ministère de l'Environnement - Arjowiggins - JCDecaux - Ministère du Tourisme - DIREN PACA - SAS - CIDJ - Institut National de l'Audiovisuel - Cogedim - Diagnostica Stago Ecureuil Gestion - Prolea - IRP-Auto - Conseil Régional Ile de France - Verspieren - Conseil Général de la Côte d'Or - Ipsos - Bouygues Telecom - Prisma Presse - Zodiac - SANEF - ETS Europe - Conseil Régional d'Ile de France - AON Assurances & Courtage - IONIS - Structis (Bouygues Construction) - Degrémont Suez - GS1-France - DxO - Conseil Régional du Centre - Beauté Prestige International - HEC - Veolia

Internet, Portails et e-Commerce

Cadremploi.fr - chocolat.nestle.fr - creditlyonnais.fr - explorimmo.com - meilleurtaux.com - cogedim.fr - capem.fr - Editions-cigale.com - hotels-exclusive.com - souriau.com - pci.fr - odit-france.fr - dsv-cea.fr - egide.asso.fr - Osmoz.com - spie.fr - nec.fr - vizzavi.fr - sogeposte.fr - ecofi.fr - idtgv.com - metro.fr - stein-heurtey-services.fr - bipm.org - buitoni.fr - aviation-register.com - cci.fr - eaufrance.fr - schneider-electric.com - calypso.tm.fr - inra.fr - cnil.fr - longchamp.com - aesn.fr - bloom.com - Dassault Systemes 3ds.com - croix-rouge.fr - worldwatercouncil.org - Projectif - credit-cooperatif.fr - editionsbussiere.com - glamour.com - nmmedical.fr - medistore.fr - fratel.org - tiru.fr - faurecia.com - cidil.fr - prolea.fr - bsv-tourisme.fr - yves.rocher.fr - jcdecaux.com - cg21.fr - veristar.com - Voyages-sncf.com - prismapub.com - eurostar.com - nationalgeographic.fr - eau-seine-normandie.fr - ETS Europe - LPG Systèmes - cnous.fr - meddispar.com - Amnesty International - pompiers.fr - Femme Actuelle - Stanhome-Kiotis - Gîtes de France Bouygues Immobilier - GPdis - DeDietrich - OSEO - AEP - Lagardère Active Média - Comexpo - Reed Midem - UCCIFE - Pagesjaunes Annonces - 1001 listes - UDF - Air Pays de Loire - Jaccede.com - ECE Zodiac - Polytech Savoie - Institut Français du Pétrole - Jeulin - Atoobi.com - Notaires de France - Conseil Régional d'Ile-de-France - AMUE

Applications métier

Renault - Le Figaro - Sucden - Capri - Libération - Société Générale - Ministère de l'Emploi - CNOUS - Neopost - Industries - ARC - Laboratoires Merck - Egide - ATEL-Hotels - Exclusive Hotels - CFRT - Ministère du Tourisme - Groupe Moniteur - Verspieren - Caisse d'Epargne - AFNOR - Souriau - MTV - Capem - Institut Mutualiste Montsouris - Dassault Systèmes - Gaz de France - CAPRI Immobilier - Croix-Rouge Française - Groupama - Crédit Agricole - Groupe Accueil - Eurordis - CDC Arkhineo

Applications décisionnelles

IEDOM – Yves Rocher - Bureau Veritas - Mindscape – Horus Finance – Lafarge – Optimus – CecimObs – ETS Europe – Auchan Ukraine – CDiscount – Maison de la France – Skyrock – Institut National de l'Audiovisuel – Pierre Audouin Consultant – Armée de l'air – Jardiland – Saint-Gobain Recherche – Xinek – Projectif – Companeo – MeilleurMobile.com – CG72 – CoachClub

Ce livre blanc

Ce livre blanc, consacré aux principes et outils de firewalling et de contrôle de flux, est le premier volume d'une collection traitant des outils d'infrastructure, dans laquelle on peut ranger également le livre blanc intitulé « Plateformes web Hautes Performances - Principes d'architecture et outils open source », paru début 2009.

Selon le schéma habituel de nos livres blancs, nous présentons ici à la fois les concepts fondamentaux, et une sélection des meilleurs outils.

Nous exposons les caractéristiques de chacun, les possibilités et outils de leur mise en œuvre et configuration, afin d'aider le lecteur dans la sélection d'outils adaptés à chaque contexte d'utilisation.

Table des matières

Introduction

Pourquoi filtrer ?

Les menaces contre les systèmes d'information sont multiples, qu'il s'agisse de menaces externes, ou parfois internes à l'entreprise, intentionnelles ou non, il est devenu vital de prêter une attention permanente à la sécurité de ces réseaux.

Un grand nombre de ces menaces prennent la forme d'une attaque directe contre une machine. Qu'il s'agisse d'un commutateur réseau, d'un serveur ou d'une simple station de travail, tout système expose différents *services* que les attaquants peuvent exploiter afin de compromettre la machine.

Une autre variété de menace est indirecte : la machine n'est plus compromise directement par une attaque réseau originaire de l'extérieur, mais par son propre utilisateur. Il peut s'agir par exemple d'un virus déclenché en ouvrant un document ou un lien internet malicieux. Dans ce cas, l'attaque en elle même n'est pas toujours détectable, mais ses conséquences sont visibles : envoi de *spam*, participation à des attaques massives de déni de service, etc.

Dans la plupart des cas, une attaque informatique transite par un composant réseau. Les firewalls sont des équipements qui viennent se placer sur le chemin réseau entre les machines et l'extérieur, et qui sont capables de détecter les utilisations anormales et menaces et de bloquer. On parle également de filtre réseau.

Routage et filtrage

La notion de firewall, est souvent liée à celle de routage c'est à dire l'acheminement des flux réseau entre les différentes machines. En effet on ne peut raisonnablement installer des systèmes complexes de filtrage réseau sur chaque machine. De plus, en cas de compromission il est indispensable de pouvoir observer le comportement d'une machine de l'extérieur, car les propres systèmes de sécurité de la machine compromise ne sont plus dignes de confiance.

Les firewalls sont donc généralement installés sur des équipements de routage, dont ils sont une partie intégrante. Ce qui signifie que le routage peut être modifié par une décision du firewall, et que le firewall appliquera des règles de filtrage différentes selon l'origine et la destination du trafic.

Ainsi, tout le trafic à destination d'un réseau distant sera soumis au filtrage. L'équipement de routage devient alors le point central de filtrage, même si on continue généralement à appliquer un filtrage supplémentaire sur chaque machine, car en matière de sécurité, la redondance n'est pas superflue.

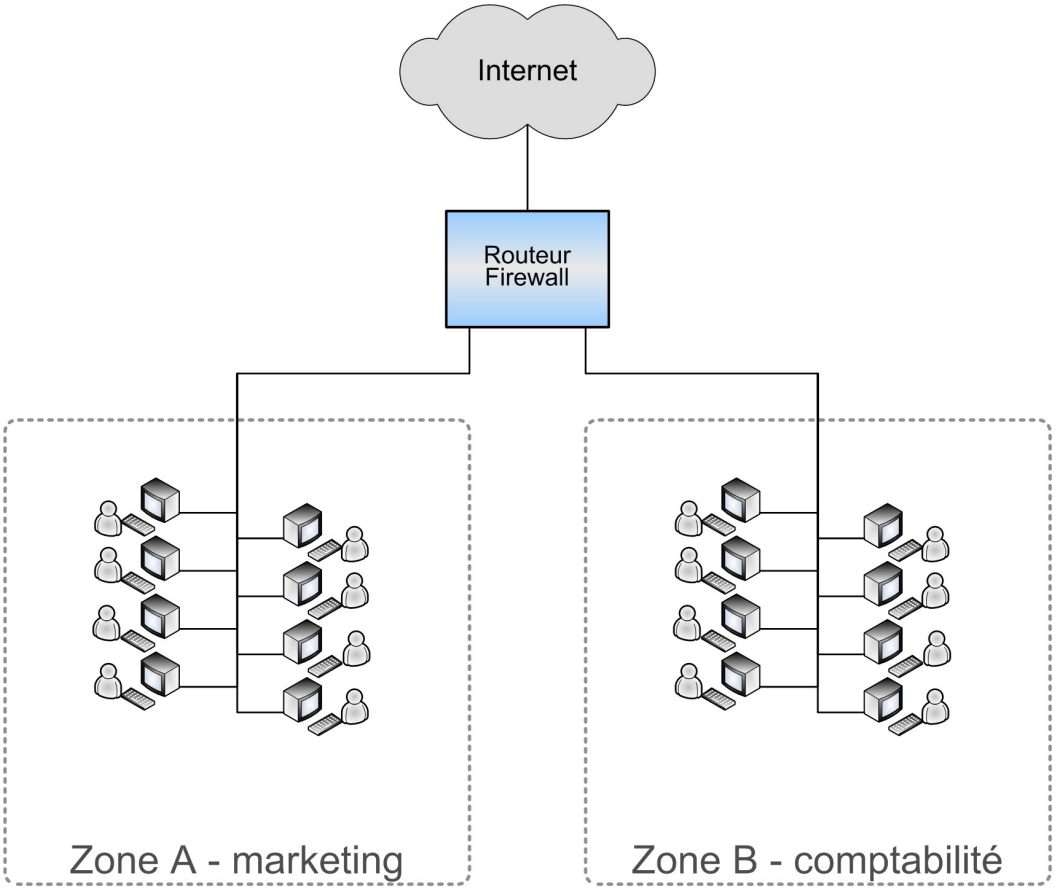
Politique de filtrage

Un des principes de base de la sécurité réseau est la notion de *zone*. La plupart des réseaux peuvent être découpés en zones pour lesquelles une même politique de sécurité s'applique. Ces zones correspondent très souvent à des sous-réseaux, c'est à dire des ensembles de machines qui communiquent directement entre elles sans passer par un routeur. Si une zone peut être composée d'un ou plusieurs sous-réseaux, il est rare qu'un même sous-réseau fasse partie de plusieurs zones, en particulier lorsque la majorité du filtrage est effectué par les routeurs.

