Le HOWTO du Firewalling et des serveurs Proxy

David Rudder, drig@execpc.com (originellement par Mark Grennan) v0.4, 8 novembre 1996 - Adaptation française Bernard Choppy, choppy@imaginet.fr 19 novembre 1997

Ce document est destiné à enseigner les bases d'un firewall ainsi que pour donner quelques détails sur la configuration d'un firewall aussi bien filtrant que serveur proxy sur un PC sous Linux. Une version HTML de la version originale en anglais de ce document est disponible à http://okcforum.org/~markg/Firewall-HOWTO.html.

Table des matières

1 Introduction **3**

1.1 Réactions . . . 3

1.2 Avertissement . . . 3

1.3 Copyright . . . 3

1.4 Les raisons qui me poussent à écrire ceci . . . 4

1.5 Reste à faire . . . 4

1.6 Autres lectures . . . 4

2 Comprendre les firewalls **5**

2.1 Inconvénients des firewalls . . . 5

2.2 Types de firewalls . . . 5

2.2.1 firewalls filtrants IP . . . 5

2.3 Serveurs proxy . . . 6

3 Configurer le firewall **6**

3.1 Matériel nécessaire . . . 6

4 Logiciel de firewall **6**

4.1 Paquetages disponibles . . . 6

4.2 FWTK contre SOCKS . . . 7

5 Préparer le système Linux **7**

5.1 Compiler le noyau . . . 7

5.2 Configurer deux cartes réseau . . . 8

5.3 Configurer les adresses réseau 9
5.4 Tester le réseau 10
5.5 Sécuriser le firewall 11

6 Configuration du filtrage IP (IPFWADM) 11

7 Installer le serveur proxy TIS 13

7.1 Trouver le logiciel 13
7.2 Compiler le TIS fwtk 14
7.3 Installer le TIS fwtk 15
7.4 Configurer le TIS fwtk 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 : 15
7.4.1 Le fichier netperm-table 15
7.4.2 Le fichier inetd.conf 19
7.4.3 Le fichier /etc/services 20

8 Le serveur proxy SOCKS 21

8.1 Installation du serveur proxy 21
8.2 Configuration du serveur proxy 21
8.2.1 Le fichier d'accès 21
8.2.2 Le fichier de routage 22
8.2.3 DNS depuis l'arrière d'un firewall 23
8.3 Travailler avec un serveur proxy 23
8.3.1 Unix 23
8.3.2 MS Windows avec Trumpet Winsock 24
8.4 Faire fonctionner le serveur proxy avec les paquets UDP 24
8.5 Inconvénients des serveurs proxy 24

9 Configurations avancées 25

9.1 Un grand réseau avec sécurité renforcée 25
9.1.1 La configuration du réseau 26
9.1.2 La configuration proxy 26

1 Introduction

Ce Firewall-HOWTO a été originellement écrit par David Rudder, mailto:drig@execpc.com. Je voudrais le remercier de m'avoir autorisé à mettre son travail à jour.

Les firewalls ont obtenu une grande renommée 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 firewall, de sa configuration, des serveurs proxy, de leur configuration, et des applications de cette technologie hors du contexte de la sécurité.

NdT : Diverses traductions ont été proposées pour le terme *firewall*, dont pare-feu, coupe-feu, mur anti-feu, etc. Aucune ne correspondant actuellement au concept précis de firewall en anglais (il s'agit d'une pièce automobile séparant l'habitacle du moteur), l'auteur a préféré conserver ce terme de firewall, plutôt qu'une traduction arbitraire.

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 en trouvez, il m'intéresse au plus haut point de les corriger. Je tenterai de répondre à tout e-mail, mais je suis occupé, donc ne m'en veuillez pas si ce n'est pas le cas pour le vôtre.

Mon adresse e-mail est : mailto:markg@netplus.net.

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 firewalls et des serveurs proxy. 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 d'autres. 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 Copyright

Sauf mention contraire, les documents Linux HOWTO sont la propriété de leurs auteurs respectifs. Les documents Linux HOWTO peuvent être reproduits et distribués en totalité ou en partie, sur tout support physique ou électronique, tant que cette notice de copyright est présente sur chaque copie. La redistribution commerciale est autorisée et encouragée ; néammoins, l'auteur souhaite être informé de toute distribution de ce genre.

Toute traduction, travail dérivé, ou agrégat incorporant tout ou partie d'un ou plusieurs documents Linux HOWTO doit être couvert par ce même copyright. Ce qui veut dire que vous ne pouvez produire un travail dérivé d'un HOWTO et imposer des restrictions supplémentaires concernant sa distribution. Des exceptions à ces règles peuvent être délivrées sous certaines conditions ; contactez le coordinateur des Linux HOWTO.

En bref, nous souhaitons promouvoir la dissémination de cette information à travers autant de canaux que possible. Néammoins, nous souhaitons conserver un copyright sur les documents HOWTO, et être avisés de tout plan de distribution les concernant.

