Le HOWTO du pare-feu et des serveurs mandataires

Mark Grennan mark@grenna.com v0.67, 26 septembre 1999 - Adaptation française Bernard Choppy, choppy@imaginet.fr 7 décembre 1999

Ce document est destiné à enseigner les bases des systèmes pare-feux ainsi que pour donner quelques détails sur la configuration d'un pare-feu aussi bien filtrant que serveur mandataire avec un PC sous Linux. Une version HTML de la version originale en anglais de ce document est disponible à *http://www.grennan.com/Firewall-HOWTO.html*.

Table des matières

1 Introduction

David Rudder *drig@execpc.com*. est l'auteur de la version d'origine de ce Firewall-HOWTO, il y a de si nombreux mois, et je voudrais encore le remercier pour m'avoir autorisé à mettre son travail à jour.

Les pare-feux sont devenus très populaires en tant que “nec plus ultra” de la sécurité sur Internet. Comme de nombreuses choses dont la renommée grandit, une certaine incompréhension s’y est jointe. Ce HOWTO présente les bases de la définition d’un pare-feu et la manière d’en configurer un.

J’utilise un noyau 2.2.15 et RedHat 6.0 pour développer ce howto, ainsi les exemples ci-dessous sont fondés sur cette distribution. Si vous trouvez des différences dans votre distribution, envoyez-moi un courriel et je mettrai à jour ce howto.

NdT : Diverses traductions ont été proposées pour le terme *firewall*, dont pare-feu, coupe-feu, mur anti-feu, etc. Le traducteur a adopté “pare-feu”, qui semble actuellement le terme le plus couramment admis.

1.1 Réactions

Toute réaction est la bienvenue. **SIGNALEZ TOUTE INEXACTITUDE DANS CET ARTICLE S’IL VOUS PLAIT !** Je suis humain, et donc sujet aux erreurs. Si vous trouvez une correction, envoyez-la moi. Je tenterai de répondre à tout courriel, mais je suis très occupé, donc ne m’en veuillez pas si je ne le fais pas.

Mon adresse courriel est : mark@grennan.com.

1.2 Avertissement

JE NE SUIS RESPONSABLE D’AUCUN DOMMAGE RESULTANT D’ACTIONS FONDEES SUR LE PRESENT DOCUMENT. Ce document est conçu comme une introduction au fonctionnement des pare-feux et des serveurs mandataires. Je ne suis, ni ne prétends être un expert ès sécurité. ;-) Je suis simplement un individu qui a trop lu et qui apprécie les ordinateurs plus que ce n’est le cas pour beaucoup. Considérez que j’écris ceci pour familiariser les gens avec ce sujet, et que je ne suis pas prêt à perdre ma jeunesse dans l’exactitude de ce qui s’y trouve.

NdT : Pour sa part, le traducteur émet les mêmes réserves que celles de l’auteur.

1.3 Droit d’auteur

Si vous avez des questions, veuillez me contacter (*cf. supra*).

NdT : Le traducteur (Bernard Choppy) se met aussi à disposition, soit pour répondre directement, dans la mesure de ses faibles moyens, à toute question, soit pour transmettre et traduire, entre l’auteur et l’interlocuteur francophone. Son adresse est : *choppy@imaginet.fr*

1.4 Les raisons qui me poussent à écrire ceci

Il y a quelques années, alors que je travaillais pour l'État de l'Oklahoma en tant qu'"administrateur Internet", il me fut demandé de "mettre l'État sur Internet", sans m'allouer de budget (note : il n'existait pas de titre de ce genre à l'époque. J'étais juste le type qui faisait tout le travail). La meilleure manière de faire en sorte que cela puisse arriver était d'utiliser le plus possible de logiciel libre et de matériel de récupération. Linux et un tas de vieux 486 étaient tout ce que j'avais pour travailler.

Les pare-feux du commerce sont hors de prix et la documentation sur leur fonctionnement est considérée quasiment comme secret-défense. J'ai découvert que la création d'un pare-feu de mon cru était à peu près impossible.

Dans mon travail suivant, il m'a été demandé de mettre en place un pare-feu. Linux venait d'intégrer le code correspondant. À nouveau sans budget, j'ai commencé à monter un pare-feu avec Linux. Six mois plus tard, mon pare-feu était en place et ce document mis à jour.

1.5 Autres sources d'information

- *The Linux Networking Overview HOWTO* ;
- *The Ethernet HOWTO* ;
- *IPchains Firewalling made Easy!* ;
- *Linux Network Address Translation* ;
- *The Net-3 HOWTO* ;
- *The NET-PPP HOWTO* ;
- *TCP/IP Network Administrator's Guide aux éditions O'Reilly and Associates* ;
- *The Documentation for the TIS Firewall Toolkit* .

Ces sites web sont des sources d'informations sur le sujet de la sécurité en général.

- Secure Linux.

Il s'agit de mon propre site de sécurité sur lequel j'ai rassemblé des livres blancs, de la documentation et des programmes concourant à la sécurisation de systèmes Unix.

[D'autres URL viendront ici]

2 Comprendre les pare-feux

Un pare-feu est une structure destinée à empêcher un feu de la traverser. Dans un immeuble, il s'agit d'un mur qui divise complètement des parties de celui-ci. Dans une voiture, un pare-feu est une pièce métallique qui sépare le moteur du compartiment passagers.

Les pare-feux Internet sont conçus pour isoler votre réseau local privé des flammes de l'Internet, ou de protéger la pureté des membres de votre réseau local en leur interdisant l'accès aux tentations démoniaques de l'Internet. ;-)

Le premier pare-feu informatique était une machine Unix sans routage avec deux connexions à deux réseaux différents. Une carte réseau était connectée à Internet et l'autre au réseau privé.

Pour atteindre Internet depuis le réseau privé, il fallait se loger sur le pare-feu (Unix). Ensuite, on utilisait les ressources de ce système pour accéder à Internet. Par exemple, on pouvait utiliser X-Window pour lancer le navigateur Netscape sur le pare-feu et en avoir l'affichage sur sa station de travail. Si le navigateur tourne sur le pare-feu, il a accès aux deux réseaux.

Cette sorte de hôte à double réseau (un système à deux connexions réseau) est bien si l'on peut faire confiance à TOUS les utilisateurs. On peut configurer simplement un système Linux et y créer un compte pour tout utilisateur souhaitant un accès à Internet. Avec cette configuration, le seul ordinateur du réseau privé qui connaisse quelque chose du monde extérieur est le pare-feu proprement dit. Personne ne peut télécharger directement sur un poste de travail personnelle il faut d'abord télécharger un fichier sur le pare-feu, puis transférer celui-ci du pare-feu au poste de travail.

NOTE IMPORTANTE : 99% des intrusions commencent par l'obtention d'un accès utilisateur sur le système attaqué. Pour cette raison, je ne recommande pas ce type de pare-feu. De plus, il est aussi extrêmement limité.

2.1 Politiques de sécurité

Il ne faut pas croire qu'un pare-feu soit la panacée. Il faut *tout d'abord définir une politique de sécurité*.

Les pare-feux sont utilisés dans deux buts :

1. pour maintenir des gens (intrus, vandales...) dehors ;
2. pour maintenir des gens (employés, enfants...) dedans.

Lorsque j'ai commencé à travailler sur les pare-feux, j'ai été surpris d'apprendre que l'entreprise pour laquelle je travaillais cherchait plus à “espionner” ses propres employés qu'à maintenir les intrus hors de ses réseaux.

Au moins dans mon état (l'Oklahoma), les employeurs ont le droit de surveiller les appels téléphoniques et l'activité Internet à condition d'informer préalablement les employés de cette surveillance (NdT : la législation française est identique de ce point de vue).

Big Brother n'est pas le gouvernement. Big Brother est le Big Businness.

Ne me méjugez pas : les gens sont au travail pour travailler, non pour jouer. Et il me semble que l'éthique du travail est en train de s'éroder. Néanmoins, j'ai aussi observé que certains types de directions se trouvent aussi les principaux transgresseurs des règles qu'ils ont eux-mêmes édictées. J'ai vu des vacataires réprimandés car ils avaient utilisé Internet pour chercher le trajet de bus pour venir au travail alos que le même directeur passait des heures au travail à chercher de bons restaurants et boîtes de nuit pour y trouver des clients potentiels.

Mon correctif de ce genre d'abus est de publier les traces du pare-feu sur une page web accessible à tout un chacun.

Le travail dans la sécurité peut être difficile. Si vous êtes gestionnaire de pare-feux, surveillez vos arrières.

2.2 Types de pare-feux

Il y a deux types de pare-feux :

1. pare-feux IP ou filtrants - ils bloquent tout le trafic sauf celui sélectionné ;
2. serveurs mandataires (parfois appelés bastions) - ils réalisent les connexions réseau pour vous.

2.2.1 Pare-feux filtrants

Le filtrage de paquets est le type de pare-feu inclus dans le noyau Linux.

Un pare-feu filtrant fonctionne au niveau du réseau. Les données ne sont autorisées à quitter le système que si les règles du pare-feu le permettent. Lorsque les paquets arrivent, ils sont filtrés en fonction de leurs type, origine, destination et port qui sont décrits dans chacun de ceux-ci.

