

Serveurs "soho" sous Ubuntu Server

Déploiement et exploitation

Michel Blanc, netWorks <mblanc.networks@gmail.com>

Serveurs "soho" sous Ubuntu Server: Déploiement et exploitation

par Michel Blanc

\$Revision: 1.23 \$

Publié le \$Date: 2007/07/07 14:26:00 \$

Dédicace

Mille mercis à Margaux, Virginie, Corentin et Hugo qui ont supporté leur *boulllu* soir après soir avec patience tout en évitant le « DocBook : The Definitive Guide [<http://www.docbook.org/tdg/en/html/docbook.html>] » qui traversait souvent la pièce à basse altitude...

Table des matières

Introduction	viii
1. Ce qu'est ce document	viii
2. Ce que ce document n'est pas	viii
3. Audience	viii
4. Nouvelles versions de ce doc	ix
5. Revisions	ix
6. Contributions	ix
7. Feedback	ix
8. Copyright	ix
9. Prérequis	ix
10. Conventions utilisées dans ce document	ix
11. Organization de ce document	x
1. Principes généraux	1
1.1. Pourquoi ces principes ?	1
1.2. L'OSS est un avantage	1
1.3. La sécurité, partie intégrante	2
1.4. Privilégier la simplicité	2
1.5. Principe du privilège minimum et séparation des pouvoirs	2
1.6. Déployer le strict nécessaire	3
1.7. Procédures	3
1.8. Dimensionner les mesures de sécurité	3
2. Post-configuration du système d'exploitation	5
2.1. Modifier un fichier de configuration	5
2.2. Suppression des services et paquetages inutiles	6
2.2.1. Objectifs	6
2.2.2. Dénicher les services	6
2.2.3. Dénicher les paquetages	6
2.3. Ajustements système	8
2.3.1. Sysctl	8
2.3.2. IPv6	13
2.4. Configuration de la pile IP	14
2.4.1. Adressage	14
2.4.2. Résolution DNS	15
2.4.3. Filtrage de base	16
2.4.4. Modifier les règles	21
2.5. Intégrité des fichiers	21
2.5.1. Installation du serveur OSSEC	22
2.5.2. Installation d'un client OSSEC	22
3. Déploiement et guide des opérations OpenSSH	23
3.1. Qu'est ce qu'OpenSSH ?	23
3.2. Installation	23
3.3. Configuration	24
3.3.1. Port d'écoute	24
3.3.2. Paramètres	26
3.4. Authentification par clef publique	27
3.4.1. Création de clef client	27
3.4.2. Mise en place des clefs sur le serveur	28
3.4.3. Régénération des clefs sur le serveur	29
3.4.4. Modification du mot de passe d'une clef privée	31
4. Déploiement et guide des opérations Apache	32
4.1. Historique et description	32
4.2. Architecture	32
4.2.1. Modèles MPM	32
4.2.2. Modules	33
4.3. Gérer le service	34

4.3.1. Démarrage et arrêt	34
4.4. Filtrage	34
4.4.1. Filtrage	35
4.4.2. « Protection » contre les dénis de service (DoS)	35
4.5. Configuration	38
4.5.1. Fichiers de configuration	38
4.5.2. Configuration générale	38
4.5.3. Site par défaut et VirtualHosts	40
4.5.4. Création d'un VirtualHost	40
4.5.5. Contrôle d'accès	42
5. Déploiement et guide des opérations PHP	45
5.1. Installation	45
5.2. Configuration	47
5.2.1. Restrictions	47
5.2.2. Limites	48
5.2.3. Gestion d'erreurs	49
5.2.4. Exceptions	49
6. Déploiement et guide des opérations MySQL	51
6.1. Installation	51
6.2. Gérer le service	52
6.3. Notions de base	53
6.3.1. Fichier de configuration	53
6.3.2. Utilisateurs	53
6.3.3. Bases de données	53
6.3.4. Outils	54
6.3.5. SQL par l'exemple en 3 minutes	54
6.4. Configuration initiale	56
6.4.1. Utilisateurs	56
6.4.2. Bases de données	59
6.4.3. my.cnf	61
6.4.4. En finir avec l'historique	62
6.4.5. Filtrage	62
6.5. Gestion des droits	63
6.5.1. GRANT	63
6.5.2. REVOKE	64
6.5.3. Visualisation des droits	64
6.5.4. Droits utilisables avec GRANT et REVOKE	65
6.6. Perte des identifiants	65
6.7. Sauvegarde et restauration de bases	66
7. Déploiement et guide des opérations ProFTPD	70
7.1. Installation	70
7.2. Configuration	72
7.2.1. /etc/proftpd/modules.conf	72
7.2.2. /etc/proftpd/proftpd.conf	73
7.3. Filtrage	73
7.4. Gérer le service	76
7.4.1. Démarrage et arrêt	76
7.4.2. Supervision des connexions	76
7.4.3. Suspendre le service	76
7.4.4. Logs	77
8. Déploiement et guide des opérations Postfix	78
8.1. Installation	78
8.2. Configuration	81
8.2.1. Reconfiguration de base	81
8.2.2. Fichiers	82
8.2.3. Principales directives	83
8.2.4. Alias	85
8.2.5. Réécriture d'adresse	86