Si vous avez des questions, veuillez contacter Mark Grennan à mailto:markg@netplus.net.

NdT : Le traducteur 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 : mailto:choppy@imaginet.fr

1.4 Les raisons qui me poussent à écrire ceci

Même s'il y a eu de nombreux fils de discussion sur comp.os.linux.* dans les dernières années sur les firewall, j'ai trouvé difficilement les informations dont j'avais besoin pour monter un firewall. La version originelle de ce HOWTO était utile, mais encore incomplète. J'espère que cette version complétée du Firewall HOWTO de David Rudder fournira à chacun l'information nécessaire pour créer un firewall opérationnel en quelques heures, et non en quelques semaines.

Je pense aussi devoir rendre quelque chose à la communauté Linux.

1.5 Reste à faire

- donner quelques instructions sur la configuration des clients ;
- trouver un bon serveur proxy UDP qui fonctionne sous Linux.

1.6 Autres lectures

- le NET-2 HOWTO ;
- l'Ethernet HOWTO ;
- le Multiple Ethernet mini HOWTO ;
- Networking with Linux ;
- le PPP HOWTO ;
- TCP/IP Network Administrator's Guide chez O'Reilly&Associates ;
- La documentation de la boîte à outils de firewall TIS.

Le site web de Trusted Information System (TIS) dispose d'une bonne collection de documentaions sur les firewalls et sujets connexes : http://www.tis.com.

Je travaille aussi sur un projet de sécurité appelé *Secure Linux* (Linux sécurisé). Je collecte sur le site de *Secure Linux* toute information, documentation et programme nécessaire pour créer un système Linux sécurisé. Envoyez-moi un e-mail si vous souhaitez des informations.

2 Comprendre les firewalls

Firewall est un terme automobile. Dans une voiture, un firewall est une pièce qui sépare le bloc-moteur du compartiment passagers. Il est prévu pour protéger les passagers en cas de feu au moteur en maintenant le contrôle de ce dernier par le conducteur.

En informatique, un firewall est un périphérique qui protège la partie privée d'un réseau de la partie publique (InterNet en entier).

L'ordinateur firewall, ci-après dénommé « firewall », peut atteindre aussi bien le réseau protégé qu'InterNet. Le réseau protégé ne peut atteindre InterNet, et ce dernier, à son tour, ne peut atteindre le réseau protégé.

Pour atteindre InterNet depuis l'intérieur du réseau protégé, il faut se connecter au firewall par le réseau, puis utiliser InterNet à partir de là.

La forme la plus simple de firewall est un système à deux connexions réseau. Si vous pouvez AVOIR CONFIANCE EN TOUS vos utilisateurs, vous pouvez configurer simplement Linux (compilé avec IP forwarding/gatewaying OFF !) et leur donner à chacun un compte dessus. Il pourront dès lors se connecter dessus, et utiliser FTP, lire le courrier et utiliser tout autre service que vous fournirez. Avec cette configuration, le seul ordinateur de votre réseau privé qui sache quelque chose du monde extérieur est le firewall. Les autres systèmes sur votre réseau protégé ne nécessitent même pas une route par défaut.

Cela doit être reformulé. Pour que le firewall ci-dessus fonctionne, **VOUS DEVEZ AVOIR CONFIANCE EN TOUS VOS UTILISATEURS !** Je ne le recommande pas.

2.1 Inconvénients des firewalls

Le problème des firewalls filtrants réside dans le fait qu'ils empêchent l'accès l'intérieur depuis InterNet. Seuls les services sur les systèmes disposant de filtres de passage sont accessibles. Avec un serveur proxy, les utilisateurs peuvent se loger sur le firewall puis accéder à tout système interne auquel ils ont accès.

De même, de nouveaux types de clients réseau apparaissent à peu près chaque jour. Lorsque c'est le cas, vous devez trouver une nouvelle manière d'autoriser l'accès contrôlé avant que ces services puissent être utilisés.

2.2 Types de firewalls

Il y a deux types de firewalls :

1. firewalls IP ou filtrants - ils bloquent tout le trafic sauf celui sélectionné ;
2. serveurs proxy - ils réalisent les connexions réseau pour vous.

2.2.1 firewalls filtrants IP

Un firewall filtrant IP fonctionne au niveau paquet. Il est conçu pour contrôler le flux de paquets en fonction de l'origine, la destination, le port et l'information de type de paquet contenue dans chacun de ceux-ci.

Ce type de firewall est très sûr mais incomplet en termes de connexion aisée. Il peut bloquer l'accès des gens à un système privé mais ne vous indiquera pas qui a accédé à vos systèmes publics ni qui a accédé à InterNet depuis l'intérieur.

Les firewalls filtrants sont des filtres absolus. Même si vous voulez donner accès à quelqu'un d'extérieur sur vos serveurs privés, ce n'est pas possible sans donner accès aux serveurs à tout le monde.

Linux inclut un logiciel de filtrage de paquets dans les noyaux depuis 1.3.x.