Les zones se caractérisent généralement par leur niveau de confiance. Ainsi, le réseau de l'entreprise est généralement considéré comme *plus sûr* que l'Internet, mais *moins sûr* que le sous-réseau dédié aux serveurs.

On peut généralement résumer la configuration des firewalls à une *politique de filtrage*, qui décrit simplement les *interactions entre les zones*. La politique de filtrage réseau n'est bien sûr qu'un sous-ensemble de la politique de sécurité de l'entreprise, et n'assure pas à elle seule la sécurité de l'ensemble des ressources informatiques.

Voici pour illustrer ces propos, un exemple de réseau d'entreprise.



Sur cet exemples on a identifié trois zones : internet, la zone A constituée de deux sous-réseaux (par exemple : deux étages différents d'un bâtiment), et la zone B, plus sensible, constituée des postes stockant des données comptables.

Et voici un exemple de politique de sécurité applicable à ce réseau :

- Autoriser les connexions de la zone A vers internet.
- Autoriser les connexions de la zone B vers internet uniquement sur certains services de confiance (fournisseurs, etc.)
- Refuser les connexions provenant d'Internet, et les connexions entre les deux zones.

	… vers Zone A	… vers Zone B	… vers Internet
De Zone A …	N/A	Interdit	Autorisé
De Zone B …	Interdit	N/A	Filtré
De Internet …	Interdit	Interdit	N/A

Différents niveaux pour différents usages

Chaque trame qui circule sur un réseau est composée d'enveloppes imbriquées qui correspondent aux différents niveaux de protocoles. Au moment où le firewall reçoit du trafic, il peut inspecter les informations contenues plus ou moins en profondeur, on parle de niveau de filtrage.

Filtrage réseau

Les filtres réseau se contentent d'examiner les couches 3 et 4, c'est à dire IP (routage) et TCP (transport) de chaque paquet: leur adresse d'origine, de destination et le protocole contacté (sous la forme d'un numéro de port). De ce fait, les filtres qui travaillent à ce niveau sont capables de travailler avec des routeurs (niveau 3) et permettent un niveau de filtrage suffisant dans la majorité des cas. Leur principale utilisation est de rendre une zone inaccessible depuis une autre, ou de limiter l'accès à certains services (mail, web...).

Filtrage applicatif

Les filtres applicatifs travaillent au niveau 7, le niveau « application », du modèle OSI c'est à dire qu'ils analysent les données applicatives elles-mêmes, et non les seules « enveloppes »; ils ont donc une meilleure compréhension du trafic qu'ils analysent. En revanche, ils requièrent un traitement spécifique au protocole applicatif utilisé pour pouvoir travailler.

Par exemple, un firewall applicatif permet de ne restreindre l'accès au web qu'à certaines URL précises, ou à filtrer le courrier indésirable (spam, virus) après avoir analysé le contenu de chaque e-mail.

La différence avec un firewall réseau est qu'un firewall applicatif permet un contrôle plus fin des accès, mais n'est pas aussi polyvalent : chaque protocole étant particulier, il faut mettre en place un filtre applicatif par protocole, ou utiliser un filtre « générique » qui intègre la connaissance de plusieurs protocoles, mais qui n'est pas souvent aussi performant qu'un logiciel spécialisé dans un seul protocole.

Cependant, les filtres applicatifs sont souvent nécessaires, en particulier pour protéger les mails et le web, car de plus en plus d'attaques prennent aujourd'hui pour cible ces médias.

Qualité de service

Le rôle d'un firewall ne se limite pas toujours à autoriser ou interdire l'accès à une ressource. Il est possible d'utiliser un firewall dans un rôle plus qualitatif, plus fin.

La bande passante disponible sur un équipement de routage est souvent limitée, notamment vers et en provenance d'Internet ou d'autres réseaux distants. Or certaines applications, en particulier la VOIP qui fera l'objet d'un autre livre blanc de Smile, ont des exigences particulières vis à vis de la qualité et l'encombrement des liaisons réseaux.

Selon le protocole utilisé, selon la source et la destination du trafic, et parfois même selon des critères applicatif, il est possible de favoriser ou de pénaliser un type de trafic par rapport à un autre.

Certains algorithmes de gestion de la qualité de service permettent de garantir une bande passante minimale, et parfois un temps de traitement maximal du trafic ainsi que la possibilité de fournir des « pics » de bande passante pendant une courte durée.

Suivi de connexion

Interdire l'ensemble du trafic réseau dans un sens mais pas dans l'autre est irréaliste, en réalité, ce que l'on cherche à faire c'est plutôt d'interdire l'établissement de connexions dans un sens, et l'autoriser dans un autre. Mais une fois la connexion ouverte, le trafic doit être capable de circuler dans les deux sens entre le client et le serveur.

Dans le cas de protocoles comme le Web (HTTP), le suivi de connexion prend tout son sens, car le gros du trafic est constitué non pas des requêtes adressées par les clients, mais des réponses émises par le serveur. Les réponses doivent évidemment traverser le firewall, alors qu'une requête provenant d'Internet à destination d'un hôte du réseau local ne doit pas être autorisée.

Pour réaliser cela, les firewalls disposent d'un mécanisme appelé **suivi de connexion** (*connection tracking*), qui leur permet de garder une trace des connexions établies, afin d'autoriser tout les échanges relatifs à une même session.

Le suivi de connexion permet également de mettre en place une translation d'adresse (NAT). Dans ce cas, le firewall réécrit l'adresse source ou l'adresse de destination de chaque paquet relatif à une connexion.

L'utilisation du NAT est indispensable dès lors que l'on possède moins d'adresses IP que de machines, ce qui est le cas le plus général.

Enfin, le suivi de connexion à parfois besoin d'examiner le contenu des paquets pour travailler, c'est notamment le cas du protocole FTP. Ce protocole archaïque mais néanmoins toujours utilisé nécessite la communication sur plusieurs ports, qui sont choisi lors d'une négociation sur un port connu. Il faut donc que le firewall capture cette négociation pour pouvoir ouvrir automatiquement les ports demandés. Le même problème se pose avec certains protocoles multimédias comme le H.323 (voix sur IP). On appelle cela du suivi de connexion applicatif (ou niveau 7).

Dans la pratique, on utilise presque uniquement des firewalls permettant un suivi de connexion.

Quel matériel ?

Le filtrage simple du trafic réseau n'est pas une activité très exigeante en termes de matériel, la plupart des firewalls commerciaux sont construits autour de puces peu puissantes. Le besoin de performance apparaît souvent au fur et à mesure que l'on remonte les couches OSI : plus on examine les paquets plus il faut de puissance, en particulier avec des outils de détection d'intrusion qui doivent appliquer un grand nombre de traitements sur chaque paquet réseau traversant la machine.

Pour un firewall/routeur travaillant principalement au niveau 3 et proposant quelques services de niveau 7 comme un proxy et un anti-spam simple, même une machine de bureau peut convenir. Dans beaucoup de petites entreprises de telles machines sont recyclées en firewall.

Pour supporter des débits élevés (supérieurs à 20-30 Mb/s), tout en offrant des services niveau 7 plus conséquents comme de la détection d'intrusion, une analyse virale du trafic web ou un anti-spam complexe, il faudra souvent s'orienter vers du matériel plus haut de gamme, un serveur doté de cartes réseau de meilleure qualité et de meilleures capacités de traitement (processeur, disques rapides...).

L'offre OpenSource en matière de filtrage

Filtrage niveau 3

La plupart des solutions de niveau 3 interagissent fortement avec le système de routage de paquets, c'est pourquoi elles font partie intégrante des principaux systèmes d'exploitation open source.