8.3. Gérer le service	87
8.3.1. Démarrage et arrêt	87
8.3.2. Gestion de la queue	87
8.4. Filtrage	88
9. Déploiement et guide des opérations Samba	90
9.1. Installation	90
9.2. Configuration	92
9.2.1. Les démons Samba	92
9.2.2. Démon permanent ou super-serveur xinetd	92
9.2.3. Choix d'architecture	93
9.2.4. /etc/samba/smb.conf	93
9.3. Filtrage	101
9.4. Gérer le service	103
9.4.1. Gestion des utilisateurs	104
10. Chiffrement SSL/TLS	106
10.1. Généralités	106
10.1.1. Problématique	106
10.1.2. Architecture	106
10.2. Préparatifs	107
10.2.1. Le paquetage OpenSSL	107
10.2.2. Création d'une autorité de certification	108
10.2.3. Création d'un certificat serveur	109
10.2.4. Liste de révocation	111
10.3. Déploiement	112
10.3.1. HTTPS	112
10.3.2. FTP/TLS	115
10.3.3. MySQL/SSL	116
10.3.4. SMTP/TLS	117
11. Déploiement et guide des opérations OpenVPN	118
11.1. Installation	118
11.2. Configuration	119
11.2.1. Génération des paramètres Diffie-Hellman	120
11.2.2. Génération de la requête de signature de certificat client	120
11.2.3. Signature de la demande de signature	121
11.2.4. Fichier de configuration serveur	122
11.2.5. Fichier de configuration client	123
11.2.6. Test de la configuration	123
11.2.7. Autres paramètres	124
11.2.8. Démarrage au boot	125
11.3. Filtrage	126
A. Firewall de base	127
B. GNU Free Documentation License	131
B.1. PREAMBLE	131
B.2. APPLICABILITY AND DEFINITIONS	131
B.3. VERBATIM COPYING	132
B.4. COPYING IN QUANTITY	133
B.5. MODIFICATIONS	133
B.6. COMBINING DOCUMENTS	134
B.7. COLLECTIONS OF DOCUMENTS	135
B.8. AGGREGATION WITH INDEPENDENT WORKS	135
B.9. TRANSLATION	135
B.10. TERMINATION	135
B.11. FUTURE REVISIONS OF THIS LICENSE	136
B.12. ADDENDUM: How to use this License for your documents	136
Glossary	137
Bibliographie	140

Liste des illustrations

2.1. Premier boot	5
2.2. Trafic <i>egress</i> et <i>ingress</i>	16
5.1. PHP Info	46
7.1. Choix d'installation	71
7.2. Mode FTP passif	75
7.3. Mode FTP actif	75
7.4. ftptop	76
8.1. Postfix: Choix du type d'installation	79
8.2. Postfix: Choix du nom du serveur	79
8.3. Postfix: Choix du relais SMTP	80
9.1. Vue du serveur dans le « Voisinage réseau »	95
10.1. Test HTTPS	115
11.1. Principe de fonctionnement d'une PKI	119

Liste des tableaux

1. Conventions typographiques	x
4.1. Apache : détermination de la restriction	43
6.1. Exemple SQL : Table des Génies	54
9.1. Samba : principales variables de substitution	101

Liste des exemples

4.1. Apache: configuration du filtrage TCP en entrée	35
4.2. Apache : Limitation de connexions avec ipt_limit	36
4.3. Apache : Limitation de connexions avec ipt_recent	37
6.1. MySQL : configuration du filtrage TCP en entrée	62
8.1. Postfix : configuration du filtrage TCP en entrée	88
8.2. Postfix : configuration du filtrage TCP en sortie	89
9.1. Samba : configuration du filtrage UDP en entrée et en sortie	102
9.2. Samba : configuration du filtrage TCP en entrée	103
9.3. Samba : configuration du filtrage TCP en sortie	103
10.1. Apache HTTPS: configuration du filtrage TCP en entrée	114
11.1. OpenVPN : configuration du filtrage UDP en entrée	126