De nombreux routeurs comportent un certain nombre de services de type pare-feu. Les pare-feux filtrants peuvent être pensés comme des types particuliers de routeurs. Pour cette raison, il faut une profonde compréhension de la structure des paquets IP pour travailler avec l'un d'eux.

Puisque très peu de données sont analysées et tracées, les pare-feux filtrants consomment peu de temps processeur et créent moins de latence sur un réseau.

Les pare-feux filtrants ne fournissent pas de contrôle par mot de passe. Un utilisateur ne peut s'identifier en tant que tel. La seule identité connue pour un utilisateur est l'adresse IP de son poste de travail. Cela peut être un problème lorsqu'on souhaite utiliser DHCP (assignation dynamique d'adresses IP). En effet, les règles étant fondées sur les adresses IP, il faut ajuster celles-ci à chaque fois que de nouvelles adresses sont assignées. Je ne sais pas comment automatiser ce processus.

Les pare-feux filtrants sont plus transparents pour les utilisateurs. Ceux-ci n'ont en effet pas à configurer des règles dans leurs applications pour utiliser Internet. Ce n'est pas vrai avec la plupart des serveurs mandataires.

2.3 Serveurs mandataires

Le meilleur exemple du fonctionnement de ceux-ci est celui d'une personne se connectant à un système puis, depuis celui-ci, au reste du monde. C'est seulement avec un serveur mandataire que ce processus est automatique. Lorsque vous vous connectez à l'extérieur, le logiciel client vous connecte en fait d'abord au serveur mandataire. Le serveur mandataire se connecte alors au serveur que vous cherchez à atteindre (l'extérieur) et vous renvoie les données reçues. NdT : en français, on utilise souvent le terme “bastion” pour désigner un serveur mandataire situé entre le réseau local interne et l'extérieur. Dans le présent document, on utilisera plutôt le terme bastion pour désigner la machine qui porte le serveur mandataire.

Puisque les serveurs mandataires gèrent toutes les communications, ils peuvent enregistrer tout ce qu'ils font (donc ce que vous faites). Pour les mandataires HTTP (web), cela comprend les URL que vous demandez. Pour les mandataires FTP, cela inclut chaque fichier téléchargé. Ils peuvent même expurger les mots “inappropriés” des sites que vous visitez ou analyser la présence de virus.

Les serveurs mandataires d'applications peuvent authentifier des utilisateurs. Avant qu'une connexion soit réalisée vers l'extérieur, le serveur peut demander à l'utilisateur de se connecter préalablement. Pour un utilisateur web, cela fonctionnera comme si chaque site requérait une connexion.

2.3.1 Mandataire SOCKS

Un mandataire SOCKS ressemble beaucoup à un vieux central téléphonique à fiches. Il interconnecte simplement une machine interne à une autre externe.

De nombreux serveurs SOCKS fonctionnent uniquement avec les connexions de type TCP. De même, comme les pare-feux filtrants, il ne permettent pas l'authentification d'utilisateurs. En revanche, ils peuvent enregistrer la destination de la connexion de chaque utilisateur.

3 Architecture de pare-feu

Il existe de nombreuses manières de structurer un réseau pour protéger des systèmes à l'aide d'un pare-feu.

Si l'on dispose de connexions dédiées à Internet par un routeur, on peut connecter directement celui-ci au système pare-feu. Au contraire, on peut passer par un hub pour permettre un accès complet aux serveurs à l'extérieur du pare-feu.

On peut configurer un certain nombre de règles de filtrage matérielles dans le routeur. Néanmoins, ce routeur peut être la propriété d'un FAI (fournisseur d'accès Internet), auquel cas on ne dispose pas du contrôle de celui-ci. Il faut demander au FAI d'y inclure des filtres (NdT : et avoir pleine confiance dans son FAI !).

```
                  ---------          ----------
   _/\__/\_      | Routeur |        |          |         -----------
  |        |     | sans    |  (DMZ) | Systeme  |  (LAN)  | Postes de |
 / Internet \----| filtre  |--(HUB)--| pare-feu |--(HUB)--|  travail  |
 \_  _  _  _/    |_________|    |    |__________|         |___________|
   \/ \/ \/                     |
                          (Serveur pour)
                          (l'exterieur)
```

On peut aussi utiliser un service commuté comme une ligne RNIS. Dans ce cas on peut utiliser une troisième carte réseau pour créer une DMZ (De-Militarized Zone, ou "zone démilitarisée") filtrée. Cela donne un contrôle total sur les services Internet et maintient la séparation avec le réseau local normal.

```
                  ----------
   _/\__/\_      |          |         -----------
  |        |     | Systeme  |  (LAN)  | Postes de |
 / Internet \----| pare-feu |--(HUB)--|  travail  |
 \_  _  _  _/    |__________|         |___________|
   \/ \/ \/           |
                    (DMZ)
                    (HUB)
```

Si l'on ne fournit pas soi-même des services Internet mais que l'on souhaite surveiller où vont les utilisateurs, on voudra utiliser un serveur mandataire (bastion). Cela peut être intrégré dans le pare-feu.

```
                  ----------
   _/\__/\_      | Systeme  |         -----------
  |        |     | pare-feu/|  (LAN)  | Postes de |
 / Internet \----| bastion  |--(HUB)--|  travail  |
 \_  _  _  _/    |__________|         |___________|
   \/ \/ \/
```

On peut aussi placer le serveur mandataire sur le réseau local. Dans ce cas, les règles du pare-feu ne doivent autoriser que le bastion à se connecter à Internet pour les services que celui-ci fournit. Ainsi les utilisateurs ne peuvent accéder à Internet que par le mandataire.

```
                  ----------
   _/\__/\_      |          |         -----------
  |        |     | Systeme  |  (LAN)  | Postes de |
 / Internet \----| pare-feu |--(HUB)--|  travail  |
 \_  _  _  _/    |__________|    |    |___________|
   \/ \/ \/                      |     ------------
                                 |    |  Serveur   |
                                 +----| mandataire |
                                      |____________|
```

Si l'on souhaite réaliser un service comme ceux de Yahoo! ou peut-être SlashDot, on peut souhaiter réaliser une architecture redondante de routeurs et pare-feux (*cf.* High Availability HOWTO).

En utilisant une technique de DNS à jeton tournant ou à l'aide de serveurs d'application à équilibrage de charge, on peut créer un service à 100% de disponibilité.

```
  _/\__/\_                                      _/\__/\__/\
 |        |                                    |           |
/ FAI No 1 \______                  (WAN)_____/ Partenaires \
\_  _  _  _/      |                 (HUB)     \_  _  _  _  _/
  \/ \/ \/        |                ___|____     \/ \/ \/ \/
                __|____           |_______ |
  _/\__/\_     |______ |         |        ||          ______
 |        |   |       ||  (DMZ)  |Systeme ||  (LAN)  |      |
/ FAI No 2 \--|Routeur||--(HUB)--|pare-feu||--(HUB)--|Postes|
\_  _  _  _/  |_______|     |    |________|     |    |______|
  \/ \/ \/                  |         |         |     ______
                        (Serveur)  (Serveur)    |    |Manda-|
                       (exterieur) (commun)     +----|taire |
                                                     |______|
```

Il est facile de voir corrompre son réseau local. Il faut conserver le contrôle de chaque connexion. Il suffit d'un utilisateur avec un modem pour compromettre tout un réseau local.

4 Configurer le pare-feu filtrant Linux

4.1 Matériel nécessaire

Les pare-feux filtrants ne nécessitent pas de matériel haut de gamme. Ils ne sont pas grand'chose de plus que de simples routeurs.

Tout ce qu'il faut est :

1. un 486-DX66 avec 16 Mo de mémoire vive ;
2. un disque dur de 200 Mo (500 Mo sont tout de même recommandés) ;
3. des connexions réseau (cartes Ethernet, ports série, connexions sans fil ?) ;
4. un moniteur et un clavier.

Avec certains systèmes, on peut même éliminer le moniteur et le clavier en utilisant une console sur port série.

Si l'on a besoin d'un serveur mandataire qui doive gérer un fort trafic, il faut prendre le système le plus puissant qui soit possible. En effet, pour chaque utilisateur qui se connectera par le bastion, il se créera un nouveau processus. Si l'on a 50 utilisateurs simultanés ou plus, je pense qu'il faut :

1. un Pentium II avec 64 Mo de mémoire vive ;
2. un disque dur de 2 Go pour contenir toutes les traces ;
3. deux connexions réseau ;
4. un moniteur et un clavier.

Les connexions réseau peuvent être de n'importe quel type (cartes NIC, RNIS ou même des modems).

4.2 Logiciel nécessaire

4.2.1 Pare-feu filtrant

Pour créer un pare-feu filtrant, aucun logiciel spécifique n'est nécessaire. Linux suffit.

Si l'on utilise un _TRES_ vieux noyau Linux (1.0.x ou plus ancien), il faut une copie de ipfwadm (NdT : il faut surtout passer à un noyau plus récent, car ces noyaux sont obsolètes, et risquent de plus de contenir des points faibles dont il sera difficile de retrouver trace).

Si l'on utilise un noyau 2.1.102 ou plus récent, il faut utiliser un ipchaining comme par exemple celui développé par *http://www.rustcorp.com/linux/ipchains/*.

4.2.2 Serveur mandataire

Si l'on veut configurer un serveur mandataire, il faut l'un des paquetages suivants :

- Squid ;
- la boîte à outils TIS Firewall (FWTK) ;
- SOCKS.