2.3 Serveurs proxy

Les serveurs proxy permettent l'accès indirect à InterNet depuis l'arrière d'un firewall. Le meilleur exemple du fonctionnement de ceux-ci est celui d'une personne se connectant à un système puis, depuis celui-ci, à un autre. C'est seulement avec un serveur proxy que ce processus est automatique. Lorsque vous vous connectez à un serveur proxy à l'aide de votre logiciel client, le serveur proxy démarre son propre logiciel client (proxy) et vous transmet les données.

Puisque les serveurs proxy dupliquent toutes les communications, il speuvent enregistrer toute leur activité.

L'idée majeure concernant les serveurs proxy est qu'ils sont complètement sûrs, lorsqu'ils sont configurés correctement. Ils ne permettront à personne d'entrer au-travers d'eux. Il n'existe pas de route IP directe.

3 Configurer le firewall

3.1 Matériel nécessaire

Pour notre exemple, l'ordinateur est un 486-DX66 avec 16 Mo de RAM et 500 Mo de partition Linux. Ce système dispose de deux cartes réseau dont l'une est connectée à notre réseau privé et l'autre à un réseau que nous appellerons « zone démilitarisée » (ZD). La ZD dispose d'un routeur connectée à InterNet.

Il s'agit d'une configuration assez standard en entreprise. Il est possible d'utiliser une seule carte réseau et un modem avec PPP vers InterNet. Le principal est que le firewall doit disposer de deux adresses IP.

Je connais un tas de gens qui ont un petit réseau à la maison avec deux ou trois ordinateurs dessus. Il est possible d'envisager de placer tous les modems sur une machine Linux (peut-être un vieux 386) et de tous les connecter à InterNet avec l'équilibrage de charge (load balancing). Avec cette configuration, lorsque quelqu'un transfère des données, la présence de deux modems peut doubler la bande passante. :-)

4 Logiciel de firewall

4.1 Paquetages disponibles

Si tout ce qu'il vous faut est un firewall filtrant, vous n'avez besoin que de Linux et des paquetages réseau de base. Un paquetage qui peut n'être pas inclus dans votre distribution est le « IP Firewall Administration tool » (IPFWADM).

Il est disponible sur : http://www.xos.nl/linux/ipfwadm/

Si vous voulez configurer un serveur proxy, il vous faut l'un de ces paquetages :

- SOCKS ;
- TIS Firewall Toolkit (FWTK).

4.2 FWTK contre SOCKS

Trusted Information Systems (http://www.tis.com) a sorti une collection de programmes destinés à faciliter la mise en place d'un firewall. Les programmes font fondamentalement la même chose que ceux du paquetage SOCKS, mais avec une stratégie de conception différente. Là où SOCKS a un programme qui couvre toutes les transactions InterNEt, TIS fournit un programme pour chaque utilitaire qui souhaite utiliser le firewall.

Pour distinguer l'un de l'autre, nous allons utiliser l'exemple du web et de l'accès telnet. Avec SOCKS, on configure un fichier et un daemon. A l'aide de ceux-ci, aussi bien telnet que le web sont activés, ainsi que tout autre service qui n'a pas été désactivé.

Avec la boîte à outils TIS, on configure un daemon pour telnet et pour le web, ainsi qu'un fichier pour chaque. Une fois cela fait, tout autre accès Internet reste interdit jusqu'à son activation expresse. Si le daemon d'un service précis n'a pas été fourni (comme pour talk), il existe un daemon « prêt-à-brancher », mais ce n'est ni aussi souple, ni aussi aisé à configurer que pour les autres outils.

Cela peut sembler une différence mineure, mais c'en est une majeure : SOCKS vous permet d'être fainéant. Avec un serveur SOCKS mal configuré, quelqu'un peut avoir plus d'accès vers InterNet depuis l'intérieur que ce qu'il avait été prévu initialement. Avec la boîte à outils TIS, les gens de l'intérieur ont seulement les accès que l'administrateur système souhaite qu'ils aient.

SOCKS est plus facile à configurer, à compiler et permet une plus grande souplesse. La boîte à outils TIS est plus sûre si l'on souhaite réguler les utilisateurs de l'intérieur du réseau. Les deux fournissent une protection absolue depuis l'extérieur.

Je parlerai de l'installation des deux.

5 Préparer le système Linux

5.1 Compiler le noyau

Commencez avec une installation propre de votre distribution Linux (j'ai utilisé RedHat 3.0.3 et les exemples sont fondés sur cette distribution). Moins vous installez de logiciel, moins votre système aura de trous de sécurité, portes dérobées et/ou bogues susceptibles d'induire des problèmes de sécurité dans votre système, donc installez le moins possible d'applications.

Prenez un noyau stable. J'ai utilisé le noyau Linux 2.0.14 pour mon système. La documentation est donc fondée sur ses paramètres.

Vous devez recompiler le noyau Linux avec les options appropriées. A ce point, je vous renvoie au Kernel HOWTO, à l'Ethernet HOWTO et au NET-2 HOWTO si vous ne l'avez pas encore fait.