Netfilter (iptables)

Netfilter est la solution native de filtrage réseau sous Linux. Cette solution est plus connue sous le nom de son principal outil d'administration en ligne de commande : *iptables*.

Ses nombreux modules lui permettent de filtrer le trafic selon des critères très divers : sous-réseau d'origine, de destination, ports, heure de la journée et, via des programmes utilisateurs, n'importe quel autre critère. Il dispose également de modules permettant le suivi d'état de protocoles tels que FTP ou H.323, qui nécessitent une inspection des paquets au niveau 7.

Son principal défaut est son système de configuration : *Netfilter* se configure par l'exécution successive d'un certain nombre de commandes *iptables*, à la syntaxe peu intuitive. Cependant il existe de nombreux outils, tels que *ufw*, qui simplifient la configuration en exécutant automatiquement ces commandes à partir d'un fichier écrit dans une syntaxe plus lisible.

L'inconvénient de ces outils simplifiant l'interface est que l'on perd en fonctionnalités puisqu'ils n'implémentent en général qu'un sous-ensemble des fonctions de *NetFilter*.

NuFW

NuFW est une surcouche à Netfilter qui lui rajoute la *gestion des utilisateurs*.

En effet, les firewalls ont un point de vue sur le trafic uniquement au niveau réseau : lorsqu'un administrateur veut déterminer quelle personne physique est responsable d'une action, il doit utiliser une autre source d'information (DNS, domaine windows, etc.). De plus, si plusieurs utilisateurs partagent le même poste ou en changent régulièrement il devient difficile de tracer leurs activités.

NuFW comble ces lacunes en authentifiant chaque utilisateur du réseau. Avant de pouvoir accéder au réseau l'utilisateur lance un programme sur sa machine, qui l'authentifie par un mot de passe ou un certificat numérique. Ensuite, le firewall est capable de gérer des règles par utilisateur et non plus par sous-réseau. A chaque connexion, le firewall dialogue avec le programme client pour valider l'accès.

Ceci permet une plus grande souplesse d'utilisation : un administrateur réseau qui dispose d'un accès privilégié à ses serveurs le conserve lors de ses déplacements, et une tierce personne ne peut plus essayer d'usurper son adresse réseau pour obtenir le même accès. De plus cela permet de relier directement un utilisateurs à chaque alerte levée par le firewall.

A ce jour aucun autre produit ne propose cette fonctionnalité : les autres firewall dits « authentifiants » se contentent en réalité de lier un utilisateur à une IP pendant la phase d'authentification puis travaillent avec cette IP ce qui est beaucoup moins fiable.

Enfin *NuFW* peut être utilisé comme source d'authentification par des services réseau, puisqu'il exporte une base de donnée SQL contenant le propriétaire de chaque connexion. N'importe quelle application capable d'accéder à cette base peut déterminer qui est l'utilisateur qui vient de se connecter à l'application.

Le principal défaut de *NuFW* est la nécessité d'installer un outil côté client sur les postes des utilisateurs.

pf

pf (*Packet Filter*) est la solution native de filtrage réseau sous les systèmes dérivés de BSD, c'est à dire aujourd'hui FreeBSD, OpenBSD, NetBSD et DragonFlyBSD.

En plus de permettre le filtrage de paquets, *pf* s'interface avec le reste de la pile réseau de ces systèmes, en particulier le routage, la QoS et les systèmes de haute disponibilité. Les interactions de *pf* avec le système de haute disponibilité d'OpenBSD sont décrites dans le livre blanc de Smile consacré aux solutions de haute disponibilité.

pf est similaire à Netfilter dans ses possibilités de base : il possède un suivi d'état, est capable de bloquer ou d'accepter du trafic en fonction des adresses et des ports de destination, et est capable de gérer des translations d'adresse.

De plus pf possède un système de tables similaire au module *recent* de *Netfilter* : il permet de gérer en temps réel une liste d'accès, par exemple pour autoriser temporairement une personne à se connecter sans devoir éditer le jeu de règles, ou pour bannir les utilisateurs qui font trop de connexions simultanées à un servir (prévention de DoS, *Denial of Service*, ou de *bruteforcing*).

pf dispose d'un système d'ancres qui permet à un programme utilisateur de rajouter des règles à la volée, ce qui permet de déléguer une partie de l'administration du firewall à des programmes; il est au cœur de l'implémentation d'une solution de haute disponibilité sous OpenBSD.

Enfin, *pf* permet d'effectuer quelques opérations d'« embellissement » du trafic, en particulier le ré-assemblage de paquets fragmentés, le blocage de paquets erronés, etc. Ce qui permet de purifier un peu le réseau et d'éviter certaines attaques qui tirent parti de bugs dans le traitement de paquets erronés sur un OS précis.

Il dispose cependant de moins de modules que *Netfilter*, en particulier concernant les possibilités de suivi de connexion au niveau 7. En effet, le suivi de connexion pour le FTP est assuré par un proxy et non pas directement par *pf*. De même le H.323 n'est pas géré.

Cependant, sa configuration est beaucoup plus simple que celle de *Netfilter*, et son intégration à des outils comme *spamd*, *relayd*, *pfsync* ou *sasyncd* le rendent très attractif pour des utilisations en haute disponibilité.

Le point fort de *pf*, notamment par rapport à *Netfilter*, est sa syntaxe. Voici à titre d'exemple une règle qui autorise le trafic web d'un LAN vers internet.

```
pass out on $net_if proto tcp from $lan_net to any port http
```

La syntaxe est proche du langage naturel, et l'utilisation de variables et de noms symboliques simplifie l'administration.

De plus, un système de « tags » permet d'obtenir une configuration d'un plus haut niveau d'abstraction, où la politique de filtrage est clairement en évidence.

```
# Classification des paquets en entrée du firewall
pass in on $lan from $lan_net to any tag LAN_TO_INTERNET
pass in on $internet from any to $lan_net tag DANGEROUS
pass in on $lan proto tcp from $lan_net to any port smtp tag MAILS_SENT

# Implementation de la politique de filtrage en sortie du firewall
block out log tagged DANGEROUS
pass out on $internet tagged LAN_TO_INTERNET
pass out log on $internet tagged MAILS_SENT
```

Avec un jeu de règles de cette forme, il est possible de facilement modifier une règle sans modifier la politique de filtrage globale, par exemple, pour interdire aux utilisateurs de se connecter à une machine sur internet par *telnet*, il suffit d'ajouter la règle suivante à la partie classification du trafic :

```
pass in on $lan proto tcp from $lan_net to any port telnet tag DANGEROUS
```

Ainsi ces paquets sont classifiés comme étant dangereux, et la partie politique du jeu de règle prendra les mesures appropriées (abandon de la connexion et notification dans le log).

De même, si on souhaite temporairement autoriser le trafic dangereux, pas besoin de changer l'ensemble des règles qui *identifient* ce trafic, il suffit de changer la règle qui *traite* ce trafic :

```
pass out tagged DANGEROUS
```

Dans un réseau complexe avec plusieurs sous-réseaux différents ayant chacun des droits différents, ce type de configuration en deux étapes prend tout son sens.

Choix d'une solution de filtrage réseau

Ainsi, *pf* et *Netfilter* sont très similaires en terme de fonctionnalités générales, et seuls quelques cas particuliers peuvent faire pencher la balance de façon définitive. *Netfilter* sera privilégié lors de la nécessité d'utiliser des modules spécialisés comme par exemple le suivi de connexion niveau 7 d'un protocole exotique. *pf* quant à lui est un système plus unifié, qui permet de gérer en même temps le filtrage, le NAT et la *QoS*.

Il est à noter que *pf* n'est disponible que sous BSD, et que *Netfilter* n'est disponible que sous Linux, bien souvent c'est ce critère qui emporte la décision.

Interfaces de configuration

Les solutions que nous avons présentées sont puissantes et très intégrées au système d'exploitation d'origine. Par conséquent leur configuration n'est pas toujours aisée en particulier pour les administrateurs qui n'ont pas l'habitude des produits Open Source. De plus, chaque produit possède sa propre “culture” en terme d'édition de règles.

Heureusement, il existe des logiciels dont le but est d'automatiser la création d'un jeu de règle, parfois même indépendamment de la solution de filtrage choisie.