Squid est un beau paquetage et fonctionne avec la fonctionnalité de Linux de mandataire transparent. Je vais décrire comment configurer ce serveur.

À l'heure où j'écris ces lignes, *Network Associates* et Trusted Information System's (TIS) ont fusionné. Il faut donc continuer à consulter leurs sites web pour toute information sur les modifications. Dans l'intervalle, la boîte à outils TIS est toujours disponible sur : *http://www.tis.com/research/software/*.

Trusted Information System fournit une collection de programmes conçue pour faciliter la gestion de pare-feux. Avec cette boîte à outils, on configure un daemon pour chaque service (web, telnet, etc.) qui sera utilisé.

5 Préparer le système Linux

5.1 Compiler le noyau

Commencez avec une installation minimale propre de votre distribution Linux. Moins vous installez de logiciels, moins votre système aura de trous de sécurité, portes dérobées et/ou bogues susceptibles d'induire des problèmes de sécurité dans votre système.

Prenez un noyau stable. J'utilise le noyau Linux 2.2.9 ou plus pour mon système. La documentation est donc fondée sur ces paramètres.

Vous devez recompiler le noyau Linux avec les options appropriées. Si vous n'avez jamais recompilé de noyau, je vous renvoie au Kernel HOWTO, à l'Ethernet HOWTO et au NET-3 HOWTO.

Voici les paramètres réseau que je sais correspondre à quelque chose qui fonctionne. J'en ai signalé quelques-uns avec un “?”. Si l'on souhaite utiliser cette fonctionnalité, il faut aussi l'activer.

J'utilise “make menuconfig” pour éditer ma configuration de noyau.

```
<*> Packet socket
[ ] Kernel/User netlink socket
[*] Network firewalls
```

```
[ ] Socket Filtering
<*> Unix domain sockets
[*] TCP/IP networking
[ ] IP: multicasting
[*] IP: advanced router
[ ] IP: kernel level autoconfiguration
[*] IP: firewalling
[?] IP: always defragment (required for masquerading)
[?] IP: transparent proxy support
[?] IP: masquerading
--- Protocol-specific masquerading support will be built as modules.
[?] IP: ICMP masquerading
--- Protocol-specific masquerading support will be built as modules.
[ ] IP: masquerading special modules support
[*] IP: optimize as router not host
< > IP: tunneling
< > IP: GRE tunnels over IP
[?] IP: aliasing support
[*] IP: TCP syncookie support (not enabled per default)
--- (it is safe to leave these untouched)
< > IP: Reverse ARP
[*] IP: Allow large windows (not recommended if <16Mb of memory)
< > The IPv6 protocol (EXPERIMENTAL)
---
< > The IPX protocol
< > Appletalk DDP
< > CCITT X.25 Packet Layer (EXPERIMENTAL)
< > LAPB Data Link Driver (EXPERIMENTAL)
[ ] Bridging (EXPERIMENTAL)
[ ] 802.2 LLC (EXPERIMENTAL)
< > Acorn Econet/AUN protocols (EXPERIMENTAL)
< > WAN router
[ ] Fast switching (read help!)
[ ] Forwarding between high speed interfaces
[ ] PU is too slow to handle full bandwidth
QoS and/or fair queueing  --->
```

Après avoir réalisé toute la configuration qu'il vous faut vous devez recompiler, réinstaller le noyau et rebouter.

J'utilise la commande :

```
make dep;make clean;make bzlilo;make modules;make modules_install;init 6
```

pour accomplir tout cela en une étape.

5.2 Configurer deux cartes réseau

Si vous avez deux cartes réseau dans votre ordinateur, vous devrez très certainement ajouter un paramètre “append” dans votre fichier /etc/lilo.conf pour décrire les IRQ et adresses des deux cartes. Le mien se présente ainsi :

```
append="ether=12,0x300,eth0 ether=15,0x340,eth1"
```

5.3 Configurer les adresses réseau

Nous arrivons à la partie amusante de notre configuration. Je ne vais pas entrer très profondément dans les détails de mise en place d'un réseau local. Pour résoudre vos problèmes à ce niveau, vous pouvez vous reporter au Networking-HOWTO.

Votre but est de fournir deux connexions réseau à votre système pare-feu filtrant : l'une est Internet (côté dangereux) et l'autre est le réseau local (côté sécurisé).

Dans tous les cas, vous devez prendre quelques décisions :

1. utiliserez-vous des adresses IP réelles ou non pour votre réseau local ?
2. votre FAI vous assigne-t'il une adresse IP ou utilisez-vous des adresses IP statiques ?

Puisque nous ne souhaitons pas laisser InterNet accéder au réseau privé, il n'est pas nécessaire d'utiliser des adresses "réelles". On peut toujours choisir des adresse arbitraires pour un réeau privé, mais ce n'est pas recommandé : au cas où des données seraient routées en-dehors de celui-ci, elles pourraient arriver sur un autre système.

Un certain nombre de plages d'adresses Internet ont été laissées de côté pour les réseaux privés. Parmi celles-ci, nous utiliserons celles de 192.168.2.xxx pour nos exemples.

Il vous faudra utiliser le masquage IP (IP masquerading) pour permettre à cela de fonctionner. Dans ce mode de fonctionnement, le pare-feu transmet les paquets en les traduisant en adresses IP "réelles" pour leur permettre de voyager sur Internet.

L'utilisation de ces adresses IP non routables rend le réseau privé plus sûr, car les routeurs Internet ne transmettront pas les paquets porteurs de ces adresses.

Il peut être judicieux maintenant de lire le IP Masquerading HOWTO.

```
           199.1.2.10   __________    192.168.2.1
  _  __  _          \ |          | /           ___________
 | \/  \/ |          \| Systeme  |/           | Postes de |
/ InterNet \--------| pare-feu |------------|  travail  |
\_/\_/\_/\_/        |__________|            |___________|
```

Vous devez assigner l'adresse IP "réelle" à la carte réseau du côté Internet. Cette adresse peut vous être assignée de manière permanente (comme adresse IP statique) ou à chaque connexion par le processus PPP.

Vous assignez vos numéros IP internes, comme 192.168.2.1 pour la carte Ethernet du réeau local. Il s'agira de votre adresse de passerelle. Vous pouvez assigner une adresse de la plage 192.168.2.xxx à toutes les autres machines du réseau protégé (soit 192.168.2.2 à 192.168.2.254).

J'utilise Linux RedHat. Pour configurer le réseau lors du démarrage, j'ai ajouté un fichier "ifcfg-eth1" dans le répertoire /etc/sysconfig/network-scripts. On peut aussi trouver des fichiers ifcfg-ppp0 ou ifcfg-tr0 dans ce répertoire. Ces fichiers "ifcfg-" sont utilisés par RedHat pour configurer et activer les périphériques réseau lors du démarrage. Leur nom est fonction du type de connexion.

Voici l'allure du fichier ifcfg-eth1 (deuxième carte Ethernet) de notre exemple :

```
DEVICE=eth1
```

```
IPADDR=192.168.2.1
NETMASK=255.255.255.0
NETWORK=192.168.2.0
BROADCAST=192.168.2.255
GATEWAY=199.1.2.10
ONBOOT=yes
```

Si vous utilisez une connexion commutée, vous devrez consulter les fichiers ifcfg-ppp0 et chat-ppp0 qui contrôlent votre connexion PPP.

Ce fichier ifcfg peut avoir l'allure suivante :

```
DEVICE="ppp0"
ONBOOT="yes"
USERCTL="no"
MODEMPORT="dev/modem"
LINESPEED="115200"
PERSIST="yes"
DEFABORT="yes"
DEBUG="yes"
INITSTRING="ATZ"
DEFROUTE="yes"
HARDFLOWCTL="yes"
ESCAPECHARS="no"
PPPOPTIONS=""
PAPNAME="LoginID"
REMIP=""
NETMASK=""
IPADDR=""
MRU=""
MTU=""
DISCONNECTTIMEOUT=""
RETRYTIMEOUT="5"
BOOTPROTO="none"
```

5.4 Tester votre réseau

Commencer en utilisant les commandes ifconfig et route. Si vous avez deux cartes réseau, votre ifconfig doit ressembler à :

```
#ifconfig
lo      Link encap:Local Loopback
        inet addr:127.0.0.0  Bcast:127.255.255.255  Mask:255.0.0.0
        UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
        RX packets:1620 errors:0 dropped:0 overruns:0
        TX packets:1620 errors:0 dropped:0 overruns:0

eth0    Link encap:10Mbps Ethernet  HWaddr 00:00:09:85:AC:55
        inet addr:199.1.2.10 Bcast:199.1.2.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0
```

```
          TX packets:0 errors:0 dropped:0 overruns:0
          Interrupt:12 Base address:0x310

eth1      Link encap:10Mbps Ethernet  HWaddr 00:00:09:80:1E:D7
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
          Interrupt:15 Base address:0x350
```

et votre table de routage :

```
#route -n
Kernel routing table
Destination  Gateway      Genmask         Flags MSS   Window Use Iface
199.1.2.0    *            255.255.255.0   U     1500  0       15 eth0
192.168.2.0  *            255.255.255.0   U     1500  0        0 eth1
127.0.0.0    *            255.0.0.0       U     3584  0        2 lo
default      199.1.2.10   *               UG    1500  0       72 eth0
```

Note : 199.1.2.0 est l'adresse du côté Internet du pare-feu et 192.168.2.0 celle du côté privé (réseau local).