Voici les paramètres réseau que je sais fonctionner dans « make config » :

1. Dans la configuration générale :
   (a) Activez le support du réseau ;
2. Dans les options réseau :
   (a) Activez le firewall réseau,
   (b) Activez le protocole TCP/IP,
   (c) Désactivez la transmission/routage IP (forwarding/gatewaying) (SAUF si vous voulez utiliser le filtrage IP),
   (d) Activez le Firewalling IP,
   (e) Activez la trace des paquets de firewall (ce n'est pas indispensable, mais c'est une bonne idée),
   (f) DESactivez le masquage (masquerading - je ne traite pas ce sujet ici),
   (g) Activez la gestion des comptes (accounting),
   (h) DESactivez le tunneling,
   (i) DESactivez l'aliasing,
   (j) DESactivez le mode de compatibilité PC/TCP,
   (k) DESactivez l'ARP inversé (Reverse ARP),
   (l) Activez l'abandon des trames routées par la source ;
3. Dans le support des interfaces réseau :
   (a) Activez le support des équipements réseau,
   (b) Activez le support du pseudo-pilote réseau,
   (c) Activez Ethernet (10 ou 100Mbit),
   (d) Choisissez votre carte réseau ;

Ensuite, vous pouvez recompiler, réinstaller le noyau et rebouter. Les cartes réseau doivent apparaître dans la séquence de démarrage. Sinon, plongez-vous à nouveau dans les autres HOWTO jusqu'à ce que cela fonctionne.

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 :

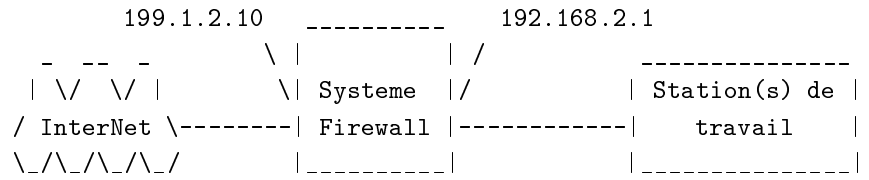
```
append="ether=12,0x300,eth0 ether=15,0x340,eth1"
```

5.3 Configurer les adresses réseau

C'est la partie réellement intéressante. Vous avez quelques décisions à y prendre. Puisque nous ne souhaitons pas laisser InterNet accéder au réseau privé, il n'est pas nécessaire d'utiliser des adresses réelles. Un certain nombre d'adresses InterNet ont été laissées de côté pour les réseaux privés. Puisque tout le monde a besoin de toujours plus d'adresses et puisque celles-ci ne peuvent circuler sur InterNet, il s'agit d'un bon choix.

Parmi celles-ci, nous utiliserons celles de 192.168.2.xxx pour nos exemples.

Votre firewall proxy sera membre des deux réseaux et ainsi pourra passer les données vers et depuis le réseau privé.



Si vous vous préparez à utiliser un firewall filtrant, ces adresses sont encore correctes. Vous devrez utiliser le masquage IP (masquerading) pour que cela fonctionne. Avec ce processus, le firewall route les paquets et les transforme en adresses IP « réelles » pour circuler sur InterNet.

Assignez l'adresse IP réelle à la carte réseau du côté InterNet (extérieur). Assignez 192.168.2.1 à la carte Ethernet du côté intérieur. Il s'agira de votre adresse proxy/gateway. Vous pouvez assigner une adresse de la plage 192.168.2.2 à 192.168.2.254 à toutes les autres machines du réseau protégé.

Puisque j'utilise RedHat Linux (Eh, les gars, faudrait voir à m'en offrir une copie, vu la pub? ;-), pour configurer le réseau lors du démarrage j'ai ajouté un fichier « ifcfg-eth1 » dans le répertoire /etc/sysconfig/network-scripts. Ce fichier est lu lors du processus de démarrage pour configurer le réseau et les tables de routage.

Voici l'allure de mon fichier ifcfg-eth1 :

```
#!/bin/sh
#>>>type de peripherique : ethernet
#>>>declaration de variables :
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
#>>>fin de declaration de variables
```

Vous pouvez aussi utiliser ces scripts pour vous connecter automatiquement par modem chez votre fournisseur. Jetez un coup d'oeil au script ipup-ppp.

Si vous utilisez un modem pour votre connexion InterNet, votre adresse IP sera assignée à votre machine à l'instant de la connexion.

5.4 Tester le réseau

Commencer par contrôler ifconfig et route. Si vous avez deux cartes réseau, votre ifconfig doit afficher quelque chose qui ressemble 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 firewall et 192.168.2.0 celle du côté privé.

Ensuite, essayez un ping sur InterNet depuis le firewall. J'avais l'habitude d'utiliser nic.ddn.mil comme point de test. C'est toujours un bon test, mais il a prouvé qu'il était moins fiable que je l'espérais. Si cela ne fonctionne pas dès l'abord, essayez un ping sur quelques autres sites qui ne soient pas connectés à votre réseau local. Si cela ne fonctionne pas, alors votre PPP est mal configuré. Relisez le NET-2 HOWTO, et essayez à nouveau.