Uncomplicated Firewall

Le but du projet *ufw*, pour «*Uncomplicated FireWall*» est de faciliter l'administration d'un firewall *Netfilter* sous Linux. Ce projet vise plus les serveurs que les routeurs, et permet de mettre en place rapidement une sécurité réseau de base. Sa syntaxe est assez proche de *pf* :

```
ufw allow proto tcp from any to any port 22
```

ufw gère la notion d'application, et permet par exemple de désactiver tout les ports utilisés par une même application simultanément :

```
ufw delete allow apache
```

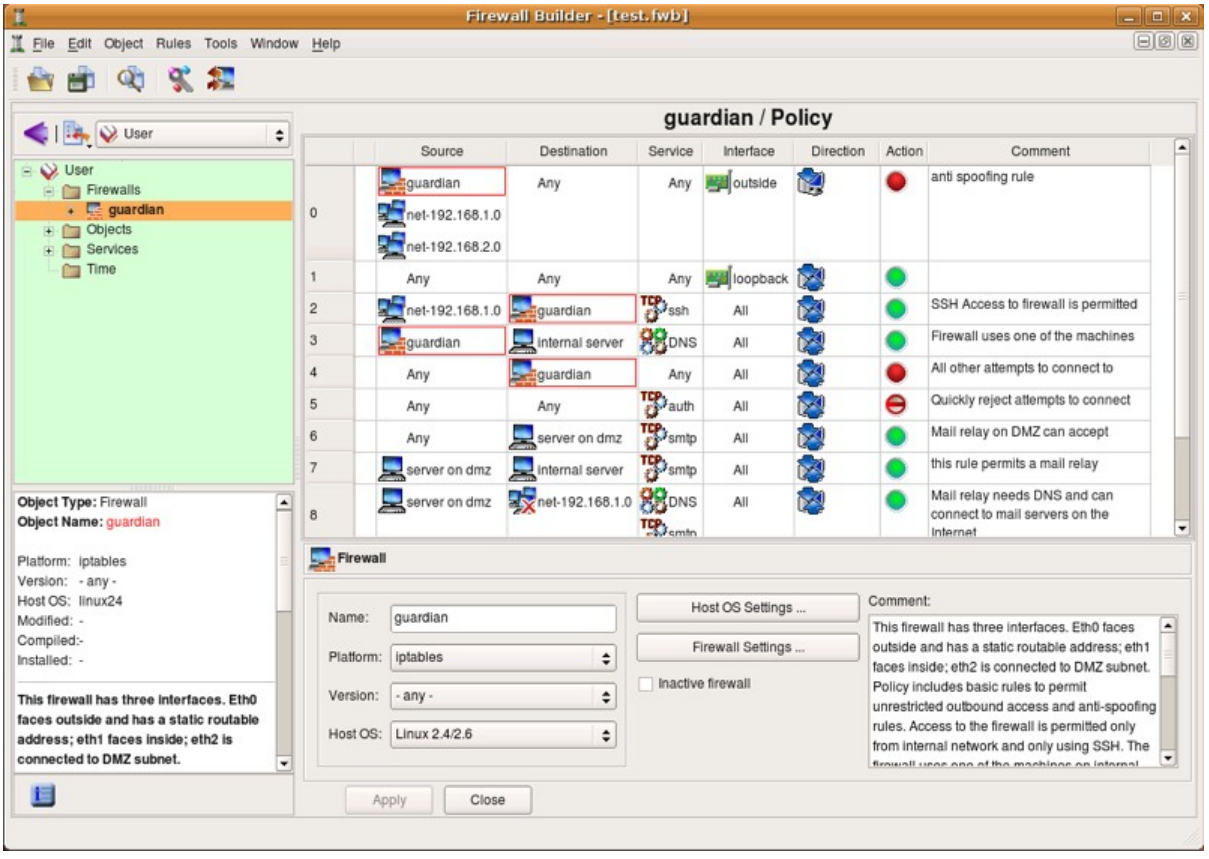
Cette commande, par exemple, permet d'interdire les connexion à Apache.

Firewall Builder

Firewall Builder est une interface graphique permettant de créer une politique de filtrage. Son avantage est qu'il permet de créer et d'installer automatiquement la configuration correspondante sur plusieurs produits différents : firewalls *Netfilter*, firewalls *pf*, et même certains équipements réseau comme les firewalls *Cisco PIX*.

Firewall Builder permet de manipuler des hôtes, des sous-réseaux, des services, gère la notion de plage horaire et permet de configurer automatiquement les translations d'adresse (NAT) et les redirections de port. Il intègre également des assistants permettant de configurer rapidement un réseau typique tel que ceux présentés en fin de ce document.

Voici un exemple :



Les outils tels que *Firewall Builder* permettent de s'affranchir d'une plate-forme particulière et de se concentrer sur la politique de filtrage et non son implémentation. La contrepartie et que, comme toute les interfaces graphiques, *Firewall Builder*, n'implémente pas la totalité des fonctionnalités de ses plate-formes de destination, et certaines applications très spécifiques, ou optimisations, nécessitent une configuration manuelle dans le langage natif de la solution de filtrage. C'est le cas par exemple des modules spécialisés de *Netfilter* ou des fonctionnalités de haute disponibilité de *pf*.

filtrage niveau 7

Squid

Squid est un serveur proxy HTTP. Son but est de servir d'intermédiaire entre les clients, et le Web (pour mémoire, *proxy* en anglais signifie *procuration*). À ce titre, il possède deux fonctionnalités principales :

La mise en cache des requêtes

Le résultat de chaque requête est conservé sur le proxy pour être délivré plus rapidement si la même requête est faite plus tard. Cela économise de la bande passante, améliore le confort de navigation et allège la charge des serveurs.

Le contrôle d'accès

Squid permet d'authentifier ses utilisateurs, et de définir des contrôles d'accès en fonction de l'utilisateur, de son sous réseau, du site web demandé, de l'heure de la journée, etc.

C'est la fonctionnalité de contrôle d'accès qui permet d'utiliser Squid pour filtrer le trafic web, en effet *Squid* possède un mode « transparent » qui permet de l'utiliser sans configuration côté client. Il est donc possible d'utiliser un firewall réseau permettant la redirection de trafic pour forcer tout le trafic web à passer par *Squid*, puis utiliser *Squid* pour filtrer le trafic web.

Squid permet d'utiliser des programmes externes pour filtrer le trafic : l'outil *SquidGuard* permet de gérer automatiquement des listes publiques d'URL classées par catégorie (jeux en ligne, réseaux sociaux, sites pornographiques etc.) et autoriser ou interdire l'accès à des catégories de sites.

Squid peut également s'interfacer avec d'autres programmes. L'utilisation la plus répandue est l'interface avec un antivirus. En effet

ces programmes sont très spécialisés, ils ne comportent pas les fonctionnalités avancées de Squid et ne peuvent donc pas être utilisés directement en tant que proxy. Une autre possibilité est le filtrage des popup, publicités, cookies intempestifs et autres nuisances.

Snort

Snort est un outil d'analyse réseau. Son but premier est la détection et la prévention des intrusions.

Snort est doté d'un moteur de détection basé sur des motifs de recherche, la société SourceFire fournit (contre abonnement payant) des motifs permettant de détecter un très grand nombre d'attaques, de l'exploitation de failles classiques aux tout derniers virus.

Des jeux de règles gratuits sont également disponibles depuis plusieurs sources : la société SourceFire fournit ses motifs gratuitement au bout de 30 jours, et le site EmergingThreats fournit des motifs pour détecter les failles récentes.

Snort dispose de deux modes de fonctionnement : sonde ou filtre.

En mode sonde, le serveur *snort* à simplement un but préventif : il analyse le trafic réseau et envoie des alertes lorsqu'une attaque est détectée.

En mode filtre, *snort* devient un système de prévention d'intrusion : dès qu'un paquet suspect est détecté, la connexion est bloquée ce qui permet d'annuler l'attaque.

En plus de véritables attaques, *snort* est capable de détecter certains comportements non souhaitables sur un réseau d'entreprise, et ainsi bloquer des actions invisibles au niveau 3 comme la connexion à un service de messagerie instantanée à travers un proxy Web, ou le téléchargement de fichiers piratés.

Les règles *snort* sont relativement simples à écrire, ce qui permet de l'adapter à ses propres besoins. Voici un exemple de règle :

```
alert tcp $HOME_NET any <> $EXTERNAL_NET 5555 (msg:"P2P Napster Client Data";
flow:established; content:".mp3"; nocase; classtype:policy-violation; sid:564;
rev:7;)
```

Cette règle déclenche une alerte lorsqu'une connexion TCP sur le port 5555 entre le réseau local et le réseau extérieur contient le message ".mp3"

Snort dispose de beaucoup de motifs, certains sont très stricts et déclencheront des alertes au moindre comportement suspect, même si celui-ci est légitime. Les journaux de *snort* sont rapidement remplis de faux positifs et peuvent décourager l'administrateur. Pour faciliter

l'utilisation d'une sonde *snort*, il sera souvent nécessaire d'utiliser une interface telle que le projet *BASE*.

BASE permet de consulter des statistiques sur les alertes, leur provenance, leur fréquence, et d'autres informations. Ainsi l'administrateur peut affiner les sondes de *snort*, par exemple en désactivant un motif déclenché par une application légitime.

spamd

spamd est un filtre applicatif spécialisé dans le traitement des mails. Son but est de bloquer les spam, c'est à dire l'envoi de courrier en masse.

spamd vient se placer devant le serveur de mails de l'entreprise, et agit lui-même comme un serveur de mails afin d'effectuer un premier travail de filtrage ce qui permet de soulager considérablement le serveur de mails réel.