Vous devez commencer par vous assurer que chaque ordinateur de votre réseau local puisse atteindre l'adresse de votre pare-feu (192.168.2.2 dans cet exemple) par ping. Dans le cas contraire, replongez-vous dans le NET-3 HOWTO et retravaillez un peu votre réseau.

Ensuite, essayez d'atteindre un système Internet depuis le pare-feu. J'utilise personnellement l'adresse *www.internic.net* pour mes tests. Si cela ne fonctionne pas, essayez un serveur de votre FAI. Si cela ne fonctionne toujours pas, il y a un problème quelque part dans la configuration de votre connexion Internet. Vous devriez pouvoir atteindre n'importe quel point d'Internet depuis votre pare-feu. Essayez de vérifier la configuration de votre passerelle par défaut. Si vous utilisez une connexion commutée, recontrôlez votre nom d'utilisateur et votre mot de passe. Replongez–vous dans le NET-3 HOWTO et essayez encore.

Maintenant, essayez d'atteindre l'adresse extérieure du pare-feu (199.1.2.10 dans notre exemple) depuis une machine du réseau local. Cela ne doit pas fonctionner. Dans le cas contraire, le masquage IP ou la transmission IP sont activés ou vous avez déjà une sorte de filtrage de paquet positionnée. Désactivez-les et réessayez. Vous devez savoir si le filtrage est en place.

Pour les noyaux postérieurs au 2.1.102, vous pouvez utiliser la commande :

```
echo "0" > /proc/sys/net/ipv4/ip_forward
```

Si, en revanche, vous utilisez un noyau plus ancien (pourquoi ?), vous devrez recompiler celui-ci en désactivant la transmission IP (IP forwarding, mais mettez plutôt votre noyau à jour).

Essayez à nouveau d'atteindre l'adresse extérieure du pare-feu (199.1.2.10 dans notre exemple) depuis une machine du réseau local. Cela ne doit pas fonctionner.

Maintenant, activez la transmission IP et/ou le masquage IP. Vous devez être à même d'atteindre n'importe quel point d'Internet depuis n'importe quelle machine de votre réseau local.

```
echo "0" > /proc/sys/net/ipv4/ip_forward
```

Note importante : Si vous utilisez des adresses IP “réelles” sur votre réseau local (et non 192.168.xxx.yyy) et que vous ne puissisez atteindre Internet, mais que vous puissiez atteindre le côté extérieur de votre pare-feu, vérifiez que votre FAI route correctement les paquets depuis l’adresse de votre réseau privé.

Un test de ce problème est de connaître quelqu’un d’autre sur Internet (un ami utilisant un fournisseur local, par exemple) et de lui demander d’effectuer un traceroute vers votre réseau. Si la trace s’arrête sur le routeur de votre fournisseur, c’est qu’il ne transmet pas votre trafic.

Ça fonctionne ? Bien. La partie la plus difficile est faite. :-)

5.5 Sécuriser le pare-feu

Le pare-feu n’est d’aucune utilité s’il reste largement ouvert aux attaques. Un “méchant” pourrait obtenir l’accès au pare-feu et le modifier pour ses desseins personnels. Vous devez désactiver tous les services inutilisés.

Regardez dans votre fichier /etc/inetd.conf. Ce fichier contrôle inetd qu’on appelle aussi “super-serveur”. Il contrôle un tas de daemons serveurs et les exécute à la demande, à partir des paquets qui arrivent sur un port “bien connu” (well known port).

Vous devez désactiver echo, discard, daytime, chargen, ftp, gopher, shell, login, exec, talk, ntalk, pop-2, pop-3, netstat, systat, tftp, bootp, finger, cfinger, time, swat ainsi que linuxconfig si vous en possédez un.

Pour désactiver un service, placez simplement un “#” (dièze) devant. Ensuite, envoyez un signal SIG-HUP au processus inetd, selon la syntaxe suivante :

```
kill -HUP <pid>
```

où “pid” est le numéro du processus inetd. Cela force inetd à relire son fichier de configuration (inetd.conf) et à se relancer sans arrêter votre système.

Testez le résultat par telnet sur le port 15 (netstat) du pare-feu. Si vous obtenez une réponse de netstat, c’est que vous n’avez pas arrêté ces services correctement.

```
telnet localhost 15
```

Vous pouvez aussi créer le fichier /etc/nologin. Placez-y quelques mots (comme “CONNEXION STOPPEE”). Lorsque ce fichier existe, login n’autorise pas de connexion d’utilisateur. Ceux-ci verront simplement le contenu de ce fichier et leur connexion sera refusée. Seul root peut alors se loger.

Vous pouvez encore éditer le fichier /etc/securetty. Si l’utilisateur est root, la connexion doit s’effectuer depuis l’un des périphériques indiqués dans /etc/securetty. Les échecs seront tracés par la facilité syslog. Avec ces deux contrôles actifs, la seule manière de se connecter sur le pare-feu est de se présenter comme root sur la console.

6 Configuration du filtrage IP (IPFWADM)

Si vous utilisez un noyau 2.1.102 ou supérieur, passez directement à la section suivante sur IPCHAINS.

Dans les anciens noyaux, la transmission IP est activée par défaut. Pour cette raison, votre réseau doit commencer par refuser l’accès à tout et vider toutes les règles de transmission (ipfw) en place depuis le dernier lancement. Le fragment de script suivant doit se trouver dans votre script de lancement réseau (en général, /etc/rc.d/init.d/network) :

```
#
```

```
# configuration de la transmission IP et
# de la trace
#
#   Transmission
#
# Par defaut INTERDIRE tous les services
ipfwadm -F -p deny
# Vider toutes les regles de trace
ipfwadm -F -f
ipfwadm -I -f
ipfwadm -O -f
```

Maintenant, nous avons un pare-feu absolu. Rien ne peut passer au travers.

Maintenant, on peut créer le fichier /etc/rc.d/rc.firewall. Ce script doit autoriser le passage des trafics courrier, web et DNS. ;-)

```
#!/bin/sh
#
# rc.firewall
#
# Lancement de la bibliotheque de fonctions
. /etc/rc.d/init.d/functions

# Lecture de la configuration
. /etc/sysconfig/network

# On controle que le reseau soit actif
if [ $NETWORKING = "no" ]
then
        exit 0
fi
case "$1" in
  start)
  echo -n "Lancement des services du pare-feu : "
  # Autorise le courriel a arriver au serveur
  /sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.1.2.10 25
  # Autorise les connexions aux serveurs courriel externes
  /sbin/ipfwadm -F -a accept -b -P tcp -S 192.1.2.10 25 -D 0.0.0.0/0 1024:65535
  # Autorise les connexions a notre serveur web
  /sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.1.2.11 80
  # Autorise les connexions aux serveurs web externes
  /sbin/ipfwadm -F -a accept -b -P tcp -S 192.1.2.* 80 -D 0.0.0.0/0 1024:65535
  # Autorise le trafic DNS
  /sbin/ipfwadm -F -a accept -b -P udp -S 0.0.0.0/0 53 -D 192.1.2.0/24
  ;;
  stop)
  echo -n "Arret des services du pare-feu : "
  ipfwadm -F -p deny
  ;;
  status)
  echo -n "Montrez-vous les statistiques du pare-feu ?"
  ;;
  restart|reload)
        $0 stop
```

```
            $0 start
            ;;
    *)
            echo "Usage: firewall {start|stop|status|restart|reload}"
            exit 1
esac
```

Note : Dans cet exemple, nous avons un serveur courriel (SMTP) qui tourne sur 192.1.2.10 qui doit être capable d'envoyer et recevoir des paquets sur le port 25. Le serveur web tourne sur 192.1.2.11. Nous autorisons quiconque sur le réseau local à accéder aux serveurs web externes ainsi qu'aux serveurs DNS.

Cet exemple n'est pas parfaitement sécurisé. Puisque le port 80 n'est pas obligatoirement utilisé comme port web, un intrus rusé pourrait utiliser celui-ci pour créer un réseau privé virtuel (VPN : virtual private network) au-travers du pare-feu. Pour contourner cela, il faut configurer un mandataire web, et n'autoriser que celui-ci à traverser le pare-feu. Les utilisateurs du réseau local devront alors passer par le mandataire pour atteindre les serveurs web extérieurs.

On peut aussi s'intéresser à la surveillance du trafic qui passe au-travers du pare-feu. Le script qui suit compte chaque paquet. On peut ajouter une ou deux lignes pour compter les paquets qui vont vers un système particulier.

```
# Flush the current accounting rules
ipfwadm -A -f
# Accounting
/sbin/ipfwadm -A -f
/sbin/ipfwadm -A out -i -S 192.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A out -i -S 0.0.0.0/0 -D 192.1.2.0/24
/sbin/ipfwadm -A in -i -S 192.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A in -i -S 0.0.0.0/0 -D 192.1.2.0/24
```

Si tout ce que vous cherchez est un pare-feu filtrant, vous pouvez vous arrêter ici. Testez-le et amusez-vous bien :-)

7 Configuration du filtrage IP (IPchains)

Les ipchains Linux sont une réécriture du code pare-feu IPv4 de Linux (en grande partie inspiré de BSD) ainsi que de ipfwadm qui était lui-même une réécriture du ipfw de BSD, je crois. Il est indispensable d'administrer les filtres de paquets IP dans les noyaux Linux versions 2.1.102 et au-delà.