Maintenant, essayez des ping sur les machines du réseau protégé depuis le firewall. Tous les ordinateurs doivent êtres capables d'atteindre tous les autres. Dans le cas contraire, repassez une couche de NET-2

HOWTO et retravaillez encore un peu votre réseau.

Puis, chaque machine du réseau protégé doit être capable d'atteindre le firewall par ping. Sinon, retournez une fois de plus en arrière.

Ensuite, essayez d'atteindre l'adresse extérieure du firewall depuis l'intérieur du réseau protégé (Note: il ne s'agit d'aucune des adresses 192.268.2.xxx). Si vous le pouvez, c'est que vous n'avez pas désactivé la transmission IP. Assurez-vous que ce soit bien ce que vous souhaitez. Si vous le laissez actif, vous devrez travailler le filtrage IP décrit plus loin dans ce document.

Enfin, essayez d'atteindre Internet depuis l'arrière de votre firewall. Utilisez la même adresse que celle qui avait fonctionné avant (par exemple, nic.ddn.mil). De même, si vous avez désactivé la transmission IP, cela ne devrait pas fonctionner, mais si vous l'avez activée, cela devrait.

Si vous avez activé la transmission IP et que vous utilisez une adresse IP « réelle » (pas 192.168.2.*) pour votre réseau privé et que vous ne puissiez atteindre InterNet, mais seulement le côté InterNet de votre firewall, contrôlez si le prochain routeur transmet bien les paquets pour votre adresse de réseau privé (votre fournisseur devrait avoir réalisé cela pour vous).

Le fait d'assigner le 192.168.2.* au réseau protégé indique qu'aucun paquet ne peut lui être transmis. Si vous avez déjà lu plus loin et que le masquage IP soit activé, ce test devrait fonctionner.

Vous avez maintenant une configuration de base.

5.5 Sécuriser le firewall

Le firewall n'est d'aucune utilité s'il reste ouvert aux attaques via un service inutilisé. Un « méchant » peut obtenir l'accès au firewall et le modifier pour ses desseins propres.

Commencez par désactiver tous les services inutilisés. Regardez le fichier `/etc/inet.conf`. Ce fichier contrôle ce qu'on appelle le « super-serveur ». Cela contrôle un tas de daemons serveurs et les exécute sur demande.

Désactivez définitivement netstat, systat, tftp, bootp et finger. Pour désactiver un service, placez simplement un « # » 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.

Testez le résultat par telnet sur le port 15 du firewall, qui est le port de netstat. Si vous obtenez une réponse de netstat, c'est que vous n'avez pas relancé inetd correctement.

6 Configuration du filtrage IP (IPFWADM)

Pour commencer, vous devez avoir activé la transmission IP (IP forwarding) dans le noyau et votre système doit être actif et transmettre tout ce que vous lui envoyez. Vos tables de routages doivent être en place et vous devez être en mlesure d'accéder à tout, aussi bien à l'extérieur depuis l'intérieur qu'à l'intérieur depuis l'extérieur.

Cela dit, nous construisons un firewall, donc il nous faut commencer à réduire ce à quoi tout le monde a accès.

Dans mon système, j'ai créé quelques scripts pour paramétrer le comportement de transmission et de la trace. J'appelle ces scripts depuis ceux de /etc/rc.d afin que mon système soit configuré dès le lancement.

Par défaut, le système de transmission IP route tout. Pour cette raison, votre script de firewall doit commencer par interdire l'accès à tout et par vider toutes les règles de trace en place depuis la dernière fois qu'il a été lancé. Le script suivant effectue le travail :

```
#
# 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 firewall absolu. Rien ne peut passer au travers. Sans doute devez-vous transmettre quelques services, donc voici quelques exemples que vous devriez trouver utiles :

```
# Autoriser l'acces de l'e-mail au serveur
ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 192.1.2.10 25

# Autoriser les connexions e-mail vers les serveurs exterieurs
ipfwadm -F -a accept -b -P tcp -S 196.1.2.10 25 -D 0.0.0.0/0 1024:65535

# Autoriser les connexions web vers le serveur web
/sbin/ipfwadm -F -a accept -b -P tcp -S 0.0.0.0/0 1024:65535 -D 196.1.2.11 80

# Autoriser les connexions web vers les serveurs web exterieurs
/sbin/ipfwadm -F -a accept -b -P tcp -S 196.1.2.* 80 -D 0.0.0.0/0 1024:65535

# Autoriser le trafic DNS
/sbin/ipfwadm -F -a accept -b -P udp -S 0.0.0.0/0 53 -D 196.1.2.0/24
```

Il peut être aussi intéressant de tracer le trafic qui passe par votre firewall. Ce script compte chaque paquet. Vous pouvez ajouter une ligne ou deux pour tracer les paquets qui vont vers un système particulier.