Contrairement à des systèmes tels que *SpamAssassin*, qui font une analyse en profondeur du contenu du mail via une intelligence artificielle élaborée, *spamd,* lui, se focalise sur la façon dont le mail est émis.

En effet, les machines envoyant de très grandes quantités de spam ne ne soucient pas d'implémenter correctement le protocole SMTP, les mails sont envoyés au moyen de quelques commandes simples et ne sont jamais ré-émis en cas d'erreur. *spamd* dispose de trois « listes » :

La liste noire

La liste noire de *spamd* est une liste de l'ensemble des serveurs dont aucun mail ne sera accepté, de nombreuses organisations à travers le monde traquent les envois massifs de spam et mettent à jour de telles listes. *spamd* est capable de se baser sur ces listes, ainsi que sur une liste maintenue par l'administrateur local, pour savoir quelles sont les machines à ne pas écouter.

La liste blanche

Contraire de la liste noire, la liste blanche contient l'ensemble des hôtes de confiance que l'administrateur a explicitement autorisé à envoyer des mails au serveur protégé. Tout mail arrivant d'une machine sur liste blanche est transmis immédiatement au serveur de mails réel.

La liste « grise »

Il ne s'agit pas d'une véritable liste telle que les deux autres, le terme « liste grise » désigne le comportement de *spamd* face à une machine qui n'est ni sur liste noire, ni sur liste blanche.

Lorsqu'une machine inconnue contacte *spamd* pour la première fois, celui-ci prétend ne pas être prêt à recevoir le mail, et invite la machine à réessayer plus tard.

La plupart des robots de spam sont suffisamment mal conçus pour ignorer cette invitation et abandonnent l'envoi du mail, cependant, si le mail est effectivement ré-émis ultérieurement, le serveur est passé sur liste blanche.

L'inconvénient de cette technique est l'introduction d'un délai systématique la première fois qu'un serveur mail contacte le système protégé. Cela peut être gênant dans des applications de messagerie ou les interlocuteurs ne sont pas connus à l'avance et ou la réactivité est primordiale (recrutement, prospection...).

Enfin, *spamd* dispose d'une fonctionnalité supplémentaire par rapport aux autres logiciels du même type. En effet lorsqu'un hôte sur liste noire se connecte à lui, *spamd* fait semblant d'accepter la connexion, mais au lieu de transmettre le mail au véritable serveur il fait traîner l'échange en acceptant le mail très lentement ce qui fait perdre un temps précieux au spammeur mais consomme très peu de ressources. On appelle cette technique un « *tar pit* », du mot anglais désignant une mare de goudron de laquelle il est difficile de s'échapper.

QoS : Qualité de service

Sous Linux, la qualité de service (*QoS: Quality of Service*) est contrôlée par le noyau via l'outil *tc*. La configuration est indépendante de *Netfilter*, mais un système de marquage de paquet permet à *tc* d'identifier les paquets ayant été interceptés par certaines règles et évite ainsi les redondances.

Sous BSD, la QoS est gérée par *ALTQ*, une structure intégrée à *pf*. Les décisions de filtrage et de priorisation du trafic sont effectuées en même temps.

Un concept important en QoS est la notion de file d'attente : sans une solution de QoS les trames réseau sont émises dans l'ordre ou elles sont déclenchées par les applications. La QoS permet de créer des files d'attentes dans lesquelles sont rangés les paquets, et qui ont des priorités différentes au moment de l'émission de la trame sur le réseau.

Il est évident que la QoS ne peut s'appliquer qu'à des paquets sortants de la machine.

Ceci peut compliquer la création de règles de QoS sur les routeurs, en effet il faudra dans un premier temps identifier le trafic au moment ou il

entre dans le routeur, puis appliquer la règle de QoS au moment ou il sort.

Il existe différents algorithmes permettant de définir une qualité de service, en voici deux parmi les plus utilisés :

PRIO

PRIO, appelé *PRIQ* sous BSD, est un algorithme basique de QoS basé sur les priorités. Chaque flux possède une certaine priorité et lorsqu'un paquet doit être émis par le noyau, celui-ci choisit d'abord celui qui a la plus haute priorité.

Cet algorithme ne permet pas un contrôle fin du flux, mais il permet d'obtenir un minimum de contrôle, et d'éviter que des flux importants ne perturbent des applications qui ont besoin de réactivité.

Cet algorithme est souvent utilisé car il est simple à configurer et élimine en grande partie la perte de réactivité observée lorsqu'une connexion sature en bande passante.

HFSC

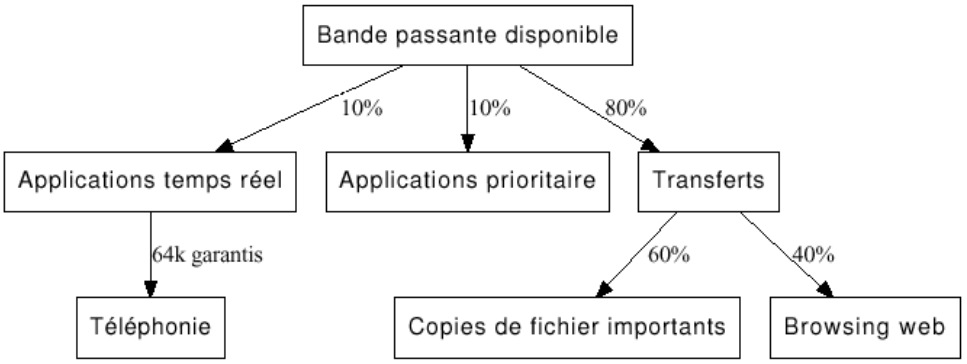
HFSC est un algorithme de QoS permettant de satisfaire simultanément deux exigences :

- La vitesse de transmission
- Le délai de transmission

En effet, certains protocoles nécessitent à la fois un débit de donnée garanti, et un faible délai de transmission, c'est le cas par exemple de la VOIP.

Contrairement à *PRIO*, il est possible avec *HFSC* de contrôler beaucoup plus finement le flux, en définissant une courbe de service pour chaque file d'attente. Cette courbe définit la bande passante maximale, et la possibilité d'allouer une bande passante plus importante au début de la transmission (on parle de « *burst* »). On peut également définir une exigence de traitement en temps réel, pour les paquets qui doivent être transférés dès la réception, et une priorité.

HFSC est un algorithme hiérarchique, c'est à dire qu'il permet de découper la bande passante disponible de façon logique, voici un exemple :



On choisit par exemple d'allouer 10% de toute la bande passante disponible aux applications temps réel (par exemple la téléphonie), 10% aux applications critiques (management, etc.) et le reste aux transferts de fichiers, tout en faisant la distinction entre le trafic web "classique" et par exemple la copie de fichiers vers un environnement externalisé, que l'on priorise.

Produits intégrés

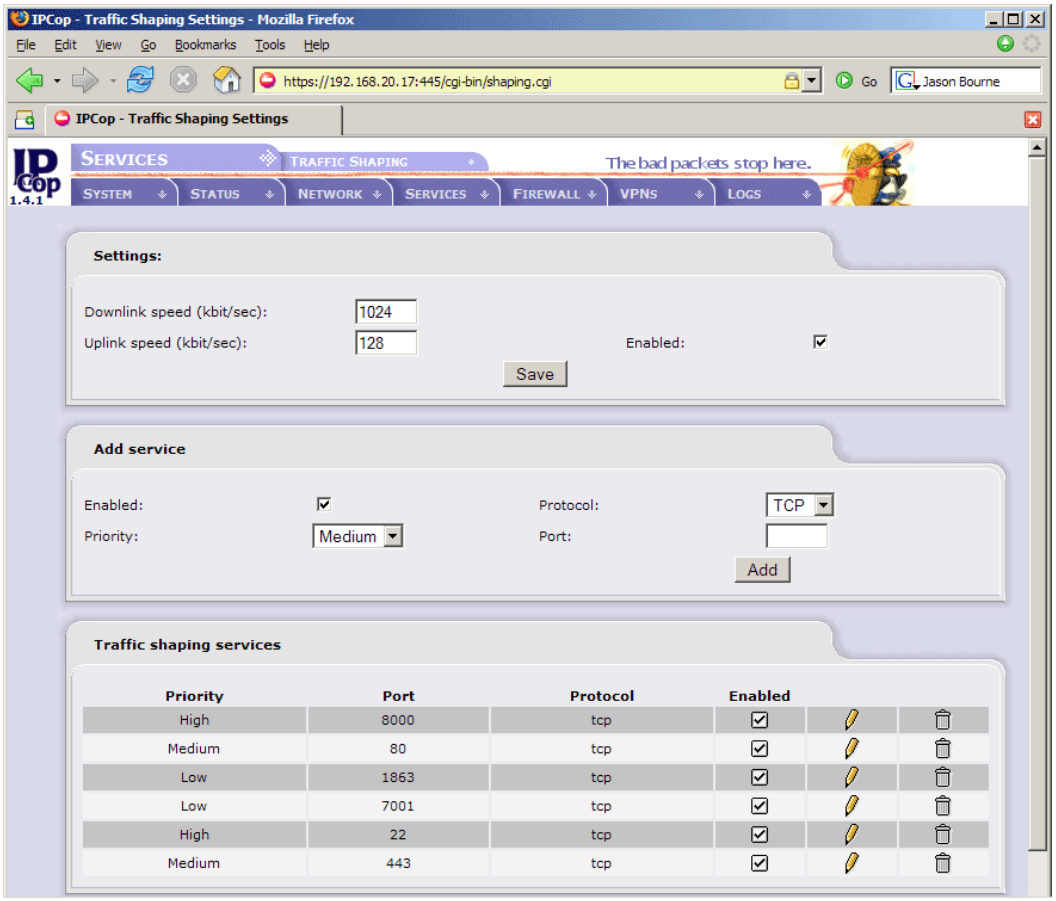
En plus de ces composants individuels permettant de construire un firewall « à la carte », certains projets open source proposent des solutions prêtes à l'emploi construites à partir des produits cités.