L'ancien code ne gère pas les fragments, a des compteurs sur 32 bits (sur Intel en tout cas), ne permet aucune autre spécification de protocole que TCP, UDP ou ICMP, ne peut réaliser de vastes modifications de manière atomique, ne peut spécifier de règles inversées, a quelques défauts et peut se révéler dur à administrer (ce qui le rend vulnérable aux erreurs d'utilisation). Ou du moins c'est ce qu'en dit l'auteur.

Je ne vais pas entrer en détail sur la manière de contrôler un pare-feu IPchains, car il existe un _BON !!_ HOWTO qui en parle sur *http://www.rustcorp.com/linux/ipchains/HOWTO.html*. Je finirais juste par le recopier ici. En voici les bases.

On travaille avec les chaînes par noms. Il existe au début trois chaînes prédéfinies input, output et forward, qu'il est impossible de supprimer. On peut créer des chaînes personnelles. Des règles peuvent ensuite être ajoutées et supprimées de ces ensembles de règles.

Les opérations nécessaires pour travailler sur les chaînes sont les suivantes :

1. création d'une nouvelle chaîne (-N) ;

2. suppression d'une chaîne vide (-X) ;
3. modification de la spécification pour une chaîne prédéfinie (-P)
4. liste des règles d'une chaîne (-F) ;
5. remise à zéro des compteurs de paquets et d'octets de toutes les règles d'une chaîne (-Z).

Il existe différentes méthodes pour manipuler les règles à l'intérieur des chaînes :

1. ajout d'une nouvelle règle dans une chaîne (-A) ;
2. insertion d'une nouvelle règle à une position donnée d'une chaîne (-I) ;
3. remplacement d'une règle à une position donnée d'une chaîne (-R) ;
4. suppression d'une règle à une position donnée d'une chaîne (-D) ;
5. suppression de la première règle correspondante dans une chaîne (-D).

Il existe quelques opérations pour le masquage qui se trouvent dans ipchains dans l'attente d'un bon emplacement pour les placer :

1. liste des connexions actuellement masquées (-M -L) ;
2. positionnement du temps d'expiration du masquage (-M -S).

Il y a quelques détails de chronologie qui interviennent dans la modification des règles de pare-feu. Si l'on n'est pas suffisamment prudent, il est possible de laisser passer quelques paquets pendant la mise en place des modifications. Une approche simpliste est la suivante :

```
# ipchains -I input 1 -j DENY
# ipchains -I output 1 -j DENY
# ipchains -I forward 1 -j DENY
```

... réalisation des modification ...

```
# ipchains -D input 1
# ipchains -D output 1
# ipchains -D forward 1
#
```

Cela interdit tout passage de paquet durant les modifications.

Voici une copie des règles de pare-feu qui précèdent dans IPChains :

```
#!/bin/sh
#
# rc.firewall
#
## Tout vider et repartir du debut
  /sbin/ipchains -F input
  /sbin/ipchains -F output
  /sbin/ipchains -F forward

## Redirection pour le mandatement transparent de HTTP
  #$IPCHAINS  -A input -p tcp -s 192.1.2.0/24 -d 0/0 80 -j REDIRECT 8080

## Creation de nos propres chaines
```

```
  /sbin/ipchains -N ma-chaine
  # On autorise le courriel entrant vers le serveur
  /sbin/ipchains -A ma-chaine -s 0.0.0.0/0 smtp -d 192.1.2.10 1024:-j ACCEPT
  # On autorise les connexions courriel vers l'extérieur
  /sbin/ipchains -A ma-chaine -s 192.1.2.10 -d 0.0.0.0/0 smtp -j ACCEPT
  # On autorise les connexions web entrantes vers le serveur
  /sbin/ipchains -A ma-chaine -s 0.0.0.0/0 www -d 192.1.2.11 1024: -j ACCEPT
  # On autorise les connexions web vers l'extérieur
  /sbin/ipchains -A ma-chaine -s 192.1.2.0/24 1024: -d 0.0.0.0/0 www -j ACCEPT
  # On autorise le trafic DNS
  /sbin/ipchains -A ma-chaine -p UDP -s 0.0.0.0/0 dns -d 192.1.2.0/24 -j ACCEPT

## Si on utilise le masquage
  # On ne masque pas le trafic interne
  /sbin/ipchains -A forward -s 192.1.2.0/24 -d 192.1.2.0/24 -j ACCEPT
  # On ne masque pas l'interface externe directe
  /sbin/ipchains -A forward -s 199.1.2.0/24 -d 0/0 -j ACCEPT
  # On masque tout paquet interne qui sort
  /sbin/ipchains -A forward -s 192.1.2.0/24 -d 0/0 -j MASQ

## On interdit tout le reste
  /sbin/ipchains -P my-chains input DENY
```

Il ne faut pas s'arrêter là. Ce n'est pas un pare-feu très puissant et je suis sûr qu'il y a d'autres services que vous souhaiteriez fournir. À nouveau, lisez le IPCHAINS-HOWTO.

8 Rendre le tout plus simple

Il existe des interfaces graphiques ou fondées sur le web développées pour fonctionner avec les règles de filtrage Linux. Un certain nombre d'entreprises ont même créé des pare-feux commerciaux fondés sur Linux en le plaçant dans leur propre machine avec leur propre code de management (joli).

gfcc (GTK+ Firewall Control Center) est une application GTK+ capable de contrôler les principes et règles de filtrage de Linux, fondée sur le paquetage ipchains. Rendez-vous sur *http://megaman.ypsilonia.net/kfirewall/*.

FCT est un outil HTML de configuration de pare-feu. Il permet la génération automatique de scripts de commandes de filtrage IP (ipfwadm) sur un pare-feu pour des interfaces multiples et tout service internet *http://www.fen.baynet.de/˜ft114/FCT/firewall.htm*

9 Installer un mandataire Squid transparent

Le mandataire squid est disponible sur *http://squid.nlanr.net/*.

Les développeurs de squid fournissent des paquetages aux formats RedHat et Debian. Si vous le pouvez, il vaut mieux utiliser l'un d'entre eux.

10 Installer le serveur mandataire TIS

10.1 Trouver le logiciel

Le TIS fwtk est disponible à *http://www.tis.com/research/software/*.

Ne commettez pas l'erreur que j'ai commise : lorsque vous téléchargez les fichiers de TIS, LISEZ LES README. Le TIS fwtk est verrouillé dans un répertoire caché sur leur serveur.

TIS impose que vous lisiez leur accord à

http://www.tis.com/research/software/fwtk_readme.html, puis que vous **envoyiez un courriel à *fwtk-request@tislabs.com*** avec le seul mot **accepted** dans le corps du message pour connaître le nom de ce répertoire caché. Aucun sujet n'est nécessaire pour ce message. Leur système vous enverra en retour le nom du répertoire par courriel (valaple 12 heures) pour charger le source.

A l'instant où j'écris, la version à jour du FWTK est 2.1.

10.2 Compilation du FWTK TIS

La version 2.1 du FWTK se compile beaucoup plus facilement que les précédentes.

EXPLIQUER ICI !!!

Maintenant, lancez **make**.

10.3 Installation du FWTK TIS

Lancez **make install**.

Le répertoire d'installation par défaut est /usr/local/etc. Il est possible de changer cela (ce que je n'ai pas fait) vers un répertoire plus sûr. J'ai choisi de changer l'accès à ce répertoire pour le mode “0700”.

Tout ce qu'il reste à faire maintenant est de configurer le pare-feu.

10.4 Configuration du FWTK TIS

Maintenant, le plaisir commence vraiment. Nous devons enseigner au système à appeler ces nouveaux services et créer les tables pour les contrôler.

Je ne suis pas en train de ré-écrire le manuel de TIS fwtk ici. Je vais montrer les paramètres que j'ai fait fonctionner et expliquer les problèmes que j'ai rencontrés et comment je les ai contournés.

Trois fichiers définissent ces contrôles :

- /etc/services : indique au système sur quel port se trouve un service ;
- /etc/inetd.conf : indique à inetd le programme à lancer lorsque quelqu'un appelle un port de service ;
- /usr/local/etc/netperm-table : indique aux services fwtk à qui autoriser ou interdire l'accès au service.

Pour faire fonctionner fwtk, vous devez éditer ces fichiers de bas en haut. Editer le fichier des services sans que les fichiers inetd.conf ou netperm-table soient corrects peut rendre votre système inaccessible.

10.4.1 Le fichier netperm-table

Ce fichier contrôle qui a accès aux services de TIS FWTK. Vous devez penser au trafic qui passe par le pare-feu depuis les deux côtés. Les gens de l'extérieur de votre réseau doivent s'identifier avant d'obtenir l'accès, mais ceux de l'intérieur doivent être autorisés simplement à passer au-travers.

Afin que les gens puissent s'identifier, le pare-feu utilise un programme appelé **authsrv** pour maintenir une base des noms et mots de passe. La section authentification de netperm-table contrôle l'emplacement et l'accès à la base.