```
# Vider les regles de trace en cours
ipfwadm -A -f
# Trace
```

```
/sbin/ipfwadm -A -f
/sbin/ipfwadm -A out -i -S 196.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A out -i -S 0.0.0.0/0 -D 196.1.2.0/24
/sbin/ipfwadm -A in -i -S 196.1.2.0/24 -D 0.0.0.0/0
/sbin/ipfwadm -A in -i -S 0.0.0.0/0 -D 196.1.2.0/24
```

Si tout ce que vous cherchez est un firewall filtrant, vous pouvez vous arrêter ici. Amusez-vous bien :-)

7 Installer le serveur proxy TIS

7.1 Trouver le logiciel

Le TIS fwtk est disponible à ftp://ftp.tis.com.

Ne commettez pas l'erreur que j'ai commise : Lorsque vous téléchargez les ficheirs de TIS, LISEZ LES README. Le TIS fwtk est verrouillé dans un répertoire caché sur leur serveur. TIS impose que vous envoyiez **un e-mail à mailto:fwtk-request@tis.com** avec le seul mot **SEND** 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 e-mail (en gros sous 12 heures) pour charger le source.

A l'instant où j'écris, TIS diffuse la version 2.0(bêta) du fwtk. Cette version semble se compiler correctement (avec quelques exceptions) et tout fonctionne pour moi. C'est la version qui sera couverte ici. Lorsque le code final sera disponible, je mettrai à jour le HOWTO.

Pour installer le fwtk, créez un répertoire fwtk-2.0 dans votre répertoire /usr/src. Déplacez votre copie de fwtk (fwtk-2.0.tar.gz) dans ce répertoire et dé-tarez-le (tar zxf fwtk-2.0.tar.gz).

Le fwtk ne route pas les documents web SSL, mais il existe un module complémentaire écrit par Jean-Christophe Touvet. Il est disponible à ftp://ftp.edelweb.fr/pub/contrib/fwtk/ssl-gw.tar.Z. Christophe Touvet ne fournit aucun support pour ce code.

J'utilise une version modifiée écrite par Eric Wedel qui inclut l'accès aux serveurs de nouvelles sécurisés Netscape. Il est disponible à ftp://ftp.mdi.meridian-data.com/pub/tis.fwtk/ssl-gw/ssl-gw2.tar.Z.

Notre exemple s'appuie sur la version modifiée d'Eric Wedel.

Pour l'installer, créez simplement un répertoire ssl-gw dans votre répertoire /usr/src/fwtk-2.0 et placez-y les fichiers.

Lorsque j'ai installé cette passerelle, il nécessitait quelques modifications avant de se compiler avec le reste du paquetage.

La première modification était dans le fichier ssl-gw.c. Je me suis rendu compte qu'il omettait l'inclusion d'un fichier nécessaire :

```
#if defined(__linux)
#include        <sys/ioctl.h>
#endif
```

Ensuite, aucun fichier Makefile n'est fourni. J'en ai copié un depuis un des autres répertoires des passerelles et remplacé le nom de la passerelle avec ssl-gw.

7.2 Compiler le TIS fwtk

La version 2.0 du fwtk se compile nettement plus facilement qu'aucune des versions précédentes. J'ai encore trouvé quelques petites choses qui devaient être changées avant que la version bêta se compile correctement. Espérons que ces modifications seront réalisées dans la version finale.

Pour les corriger, commencez par aller dans le répertoire /usr/src/fwtk/fwtk et copiez le fichier Makefile.config.linux par-dessus le Makefile.config.

Ne lancez pas fixmake. Les instructions vous disent de le faire. Son exécution casse les Makefiles de tous les répertoires.

J'ai une correction pour fixmake. Le problème est que le script sed ajoute un « . » et un « " » sur la ligne d'include de chaque Makefile. Le script sed suivant fonctionne :

```
sed 's/^include[        ]*\([^  ].*\)/include \1/' $name .proto > $name
```

Ensuite, il faut éditer le fichier Makefile.config. Deux modifications sont nécessaires.

L'auteur définit le répertoire source dans son répertoire personnel. Nous compilons notre code dans /usr/src, donc vous devez changer la variable FWTKSRCDIR pour tenir compte de cela :

```
FWTKSRCDIR=/usr/src/fwtk/fwtk
```

Par ailleurs, au moins certains systèmes Linux utilisent la base gdbm. Le Makefile.config utilise dbm. Il est possible que vous deviez changer cela. J'ai dû le faire pour RedHat 3.0.3 :

```
DBMLIB=-lgdbm
```

La dernière correction est dans le x-gw. Le bogue de la version bêta est dans le code socket.c. Pour le corriger, retirez les lignes suivantes du code :

```
#ifdef SCM_RIGHTS  /* 4.3BSD Reno et suivants */
                   + sizeof(un_name->sun_len) + 1
#endif
```

Si vous avez ajouté ssl-gw dans votre répertoire source fwtk, vous devez l'ajouter dans la liste des répertoires dans le Makefile :