IpCop

IpCop est une distribution Linux utilisable comme routeur, firewall et point d'accès VPN, elle est construite autour de solutions Open Source présentées précédemment comme Netfilter, Squid, OpenVPN, et d'autres. Elle est souvent utilisée pour redonner vie à une machine de bureau en tant que firewall domestique ou de petite entreprise en raison de ses faibles pré-requis en matières de ressources.

La configuration d'*IPCop* est très simple et repose principalement sur une interface web. Chaque réseau connecté à la machine est identifié par une couleur représentant son niveau de confiance : rouge pour internet, vert pour le réseau local, etc. L'élaboration d'un jeu de règle est rendu plus intuitif par cette convention.

Cependant, *IPCop* est avant tout un outil de filtrage réseau, ses capacité d'analyse applicative sont limitées.



pfSense

Le projet *pfSense* propose une distribution de FreeBSD spécialement conçue pour une utilisation en firewall/routeur, entièrement configurable via une interface web. Il est l'équivalent d'*IPCop* sous FreeBSD mais possède plus de fonctionnalités que ce dernier, en particulier la possibilité de répartir le trafic entre plusieurs liens internet, dans un but de redondance ou d'augmentation de la bande passante.

Le principal défaut de la plupart des composants présentés précédemment est leur difficulté de prise en main. *pfSense* y remédie en proposant un système unifié, extrêmement complet en terme de fonctionnalités, orienté filtrage niveau 3 et disposant d'un système de paquets permettant d'y installer des filtres de niveau applicatif.

pfSense est un produit très stable, grâce à la robustesse des outils open source qui le composent (en particulier la pile réseau FreeBSD), et utilisé par de nombreuses institutions et particuliers.

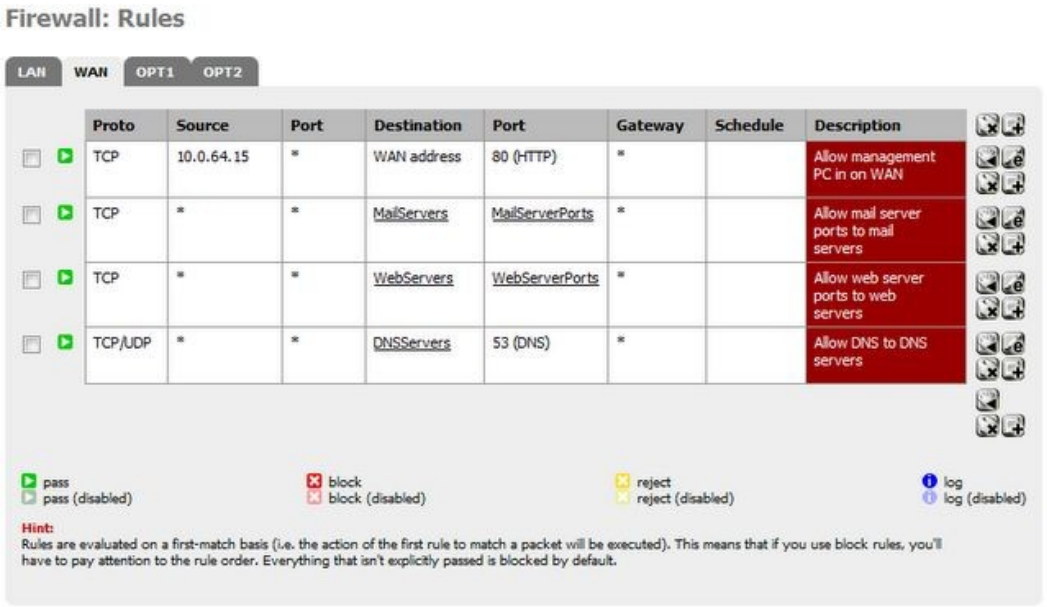
Ses principaux avantages sont :

- Une interface d'administration Web facile à utiliser
- Un système de configuration par alias qui permet de gérer facilement des jeux de règles complexes.
- La cohérence entre les différents composants (routage, filtrage, VPN, QoS)
- Des fonctions avancées de QoS et de haute disponibilité telles que HFSC et CARP/*pfsync* .
- La possibilité de rajouter des composants supplémentaires qui s'intègrent alors dans l'interface (*Squid*, outils de monitoring, etc.)

En revanche, ses capacités de traitement applicatif sont limitées :

- *Snort* n'est pas officiellement disponible sur *pfSense* : il peut être installé via FreeBSD mais n'est pas supporté lors des upgrades et ne dispose pas d'une interface d'administration web intégrée.
- Le système de suivi d'état niveau 7 ne fonctionne que pour le FTP sortant.
- La seule solution antispam disponible est *spamd*

Configuration des règles de filtrage sous *pfSense* :



Untangle

Untangle est un projet similaire à *pfSense*, basé sur Linux, beaucoup plus axé sur le filtrage applicatif que ce dernier au détriment des capacités réseau.

Ses avantages sont :

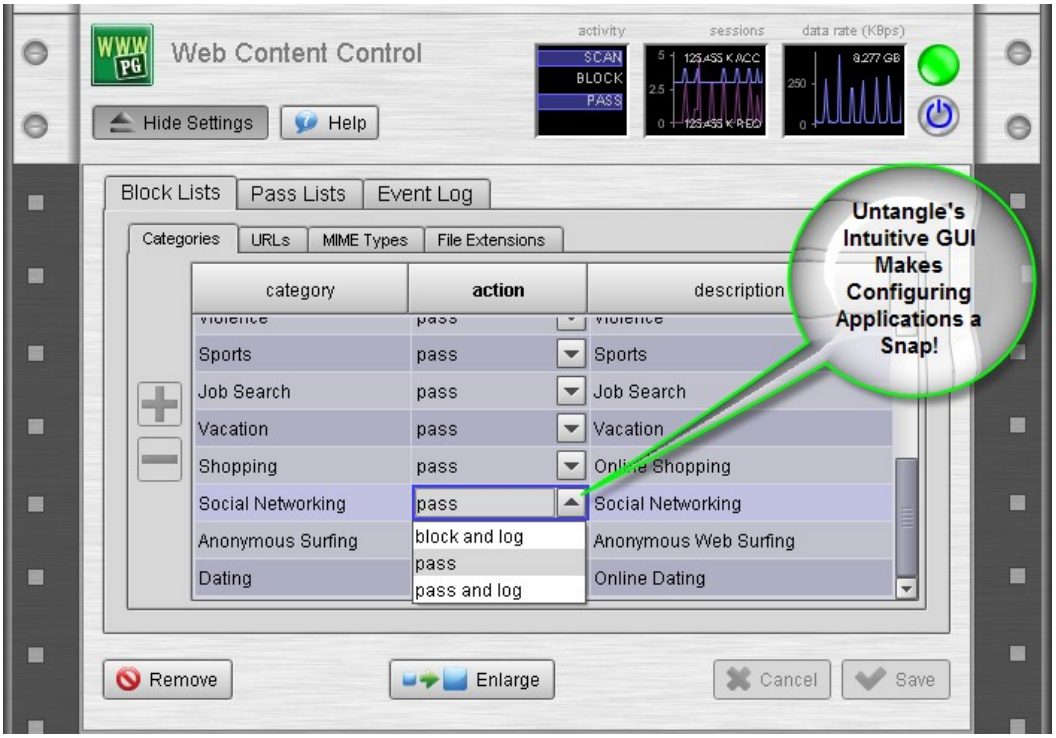
- Intégration d'un filtre antispam, d'un antivirus et d'un IDS.
- Possibilité de management centralisé.

Ses limitations principales sont :

- Pas de redondance.
- N'est pas utilisable comme concentrateur VPN IPSEC

De plus, *Untangle* est capable de fonctionner de façon transparente sur le réseau, on peut donc le placer juste derrière un routeur plus fonctionnel (*pfSense* par exemple) pour lui rajouter des capacités de filtrage applicatif avancées.