J'ai eu quelques difficultés à fermer l'accès à ce service. Notez que la ligne permit-hosts que je montre utilise un "*" pour donner l'accès à tout le monde. Le paramétrage correct de cette ligne est : `authsrv: permit-hosts localhost`, si vous arrivez à la faire fonctionner.

```
#
# Table de configuration du mandataire
#
# Regles d'authentification client et serveur
authsrv:        database /usr/local/etc/fw-authdb
authsrv:        permit-hosts *
authsrv:        badsleep 1200
authsrv:        nobogus true
# Applications client utilisant le serveur d'authentification
*:              authserver 127.0.0.1 114
```

Pour initialiser la base, passez root et lancez **./authsrv** dans le répertoire /var/local/etc pour créer l'enregistrement de l'utilisateur d'administration.

La documentation de FWTK indique la manière d'ajouter des utilisateurs et des groupes.

Voici un exemple de session :

```
#
# authsrv
authsrv# list
authsrv# adduser admin "Auth DB admin"
ok - user added initially disabled
authsrv# ena admin
enabled
authsrv# proto admin pass
changed
authsrv# pass admin "plugh"
Password changed.
authsrv# superwiz admin
set wizard
authsrv# list
Report for users in database
user   group  longname           ok?    proto   last
------ ------ ------------------ -----  ------  -----
admin         Auth DB admin      ena    passw   never
authsrv# display admin
Report for user admin (Auth DB admin)
Authentication protocol: password
```

```
Flags: WIZARD
authsrv# ^D
EOT
#
```

Les contrôles de la passerelle telnet (tn-gw) vont de soi et sont la première chose que vous deviez configurer.

Dans mon exemple, j'autorise une machine du réseau privé à passer sans s'authentifier (permit-hosts 19961.2.* -passok). En revanche, tout autre utilisateur doit entrer ses nom et mot de passe pour utiliser le mandataire (permit-hosts * -auth).

J'autorise aussi un autre système (192.1.2.202) à accéder au pare-feu directement sans passer du tout par celui-ci. Les deux lignes inetacl-in.telnetd font cela. J'expliquerai plus loin comment ces lignes sont utilisées.

Le timeout de telnet doit rester court :

```
# regles de passerelle telnet :
tn-gw:          denial-msg      /usr/local/etc/tn-deny.txt
tn-gw:          welcome-msg     /usr/local/etc/tn-welcome.txt
tn-gw:          help-msg        /usr/local/etc/tn-help.txt
tn-gw:          timeout 90
tn-gw:          permit-hosts 192.1.2.* -passok -xok
tn-gw:          permit-hosts * -auth
# Seul l'administrateur peut telneter directement le pare-feu
# sur le port 24
netacl-in.telnetd: permit-hosts 192.1.2.202 -exec /usr/sbin/in.telnetd
```

Les commandes “r-” fonctionnent de la même manière que telnet :

```
# regles de passerelle rlogin :
rlogin-gw:      denial-msg      /usr/local/etc/rlogin-deny.txt
rlogin-gw:      welcome-msg     /usr/local/etc/rlogin-welcome.txt
rlogin-gw:      help-msg        /usr/local/etc/rlogin-help.txt
rlogin-gw:      timeout 90
rlogin-gw:      permit-hosts 192.1.2.* -passok -xok
rlogin-gw:      permit-hosts * -auth -xok
# Seul l'administrateur peut telneter directement le pare-feu
# sur le port
netacl-rlogind: permit-hosts 192.1.2.202 -exec /usr/libexec/rlogind -a
```

Personne ne devrait avoir accès directement au pare-feu, et cela inclut FTP, donc ne placez pas de serveur FTP sur votre pare-feu.

À nouveau, la ligne permit-hosts autorise quiconque depuis le réseau protégé à accéder librement à InterNet et tous les autres utilisateurs doivent s'authentifier. J'ai ajouté la trace de chaque fichier envoyé et reçu dans mes contrôles (-log { retr stor }).

Le timeout FTP contrôle le temps mis à fermer une mauvaise connexion, ainsi que le temps d'inactivité maximal d'une session ouverte :

```
# regles de passerelle ftp :
ftp-gw:         denial-msg      /usr/local/etc/ftp-deny.txt
ftp-gw:         welcome-msg     /usr/local/etc/ftp-welcome.txt
ftp-gw:         help-msg        /usr/local/etc/ftp-help.txt
```

```
ftp-gw:         timeout 300
ftp-gw:         permit-hosts 192.1.2.* -log { retr stor }
ftp-gw:         permit-hosts * -authall -log { retr stor }
```

Le web, gopher et le ftp fondé sur un butineur sont contrôlés par le http-gw. Les deux premières lignes créent un répertoire pour stocker les documents ftp et web lorsqu'ils passent au-travers du pare-feu. Je rends root propriétaire de ces fichiers et je les place dans un répertoire accessible seulement par root.

La connexion web doit être maintenue courte. Elle contrôle le temps durant lequel un utilisateur attendra lors d'une mauvaise connexion :

```
# regles de passerelle www et gopher :
http-gw:        userid          root
http-gw:        directory       /jail
http-gw:        timeout 90
http-gw:        default-httpd   www.afs.net
http-gw:        hosts           192.1.2.* -log { read write ftp }
http-gw:        deny-hosts      *
```

Le ssl-gw est juste une passerelle “gruyère”. Faites-y attention. Dans cet exemple, j'autorise quiconque depuis le réseau protégé à se connecter en-dehors du réseau sauf les adresses 127.0.0.* et 192.1.1.*, puis seulement sur les ports 443 à 563. Ces derniers sont les ports SSL connus :

```
# Regles de passerelle SSL :
ssl-gw:  timeout 300
ssl-gw:  hosts   192.1.2.* -dest { !127.0.0.* !192.1.1.* *:443:563 }
ssl-gw:  deny-hosts *
```

Voici un exemple d'utilisation de plug-gw pour autoriser des connexions à un serveur de nouvelles. Dans cet exemple j'autorise quiconque depuis le réseau protégé à se connecter seulement à un système et seulement sur son port de nouvelles.

La seconde ligne permet au serveur de renvoyer ses données au réseau protégé.

Puisque de nombreux clients s'attendent à rester connectés pendant que l'utilisateur lit les nouvelles, le timeout pour un serveur de nouvelles doit être long :

```
# passerelle plug-in pour les nouvelles :
plug-gw: timeout 3600
plug-gw: port nntp 192.1.2.* -plug-to 199.5.175.22 -port nntp
plug-gw: port nntp 199.5.175.22 -plug-to 192.1.2.* -port nntp
```

La passerelle finger est simple. Quiconque depuis le réseau protégé doit se connecter d'abord, puis nous l'autorisons à utiliser le programme finger du pare-feu. Tout autre reçoit simplement un message :

```
# Autorise le service finger :
netacl-fingerd: permit-hosts 192.1.2.* -exec /usr/libexec/fingerd
netacl-fingerd: permit-hosts * -exec /bin/cat /usr/local/etc/finger.txt
```

Je n'ai pas configuré les services courriel ni X-Window, donc je n'inclus pas les exemples. Si quelqu'un dispose d'un exemple qui fonctionne, qu'il me l'envoie par courriel.

10.4.2 Le fichier /etc/services

C'est là que tout commence. Lorsqu'un client se connecte sur le pare-feu, il le fait sur un port connu (inférieur à 1024). Par exemple, telnet se connecte sur le port 23. Le daemon inetd détecte cette connexion et cherche le nom du service dans le fichier /etc/services. Ensuite, il lance le programme assigné au nom dans le fichier /etc/inetd.conf.

Certains des services que nous créons ne sont pas normalement dans le fichier /etc/services. Vous pouvez assigner à certains d'entre eux le port que vous souhaitez. Par exemple, j'ai assigné le port telnet de l'administrateur (telnet-a) sur le port 24. Pous pouvez l'assigner au port 2323 si vous voulez. Pour que l'administrateur (VOUS) se connecte directement sur le pare-feu, il doit utiliser telnet sur le port 24 et non 23 et si vous paramétrez votre netperm-table comme je l'ai fait, vous serez seulement capable de faire cela depuis un système situé à l'intérieur du réseau protégé.

```
telnet-a    24/tcp
ftp-gw      21/tcp          # ce nom est modifie
auth        113/tcp  ident  # Verification utilisateur
ssl-gw      443/tcp
```

11 Le serveur mandataire SOCKS

11.1 Installation du serveur mandataire

Le serveur mandataire SOCKS est disponible sur : *http://www.socks.nec.com/*.

Décompressez et "dé-tarez" les fichiers dans un répertoire de votre système, et suivez les instructions pour le confectionner. J'ai eu quelques problèmes pour le réaliser. Vérifiez que vos Makefiles soient corrects.

Une chose importante est que le serveur mandataire doit être ajouté dans /etc/inetd.conf. Vous devez ajouter une ligne :

```
socks  stream  tcp  nowait  nobody  /usr/local/etc/sockd  sockd
```

pour indiquer au serveur de s'exécuter sur demande.

11.2 Configuration du serveur mandataire

Le programme de connexion nécessite deux fichiers de configuration distincts : l'un pour indiquer les accès autorisés, l'autre pour rediriger les requêtes vers le serveur mandataire approprié. Le fichier d'autorisations d'accès doit se trouver sur le serveur. Le fichier de routage peut être placé sur n'importe quelle machine Unix. Les ordinateurs DOS et, je pense, les Macintosh font leur propre routage.