```
DIRS=   smap smapd netacl plug-gw ftp-gw tn-gw rlogin-gw http-gw x-gw ssl-gw
```

Maintenant, vous pouvez lancer **make**.

7.3 Installer le TIS fwtk

Lancer **make install**.

Le répertoire d'installation par défaut est /usr/local/etc. Vous pouvez modifier cela (je ne l'ai pas fait) pour un répertoire plus sécurisé. J'ai choisi de placer les droits d'accès à ce répertoire à 700 (« chmod 700 »).

Ce qu'il reste au moins maintenant est de configurer le firewall.

7.4 Configurer le TIS fwtk 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.

7.4.1 Le fichier netperm-table

Ce fichier contrôle qui a accès aux services de TIS fwtk. Vous devez penser au trafic qui passe par le firewall 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 firewall 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 permi-hosts que je montre utilise un « * » pour donner l'accès à tout le monde. Le paramétrage correcte de cette ligne est : `authsrv: premit-hosts localhost`, si vous arrivez à la faire fonctionner.

```
#
# Table de configuration proxy
#
# 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.

Lisez la documentation de fwtk pour 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 proxy (permit-hosts * -auth).

J'autorise aussi un autre système (196.1.2.202) à accéder au firewall 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 196.1.2.* -passok -xok
tn-gw:          permit-hosts * -auth
# Seul l'administrateur peut telneter directement le firewall
# sur le port 24
netacl-in.telnetd: permit-hosts 196.1.2.202 -exec /usr/sbin/in.telnetd
```

Les commandes « r » fonctionnent de la même manière que pour 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 196.1.2.* -passok -xok
rlogin-gw:      permit-hosts * -auth -xok
# Seul l'administrateur peut telneter directement le firewall
# sur le port
netacl-rlogind: permit-hosts 196.1.2.202 -exec /usr/libexec/rlogind -a
```

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

A 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 196.1.2.* -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 firewall. 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           196.1.2.* -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 :

```
# ssl gateway rules:
ssl-gw:  timeout 300
ssl-gw:  hosts   196.1.2.* -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 connecté 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 196.1.2.* -plug-to 199.5.175.22 -port nntp
plug-gw: port nntp 199.5.175.22 -plug-to 196.1.2.* -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 firewall. Tout autre reçoit simplement un message :

```
# Autorise le service finger :
netacl-fingerd: permit-hosts 196.1.2.* -exec /usr/libexec/fingerd
netacl-fingerd: permit-hosts * -exec /bin/cat /usr/local/etc/finger.txt
```

Je n'ai pas configuré les services e-mail 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 e-mail.

7.4.2 Le fichier inetd.conf

Voici un fichier inetd.conf complet. Tous les services non indispensables ont été placés en commentaire. J'ai inclus le fichier complet pour indiquer quoi désactiver, ainsi que la manière de configurer les nouveaux services du firewall:

```
#echo    stream tcp  nowait  root       internal
#echo    dgram  udp  wait   root        internal
#discard stream tcp  nowait  root       internal
#discard dgram  udp  wait   root        internal
#daytime stream tcp  nowait  root       internal
#daytime dgram  udp  wait   root        internal
#chargen stream tcp  nowait  root       internal
#chargen dgram  udp  wait   root        internal
# passerelle firewall FTP
ftp-gw   stream tcp nowait.400 root /usr/local/etc/ftp-gw ftp-gw
# passerelle firewall Telnet
telnet   stream tcp nowait root /usr/local/etc/tn-gw /usr/local/etc/tn-gw
# services locaux telnet
telnet-a stream  tcp  nowait root /usr/local/etc/netacl in.telnetd
# passerelle firewall Gopher
gopher stream tcp nowait.400 root /usr/local/etc/http-gw /usr/local/etc/http-gw
# passerelle firewall WWW
http    stream tcp  nowait.400 root /usr/local/etc/http-gw /usr/local/etc/http-gw
# passerelle firewall SSL
ssl-gw stream tcp nowait root /usr/local/etc/ssl-gw ssl-gw
# NetNews firewall proxy (using plug-gw)
nntp stream tcp nowait root /usr/local/etc/plug-gw plug-gw nntp
#nntp stream tcp nowait root /usr/sbin/tcpd in.nntpd
# passerelle firewall SMTP (e-mail)
#smtp stream tcp nowait root /usr/local/etc/smap smap
#
# Shell, login, exec et talk sont des protocoles BSD.
#
#shell stream tcp nowait root /usr/sbin/tcpd in.rshd
#login stream tcp nowait root /usr/sbin/tcpd in.rlogind
#exec  stream tcp nowait root /usr/sbin/tcpd in.rexecd
#talk  dgram  udp wait   root /usr/sbin/tcpd in.talkd
#ntalk dgram  udp wait   root /usr/sbin/tcpd in.ntalkd
#dtalk stream tcp wait nobody /usr/sbin/tcpd in.dtalkd
#
# services Pop et e-mail imap et autres
#
```