Configuration du filtre de contenu sous *Untangle* :



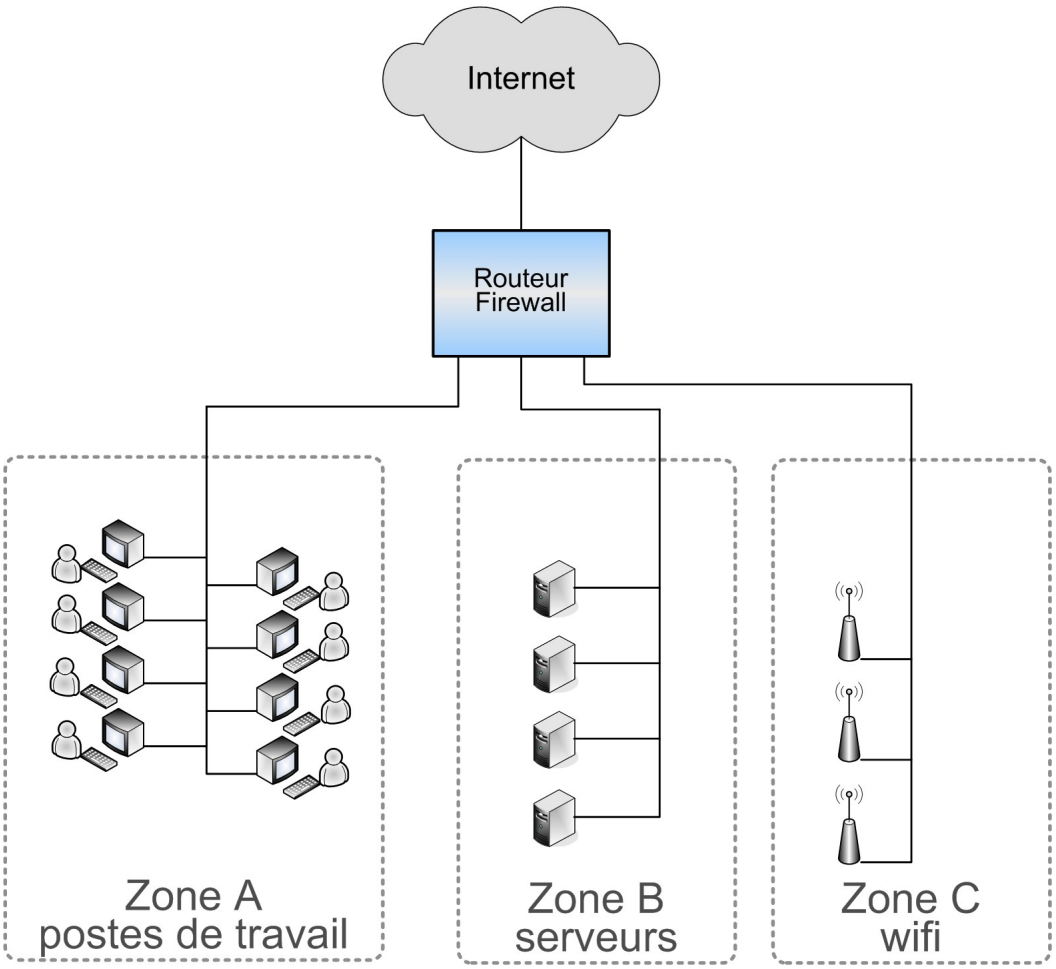
ARCHITECTURES CLASSIQUES

Afin de faciliter la mise en place d'un firewall, on se ramène souvent à une des architectures classiques présentées ici. Cette section présente les pratiques courantes en matière de filtrage dans 3 cas de figure habituels.

Petite entreprise / Agence régionale

Ce premier cas de figure représente un petit réseau. Typiquement constitué de moins d'une centaine de postes, quelques serveurs (mail, intranet, application métier...) et un accès internet. On rencontre ce type de réseau dans les PME/TPE ou les antennes régionales de plus grandes entreprises, auquel cas on rajoutera un VPN comme présenté dans le livre blanc de Smile consacré aux VPN open source.

Plan du réseau



Un réseau classique de petite entreprise contient des stations de travail, des serveurs, et parfois des équipements sans fil (PC portables, PDA...)

Un firewall central sert de routeur pour tous les sous-réseaux correspondant, et filtre tout le trafic vers chaque sous réseau.

Politique de filtrage

Voici la politique mise en place au niveau réseau :

- Tout trafic non autorisé explicitement est interdit.
- L'accès aux services offerts par les serveurs est autorisé depuis internet et depuis les postes de travail et les équipements mobiles.
- Les serveurs, les postes de travail et les équipements mobiles ont accès au Web.
- Les serveurs peuvent relayer les mails et les résolutions de nom vers internet.

Et au niveau applicatif :

- Les accès au Web des postes et des équipements mobiles sont filtrés par *Squid*
- L'arrivée de mails sur le serveur mail depuis internet est protégée par *spamd*
- L'ensemble du trafic est surveillé par *snort*.

Implémentation

Voici une implémentation possible avec *pf* de la politique réseau :

```
# Tout traffic non autorise par une regle est interdit et journalise
block in log
pass out

# Les serveurs sont accessibles depuis internet, les postes, et le wifi
pass in on $inet_if proto tcp from any to $serveurs port $services_ext
pass in on $lan_if proto tcp from $lan_net to $serveurs port $services_int
pass in on $lan_if proto udp from $lan_net to $serveurs port $services_int_udp
pass in on $wifi_if proto tcp from $wifi_net to $serveurs port $services_int
pass in on $wifi_if proto udp from $wifi_net to $serveurs port $services_int_udp

# Les serveurs peuvent envoyer des mails et des requêtes DNS
pass in on $dmz_if proto {tcp,udp} from $serveurs to any port domain
pass in on $dmz_if proto tcp from $serveurs to any port smtp
```

```
# Les clients peuvent accéder au web via le proxy local
pass in on $lan_if proto tcp from $lan_net to localhost port 3128
pass in on $wifi_if proto tcp from $wifi_net to localhost port 3128
```

Particularités

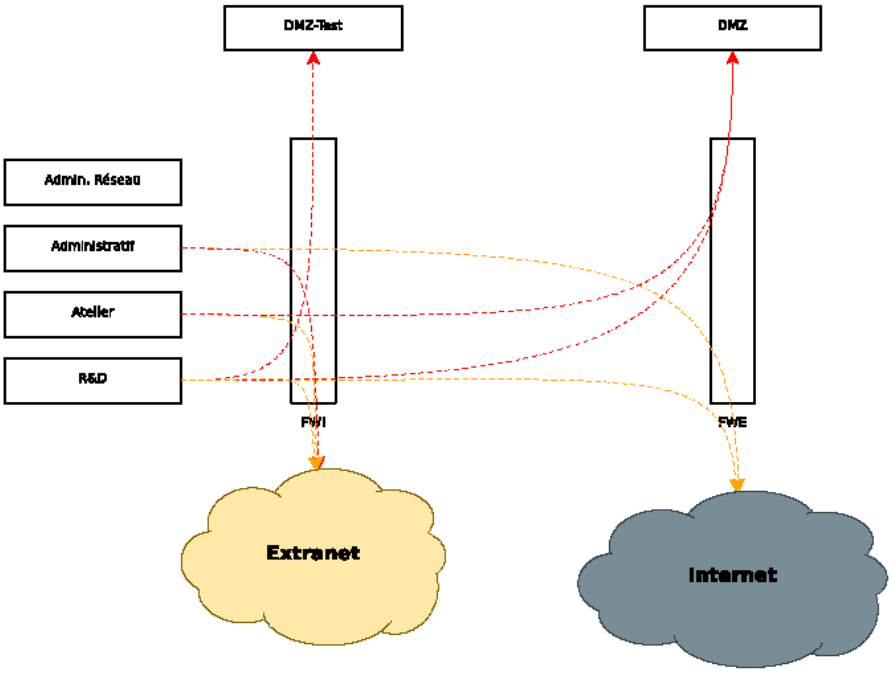
Une architecture de ce type est relativement simple à mettre en place et demande peu de maintenance. Il faudra simplement veiller à tenir à jour les listes d'URL de *Squid*, les blacklist mail de *spamd*, et les motifs de *Snort*.

Entreprise de taille moyenne

Nous avons choisi comme exemple pour illustrer les concepts courants dans les réseaux de taille moyenne, une usine comportant une partie administrative, un atelier, une équipe de recherche et développement travaillant sur les applications métier, et une équipe d'administration réseau.

Cette usine possède 3 types de serveurs : des serveurs de support (messagerie, comptabilité, etc.) infogérés dans un extranet, des serveurs pour les applications métier, hébergés et gérés en interne, et des serveurs de tests pour les futures versions des applications métiers.