11.2.1 Le fichier d'accès

Avec socks4.2 bêta, le fichier d'accès s'appelle "sockd.conf". Il doit contenir deux lignes : une ligne d'autorisations et une ligne d'interdictions. Chaque ligne présente trois champs :

- l'identificateur (permit ou deny) ;
- l'adresse IP ;
- le modificateur d'adresse.

L'identificateur est soit permit, soit deny. Vous devez avoir aussi bien une ligne permit qu'une ligne deny.

L'adresse IP contient une adresse à quatre octets en notation classique IP, soit, par exemple, 192.168.2.0.

Le masque de modification d'adresse est aussi une adresse à quatre octets en notation classique IP, et fonctionne comme un masque réseau. Représentez-vous ce nombre sur 32 bits. Si un bit est à 1, le bit correspondant de l'adresse qu'il contrôle doit concorder avec le bit correspondant du champ de l'adresse IP.

Par exemple, une ligne :

```
permit 192.168.2.23 255.255.255.255
```

autorisera seulement l'adresse dont chaque bit correspond à 192.168.2.23, donc seulement 192.168.2.23.

Tandis que la ligne :

```
permit 192.168.2.0 255.255.255.0
```

autorisera toute adresse du groupe 192.168.2.0 à 192.168.2.255, soit tout le domaine de la classe C.

Il ne faut pas spécifier la ligne :

```
permit 192.168.2.0 0.0.0.0
```

qui autoriserait toute adresse, sans distinction.

Aisni, autorisez toute adresse que vous souhaitez, puis interdisez le reste. Pour autoriser quiconque dans le domaine 192.168.2.xxx, les lignes :

```
permit 192.168.2.0 255.255.255.0
deny 0.0.0.0 0.0.0.0
```

fonctionneront très bien. Notez le premier "0.0.0.0" dans la ligne "deny". Avec un modificateur de 0.0.0.0, le champ adresse IP n'a aucune importance. Tous les champs à 0 représentent la norme, car c'est facile à écrire.

On peut utiliser plusieurs lignes de chaque type.

Des utilisateurs spécifiques peuvent aussi se voir accorder ou refuser l'accès. Cela est réalisé par l'authentification d'identité. Tous les systèmes ne supportent pas le système ident, y compris Trumpet Winsock, donc nous n'irons pas plus loin en ce qui concerne cela. La documentation de socks est tout à fait adéquate sur ce sujet.

11.2.2 Le fichier de routage

Le fichier de routage de socks est bêtement nommé "socks.conf". Je dis "bêtement", car il est si proche du nom du fichier d'accès qu'il est aisé de les confondre.

Le fichier de routage sert à indiquer aux clients de socks quand il est nécessaire de l'utiliser et quand ce n'est pas le cas. Par exemple, dans notre réseau, 192.168.2.3 ne nécessite pas l'usage de socks pour communiquer avec le pare-feu 192.168.2.1. Il a une connection directe Ethernet. Il définit 127.0.0.1, le port de bouclage, automatiquement. Evidemment, il n'est pas nécessaire d'utiliser socks pour vous parler à vous-même.

Il y a trois entrées :

- deny ;
- direct ;
- sockd.

L'entrée "deny" indique à socks quand rejeter une requête. Cette entrée a les trois mêmes champs que ceux de sockd.conf : identificateur, adresse et modificateur. Généralement, puisqu'il est aussi manipulé par sockd.file, le fichier d'accès, le champ modificateur est positionné à 0.0.0.0. Si vous voulez vous interdire tout appel vers l'extérieur, vous pouvez le réaliser ici.

L'entrée "direct" indique pour quelles adresses ne pas utiliser socks. Il s'agit des adresses pouvant être atteintes sans le serveur mandataire. A nouveau, nous avons les trois champs identificateur, adresse et modificateur.

Dans notre exemple, nous aurions :

```
direct 192.168.2.0 255.255.255.0
```

donnant ainsi l'accès direct pour toute machine de notre réseau protégé.

L'entrée "sockd" indique à l'ordinateur l'emplacement du démon serveur de socks.

La syntaxe est la suivante :

```
sockd @=<liste de serveurs> <adresse IP> <modificateur>
```

Notez l'entrée @=. Elle vous permet de configurer les adresses IP de plusieurs serveurs mandataires. Dans notre exemple, nous utilisons un seul serveur mandataire, mais vous pouvez en avoir plusieurs pour permettre un plus grand trafic et pour assurer une tolérance aux pannes.

Les champs adresse IP et modificateur fonctionnent exactement comme dans les autres exemples. Vous spécifiez ainsi où va quelle adresse.

11.2.3 DNS depuis l'arrière d'un pare-feu

Configurer un service de noms de domaines depuis l'arrière d'un pare-feu est une tâche relativement simple. En gros, il vous faut configurer le DNS sur la machine pare-feu. Ensuite, indiquez à chaque machine derière le pare-feu d'utiliser celui-ci.

11.3 Travailler avec un serveur mandataire

11.3.1 Unix

Pour faire fonctionner vos applications avec un serveur mandataire, celles-ci doivent être "SOCK-ifiées". Il vous faudra deux telnet différents : un pour la communication directe, et un autre pour celle avec le serveur mandataire. Le paquetage socks contient des indications pour SOCK-ifier un programme, ainsi qu'un certain nombre de programmes pré-SOCK-ifiés. Si vous utilisez la version SOCK-ifiée pour aller à un emplacement direct, socks basculera automatiquement sur la version directe pour vous. Pour cette raison, il nous faut renommer tous les programmes sur notre réseau protégé et les remplacer par leur version SOCK-ifiée. "finger" devient "finger.orig", "telnet" devient "telnet.orig", etc. Vous devez indiquer chacun d'eux à socks à l'aide du fichier include/socks.

Certains programmes traitent le routage et la SOCK-ification eux-mêmes. Netscape est l'un d'entre eux. Vous pouvez utiliser un serveur mandataire sous Netscape en donnant l'adresse du serveur (192.168.2.1 dans le cas qui nous intéresse) dans le champ SOCKs sous Proxies. Chaque application nécessite au moins un petit coup d'oeil, quelle que soit son attitude vis-à-vis d'un serveur mandataire.

11.3.2 MS Windows avec Trumpet Winsock

Trumpet Winsock contient des fonctionnalités de serveur mandataire incluses. Dans le menu “setup”, donnez l'adresse IP du serveur, ainsi que celles de tous les ordinateurs directement accessibles. Trumpet se débrouillera alors avec tous les paquets sortants. NdT : Trumpet Winsock est une couche IP destinée à MS-Windows 3. Depuis la version 3.11 de Windows, Microsoft fournit une couche IP dont les fonctionnalités sont très différentes.

11.4 Faire fonctionner le serveur mandataire avec les paquets UDP

Le paquetage SOCKS fonctionne seulement avec les paquets TCP, pas avec les UDP. Cela le rend quelque peu moins utile. De nombreux programmes très utiles, comme talk et Archie, utilisent UDP. Il existe un paquetage prévu pour être utilisé comme serveur mandataire pour les paquets UDP appelé UDPrelay, de Tom Fitzgerald *fitz@wang.com*. Malheureusement, à l'heure où ces lignes sont écrites, il n'est pas compatible avec Linux.

11.5 Inconvénients des serveurs mandataire

Le serveur mandataire est, avant tout, un `système de sécurité`. Son utilisation pour augmenter le nombre d'accès Internet avec un nombre limité d'adresses aura de nombreux inconvénients. Un serveur mandataire autorisera un plus grand accès de l'intérieur du réseau protégé vers l'extérieur, mais laissera l'intérieur totalement inaccessible de l'extérieur. Ce qui implique aucun serveur, aucune connexion talk ni Archie, ni courriel direct vers les ordinateurs de l'intérieur. Ces inconvénients peuvent sembler légers, mais regardez-les sous l'angle suivant :

- Vous avez laissé un document en cours sur votre ordinateur à l'intérieur du réseau protégé. Vous êtes à la maison, et décidez que vous voulez retravailler celui-ci. Vous ne le pouvez pas. Vous ne pouvez atteindre votre ordinateur, car il est derrière le pare-feu. Vous essayez de vous loger d'abord sur le pare-feu, mais comme tout le monde a accès au serveur mandataire, personne ne vous a créé de compte dessus.
- Votre fille va à l'université. Vous souhaitez lui envoyer un courriel. Vous avez différents choses de caractère privé à discuter, et préfèreriez recevoir directement votre courrier sur votre machine. Vous avez pleine confiance dans votre administrateur réseau, mais, malgré tout, il s'agit de courrier privé.
- L'impossibilité d'utiliser les paquets UDP représente un gros inconvénient avec les serveurs mandataire. Je pense que les fonctionnalités UDP arriveront sous peu.

FTP cause un autre problème avec les serveurs mandataire : Lorsque FTP récupère une liste de fichiers, le serveur ouvre une socket sur la machine client pour lui envoyer les informations. Un serveur mandataire ne permettra pas cela, donc FTP en particulier ne fonctionne pas.

De plus, les serveurs mandataires sont lents. A cause de la dégradation du rapport information/protocole, n'importe quel autre moyen d'obtenir cet accès sera plus rapide.