```
#pop-2 stream tcp nowait root /usr/sbin/tcpd ipop2d
#pop-3 stream tcp nowait root /usr/sbin/tcpd ipop3d
#imap  stream tcp nowait root /usr/sbin/tcpd imapd
#
# Le service InterNet UUCP
#
#uucp stream tcp nowait uucp /usr/sbin/tcpd /usr/lib/uucp/uucico -l
#
# Le service tftp sert essentiellement pour bouter. De nombreux sites
# l'utilisent seulement sur les "serveurs de boot"
# Ne pas decommenter sauf s'il vous le *faut*.
#
#tftp   dgram udp wait root /usr/sbin/tcpd in.tftpd
#bootps dgram udp wait root /usr/sbin/tcpd bootpd
#
# Finger, systat et netstat donnent des informations qui peuvent etre
# utiles a des "craqueurs de systemes" potentiels. De nombreux
# sites coisissent de desactiver certains ou tous ces services
# pour ameliorer la securite.
#
# cfinger est GNU finger, qui n'est pas utilise dans RHS Linux
#
finger   stream tcp nowait root  /usr/sbin/tcpd in.fingerd
#cfinger stream tcp nowait root  /usr/sbin/tcpd in.cfingerd
#systat  stream tcp nowait guest /usr/sbin/tcpd /bin/ps -auwwx
#netstat stream tcp nowait guest /usr/sbin/tcpd /bin/netstat -f inet
#
# Le service Time sert a la synchronisation des horloges.
#
#time stream tcp nowait root /usr/sbin/tcpd in.timed
#time dgram  udp wait   root /usr/sbin/tcpd in.timed
#
# Authentification
#
auth    stream tcp wait   root /usr/sbin/tcpd in.identd -w -t120
authsrv stream tcp nowait root /usr/local/etc/authsrv authsrv
#
# Fin de inetd.conf
```

7.4.3 Le fichier /etc/services

C'est là que tout commence. Lorsqu'un client se connecte sur le firewall, 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 firewall, 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
```

8 Le serveur proxy SOCKS

8.1 Installation du serveur proxy

Le serveur proxy SOCKS est disponible sur : ftp://sunsite.unc.edu/pub/Linux/system/Network/misc/socks-linux-src.tgz. Il ya aussi un exemple de fichier de configuration dans ce répertoire « socks-conf ». décompressez et « détarez » les fichiers dans un répertoire de votre système, et suivez les instructions pour le confectionner. J'ai eu quelques problèmes pour le réaliser. Vérifiez les Makefiles.

Une chose importante est que le serveur proxy 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.

8.2 Configuration du serveur proxy

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 proxy 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.

8.2.1 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 classe C.

Il ne faut pas spécifier la ligne :

```
permit 192.168.2.0 0.0.0.0
```

qui autoriserait toute adresse, sans distinction.

Bien, d'abord 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 est 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.

8.2.2 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 d'utiliser celui-ci. Par exemple, dans notre réseau, 192.168.2.3 ne nécessite pas l'usage de socks pour communiquer avec le firewall 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 protéger de tout appel externe, vous pouvez le faire là.

L'entrée « direct » indique pour quelles adresses ne pas utiliser socks. Il s'agit des adresses pouvant être atteintes sans le serveur proxy. 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 proxy. Dans notre exemple, nous utilisons un seul serveur proxy, 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.

8.2.3 DNS depuis l'arrière d'un firewall

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

8.3 Travailler avec un serveur proxy

8.3.1 Unix

Pour faire fonctionner vos applications avec un serveur proxy, 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 proxy.

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 proxy 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 proxy.

8.3.2 MS Windows avec Trumpet Winsock

Trumpet Winsock contient des fonctionnalités de serveur proxy 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.

8.4 Faire fonctionner le serveur proxy 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 proxy 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.

8.5 Inconvénients des serveurs proxy

Le serveur proxy 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 proxy 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 e-mail 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 firewall. Vous essayez de vous loger d'abord sur le firewall, mais comme tout le monde a accès au serveur proxy, Persoinne ne vous a créé de compte dessus.
- Votre fille va à l'université. Vous souhaitez lui envoyer un e-mail. 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 proxy. Je pense que les fonctionnalités UDP arriveront sous peu.

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

De plus, les serveurs proxy 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 firewall ni un serveur proxy. 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 serveur proxy sont utiles pour ces réseaux qui comportent de nombreuses machines se connectant au vol à InterNet, avec un setup et peu de travail ensuite.

9 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 proxy et des firewalls vous intéresse, lisez encore.

9.1 Un grand réseau avec sécurité renforcée

Disons, par exemple, que vous êtes le gourou de la secte de la 23ème Cabale de la Discorde du 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. 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 du Milwaukee, issue du texte initial, contrairement à ce qui a été modifié dans les versions plus récentes (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.

9.1.1 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.

9.1.2 La configuration proxy

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 proxy ici. Les réseaux des sages et des adeptes sont derrière des firewalls, il est donc nécessaire de leur configurer des serveurs proxy.

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 World Wide 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 proxy 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 !