Plan du réseau



Les réseaux sont protégés par deux firewalls, un firewall interne *FWI* et un firewall externe *FWE*. Le firewall interne est relativement basique et ne se contente de filtrage niveau réseau. Le firewall externe lui est plus sensible car il assure la connexion avec internet, il devra donc être doté de fonctionnalités de filtrage applicatif et d'un IDS.

Politique de filtrage

Sur le schéma sont montrés les flux principaux :

- L'administratif accède aux applications de l'extranet
- L'administratif accède à internet
- L'atelier n'accède qu'aux applications métier et extranet
- La R&D accède aux serveurs de tests, aux serveurs de production, et à l'extranet

Les autres flux sont :

- Les administrateurs réseau accèdent à tout le réseau.
- La DMZ de tests accède à la DMZ de production et à l'extranet
- La DMZ de production accède à l'extranet
- La DMZ de production est accessible depuis internet (de façon restreinte)

Particularités

Ce type d'architecture est fréquemment rencontré dans les entreprises de taille moyenne disposant d'un extranet, ou d'agences qui viennent s'interconnecter à leur réseau. Souvent, l'ensemble du trafic vers ou en provenance d'internet est centralisé et passe par un point de contrôle unique : le firewall extérieur, généralement situé au siège de l'entreprise.

Bien sûr, pour être fiable, une infrastructure de ce type devra être tolérante aux pannes : un futur livre blanc de Smile expliquera comment mettre en place des couples de firewalls qui agissent comme un unique point de passage, afin d'assurer cette tolérance.

Hébergeur

Un réseau d'hébergeur est légèrement différent d'un réseau de type entreprise, on n'y trouve en effet que des serveurs, et les besoins en filtrage y sont moins élevés, cependant la qualité de service devient primordiale de même que la détection d'intrusions.

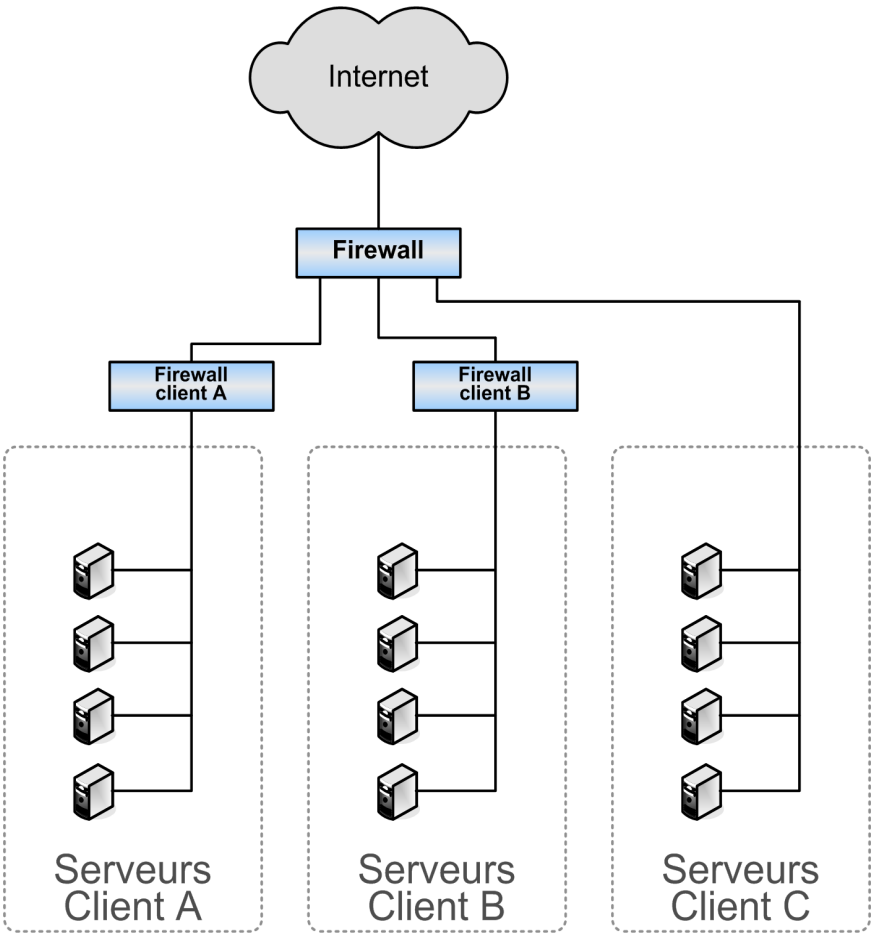
Plan du réseau

Dans cette architecture, un firewall principal géré par l'hébergeur assure l'interconnexion avec internet. Puis, suivant les clients, un firewall secondaire, géré par le client peut être intercalé. Ceci permet aux clients de gérer eux même une partie du contrôle d'accès, et de pouvoir utiliser connecter leur plateforme externe à leur réseau au moyen d'un VPN.

Politique de filtrage

La politique de filtrage est plus simple que dans un réseau de type entreprise. Il s'agit simplement de n'autoriser que les services fournis par les serveurs depuis internet, et les services d'administration depuis le réseau privé de l'hébergeur. Les serveurs ont également souvent besoin de pouvoir se connecter à internet pour leur mise à jours ou pour récupérer des données d'une autre application.

Les fonctions de QoS sont utilisées afin de garantir une certaine bande passante à chaque client. De plus un IDS ou un IPS permettent de détecter rapidement les attaques comme le déni de service, le scan de ports, etc. et de réagir immédiatement dans le cas d'un IPS.



En plus de son rôle de filtre, le firewall a également un rôle de monitoring, comme il reçoit tout le trafic, il peut comptabiliser le trafic de chaque client pour facturation.

Implémentation

Voici le jeu de règles pf du firewall principal correspondant à cette architecture :

```
##############################
## Politique par defaut      ##
block all

############################


## Classification du trafic ##
# Clients disposant de leur firewall
pass in on $internet from any to $client1 tag  IN_CLIENT1
pass in on $serveurs from $client1 to any tag OUT_CLIENT1
pass in on $internet from any to $client2 tag  IN_CLIENT2
pass in on $serveurs from $client2 to any tag OUT_CLIENT2

# Clients protégés par ce firewall
pass in on $internet proto tcp from any to $client3 \
       port $client3_services_tcp tag IN_CLIENT3
#(etc. une regle de classification par flux entrant ou sortant)

# Administration
pass in on $intranet proto tcp from $administrateurs to any \
       port $admin_services_tcp tag IN_ADMIN

# Flux à risques
pass in on $internet proto tcp from any to $client3 \
       port  $admin_services_tcp tag IN_DANGEROUS
#(etc.)

###################################
## Politique de filtrage et de QoS ##
# Trafic entrant
pass out on $serveurs tagged IN_CLIENT1 queue qos_in_client1
pass out on $serveurs tagged IN_CLIENT2 queue qos_in_client2
pass out on $serveurs tagged IN_CLIENT3 queue qos_in_client3
pass out on $serveurs tagged IN_ADMIN   queue qos_in_admin
block out log on $serveurs tagged IN_DANGEROUS

# Trafic sortant
pass out on $internet tagged OUT_CLIENT1 queue qos_out_client1
pass out on $internet tagged OUT_CLIENT2 queue qos_out_client2
pass out on $internet tagged OUT_CLIENT3 queue qos_out_client3
block out log on $internet tagged OUT_DANGEROUS
```

Particularités

Dans ce jeu de règle, on utilise une « politique de filtrage » basée sur les tags, qui à été déjà présentée dans la partie sur *pf*, ce type de jeu de règle est très bien adapté à un hébergeur qui doit pouvoir agir rapidement et globalement sur sa politique de filtrage sans devoir retoucher chaque règle une par une, tout en étant capable de gérer des exceptions ponctuelles.

Conclusion

En matière de sécurité, l'open source présente des atouts essentiels:

- La robustesse associée à des centaines de milliers de déploiements opérationnels
- L'ouverture du code au « peer review », qui garantit l'absence de « back-door » dans des composants critiques
- Le moindre coût de déploiement, qui permet de ne jamais avoir de compromis à faire.

C'est pourquoi dans ce domaine plus encore, déployer des solutions propriétaires n'a pas de sens. Certains boîtiers prêts à l'emploi, en mode « appliance », peuvent être des alternatives gages de simplicité. Mais, si c'est la simplicité qui est visée, alors des distributions dédiées, déployées sur un petit serveur, offriront un meilleur rapport qualité/prix.

Et pour des infrastructures haut de gamme, les meilleurs outils de firewalling open source permettront de combiner flexibilité et maîtrise.

SYSTEME ET INFRASTRUCTURE

Firewalls et contrôle des flux

Principes, mise en oeuvre et outils open source

Maxime **BESSON**
Expert technique

Smile
OPEN SOURCE SOLUTIONS

www.smile.fr • +33 (0)1 41 40 11 00 • contact@smile.fr
www.smile-oss.com • blog.smile.fr • twitter: @GroupeSmile