En résumé, si vous avez les adresses IP nécessaires, et que la sécurité ne soit pas un impératif pour vous, n'utilisez ni un pare-feu ni un serveur mandataire. Si vous n'avez pas suffisamment d'adresses IP, mais que, de même, la sécurité n'est pas fondamentale, vous pouvez jeter un coup d'oeil aux émulateurs IP, comme Term, Slirp ou TIA. Term est disponible sur *ftp://sunsite.unc.edu*, Slirp est disponible sur *ftp://blitzen.canberra.edu.au/pub/slirp* et TIA est disponible sur marketplace.com. Ces paquetages iront plus vite, permettront de meilleures connexions, et fourniront un accès supérieur à l'intérieur du réseau depuis InterNet. Les serveurs mandataires sont utiles pour ce genre de réseaux qui comportent de nombreuses machines qui se connectent au vol à InterNet, avec une configuration et peu de travail ensuite.

12 Configurations avancées

Je voudrais aborder une configuration particulière avant de refermer ce document. Celle que j'ai soulignée précédemment suffira probablement pour de nombreux cas. Néammoins, je pense que la situation suivante montrera une configuration plus avancée qui éclaircira certains points d'ombre. S'il vous reste des questions après ce que je viens de décrire, ou simplement que l'adaptabilité des serveurs mandataires et des pare-feux vous intéresse, lisez encore.

12.1 Un grand réseau avec sécurité renforcée

Disons, par exemple, que vous êtes le gourou de la secte de la 23ème Cabale de la Discorde de Milwaukee. Vous souhaitez mettre votre site en réseau. Vous avez cinquante ordinateurs et un sous-réseau de trente-deux adresses IP (sur cinq bits). Vous avez différents niveaux d'accès parce que vous dites à vos disciples différentes choses en fonction de leur niveau. C'est pourquoi vous devez protéger certaines parties du réseau du reste.

(NdT : Le traducteur a conservé la 23ème Cabale de la Discorde de Milwaukee, issue du texte initial, contrairement à ce que contiennent les nouvelles versions (millisha : version militarisée) qui serait moins explicite du principe pour un public francophone).

Les niveaux sont les suivants :

1. Le niveau extérieur. C'est celui qui est montré à tout un chacun. En gros, c'est l'histoire et les ragots sur Eris, la divinité de la Discorde, et tout le reste du dogme ;
2. Sage. C'est le niveau des gens qui ont passé le niveau extérieur. C'est là que vous leur dites que la discorde et la structure ne font qu'un, et qu'Eris est aussi le Dieu tout-puissant ;
3. Adepte. C'est là que se trouve le plan réel. Dans ce niveau sont stockées toutes les informations sur la manière dont la secte des Discordiens prendra le pouvoir sur le monde, à l'aide d'un plan déviationniste, mais humoristique, impliquant PetitMou, HAL, R2D2, Nounours et cinq cents cristaux, tous marqués "80585,9999999997" par erreur.

12.1.1 La configuration du réseau

Les numéros IP sont arrangés ainsi :

- l'adresse 192.168.2.255 est l'adresse de diffusion, et n'est pas disponible ;
- 23 des 32 adresses IP sont allouées à 23 machines qui seront accessibles depuis InterNet ;
- Une adresse IP supplémentaire va à une machine Linux sur ce réseau ;
- Une autre va à une autre machine Linux de ce réseau ;
- 2 numéros IP vont au routeur ;
- 4 sont laissées libres, mais recoivent les noms de domaine paul, ringo, george et john, juste pour compliquer un peu les choses ;
- Les réseaux protégés ont tous deux des adresses 192.168.2.xxx.

Puis, deux réseaux séparés sont construits, chacun dans une pièce différente. Ils sont routés par Ethernet infrarouge pour les rendre complètement invisibles de la pièce extérieure. Par chance, l'Ethernet infrarouge fonctionne tout à fait comme l'Ethernet normal, donc il nous suffit de les considérer comme normaux.

Ces réseaux sont connectés chacun à sa machine Linux avec une adresse IP supplémentaire.

Un serveur de fichiers relie les deux réseaux protégés. C'est parce que les plans pour prendre le pouvoir sur le monde prennent en compte certains des sages les plus élevés. Le serveur de fichiers a les adresses 192.168.2.17

pour le réseau des sages et 192.168.2.23 pour le réseau des adeptes. Il doit avoir des adresses IP différentes, car il doit avoir deux cartes Ethernet différentes. La transmission IP y est désactivée.

La transmission IP est aussi désactivée sur les deux machines Linux. Le routeur ne transmettra pas les paquets destinés à 192.168.2.xxx sauf si on lui demande explicitement de le faire, donc InterNet ne pourra pas entrer. La raison de la désactivation de la transmission IP ici est d'empêcher les paquets du réseau des sages d'atteindre le réseau des adeptes, et vice versa.

Le serveur NFS peut aussi être configuré pour présenter différents fichiers aux différents réseaux. Cela peut devenir pratique, et assez astucieux d'utiliser les liens symboliques pour partager les fichiers communs. Cette configuration associée à une autre carte Ethernet peut ainsi permettre l'usage d'un seul serveur de fichiers pour les trois réseaux.

12.1.2 La configuration du bastion

Maintenant, puisque les trois niveaux doivent être capables de piloter le réseau pour leurs propres besoins déviationnistes, tous les trois ont besoin d'un accès InterNet. Le réseau extérieur est connecté directement à celui-ci, donc nous n'avons pas à nous préoccuper d'un serveur mandataire ici. Les réseaux des sages et des adeptes sont derrière des pare-feux, il est donc nécessaire de leur configurer des serveurs mandataires.

Les deux réseaux seront configurés de manière similaire. Tous deux ont les mêmes adresses IP assignées. Je vais ajouter quelques paramètres, afin de rendre les choses encore plus intéressantes :

1. Personne ne peut utiliser le serveur de fichiers pour l'accès Internet. Cela exposerait le serveur de fichiers aux virus et autres choses désagréables, et il est très important, donc il est derrière les limites ;
2. Nous ne voulons pas donner aux sages l'accès au web. Il sont encore en entraînement, et cette puissance de recherche d'informations peut se révéler dangereuse.

Ainsi, le fichier sockd.conf de la machine Linux des sages contiendra cette ligne :

```
deny 192.168.2.17 255.255.255.255
```

Et sur la machine des adeptes :

```
deny 192.168.2.23 255.255.255.255
```

Et la machine Linux des sages contiendra cette ligne :

```
deny 0.0.0.0 0.0.0.0 eq 80
```

Cela indique l'interdiction d'accès pour toutes les machines tentant d'accéder au port 80, le port http. Cela laisse l'accès à tous les autres services, et interdit juste l'accès Web.

Ensuite, les deux fichiers auront :

```
permit 192.168.2.0 255.255.255.0
```

pour permettre à tous les ordinateurs du réseau 192.168.2.xxx d'utiliser ce serveur mandataire sauf pour ceux à qui cela a déjà été interdit (i.e. le serveur de fichiers et l'accès Web pour le réseau des sages).

Le fichier sockd.conf du réseau des sages aura l'allure suivante :

```
deny 192.168.2.17 255.255.255.255
deny 0.0.0.0 0.0.0.0 eq 80
permit 192.168.2.0 255.255.255.0
```

et le fichier des adeptes aura celle-ci :

```
deny 192.168.2.23 255.255.255.255
permit 192.168.2.0 255.255.255.0
```

Cela doit tout configurer correctement. Chaque réseau est isolé comme il faut, avec le niveau d'interaction approprié. Chacun peut être heureux. Maintenant, prenez le pouvoir sur le monde !

13 Simplifier l'administration

Un certain nombre de paquetages peuvent rendre l'administration de votre pare-feu plus simple :

– Webmin *http://www.webmin.com/*.

[Lister les URL ICI]

14 Outrepasser un pare-feu mandataire

Juste pour vous gâcher la journée et vous maintenir dans l'esprit de sécurité, je vais décrire combien il est facile d'outrepasser un pare-feu mandataire.

Imaginons que vous ayez réalisé tout ce qui se trouve dans le présent document et que vous disposiez d'un réseau et d'un serveur très sécurisés. Vous disposez d'une zone démilitarisée, personne ne peut entrer dans votre réseau et vous tracez chaque connexion réalisée vers le monde extérieur. Vous obligez tous vos utilisateurs à passer par un mandataire et le seul service que vous autorisiez directement vers l'extérieur est le DNS (port 53).

Un port, c'est tout ce qu'il faut pour rendre un pare-feu inutile. Voici comment cela se passe :

Commencez par configurer une machine Linux quelque part en-dehors de votre réseau. Un bon choix pourrait être une machine personnelle connectée à Internet par un modem-câble.

Demandez trois adresses IP à votre FAI. De nombreuses entreprises en fournissent jusqu'à trois.

Sur cette machine, vous devez installer la partie client d'un réseau privé virtuel (VPN). Cherchez sur *http://sunsite.auc.dk/vpnd/*.

Maintenant, configurez le côté serveur du VPN sur une autre machine Linux. Connectez ce serveur à son client par le port 53. Activez le routage et la transmission IP et placez une adresse IP inutilisée (obtenue de votre FAI) sur son port réseau local.

Finalement, sur un poste du réseau privé, changez la passerelle par défaut afin qu'elle pointe sur le serveur du VPN et ajoutez la troisième adresse IP sur son port réseau local.

Maintenant, depuis ce poste, vous pouvez aller n'importe où. La seule chose que l'administrateur voie est une recherche DNS particulièrement longue.

Maintenant, prenez le pouvoir sur le monde !