Introduction

1. Ce qu'est ce document

Ce document détaille les différentes étapes nécessaires à la mise en place de serveur type SOHO sous Ubuntu linux. Par SOHO, j'entends un serveur de fichiers, un serveur web, quelques applications relais (proxy cache, serveur mail sortant, cache DNS ¹) et une passerelle Internet. Ce document décrira les premiers pas à suivre afin de :

- sécuriser une installation par défaut Ubuntu
- déployer une passerelle internet
- déployer un serveur *LAMP* + FTP
- déployer un serveur Samba
- installer une petite *PKI* et déployer un serveur OpenVPN

La plupart des opérations d'exploitation courantes seront détaillées pour chaque service.

Lors de l'installation et de la configuration de chaque élément, l'impact sur la sécurité globale sera évoqué. Parallèlement, des mesures seront mises en place à chaque étape afin de conserver un niveau de sécurité maximum pour l'installation.

Ces serveurs peuvent être ou non déployés sur une seule machine. Suivant la fonction du serveur, il sera plus raisonnable de ne pas l'exposer directement sur Internet et d'en faire, si possible, un serveur spécifique à placer derrière un pare-feu.

2. Ce que ce document n'est pas

Ce document n'est pas un guide d'administration système Linux, pas plus qu'un manuel de configuration des différents services évoqués. Il existe d'excellents documents décrivant ces aspects. Le lecteur intéressé par la sécurité sera probablement comblé par le [SecDebian].

Comme largement expliqué par ailleurs, la sécurité n'est pas un produit, et il n'y a pas de potion magique permettant de rendre un système sûr. En revanche, il existe des bonnes pratiques que nous essaierons de mettre en œuvre le long de ce document. Evidemment, ce document ne peut être exhaustif en la matière; il faudra donc garder les yeux et les oreilles ouverts en ce qui concerne la sécurité.

3. Audience

Ce document est principalement destiné :

- aux admins système débutants qui cherchent un guide des opérations pour ce type de serveur
- aux sysadmins habitués, qui se disent depuis des années qu'ils devraient se faire une checklist afin de ne rien oublier lorsqu'ils déploient un serveur; une telle checklist peut être facilement déduite de ce document et un squelette de règles est donnée dans « Annexe A, *Firewall de base* ».
- aux wizards de la sécurité et des différents services évoqués sous Linux qui ne manqueront pas d'envoyer des commentaires ou de modifier directement ce document afin de l'améliorer

¹Tous ces services ne seront pas abordés dans ce document

4. Nouvelles versions de ce doc

Les nouvelles versions de ce document seront disponibles sur reseau.erasme.org [<http://reseau.erasme.org>].

Ce document étant réalisé sous DocBook, il est disponible au formats XML, PDF, PS, HTML, man, rtf, tex, texte, ...

5. Revisions

Historique des versions		
Version 1.1	2007-05-01	MB
Création.		
Version 1.0	2002-12-29	MG
Initial release for TLDP		

6. Contributions

Aucune pour l'instant. Les vôtres seront les bienvenues.

7. Feedback

Toutes les remarques, contributions, corrections sont les bienvenues:

`<mblanc.networks@gmail.com>`

8. Copyright

Copyright © 2007 Michel Blanc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts and no Back-Cover Texts. A copy of the license is included in Annexe B, *GNU Free Documentation License* entitled « GNU Free Documentation License ».

Read The GNU Manifesto [<http://www.fsf.org/gnu/manifesto.html>] if you want to know why this license was chosen for this book.

The author and publisher have made every effort in the preparation of this book to ensure the accuracy of the information. However, the information contained in this book is offered without warranty, either express or implied. Neither the author nor the publisher nor any dealer or distributor will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

The logos, trademarks and symbols used in this book are the properties of their respective owners.

9. Prérequis

Le lecteur gagnera à être familier avec les ligne de commande. Même si des outils graphiques très puissants pour gérer des services existent (SWAT, Webmin [<http://www.webmin.com/>]), on privilégiera le « contact direct » avec les fichiers de configuration. Une certaine aisance avec la manipulation de fichiers et la ligne de commande est donc souhaitable.

10. Conventions utilisées dans ce document

Les conventions typographiques suivantes sont utilisées tout au long de ce document :

Tableau 1. Conventions typographiques

Type de texte	Signification
« Texte cité »	Citations, texte cité tel quel, expressions
<code>terminal</code>	Sortie sur un terminal.
saisie	Saisie utilisateur sur un terminal.
commande	Nom d'une commande qui peut être utilisée en ligne de commande. Eventuellement nom d'un démon.
VARIABLE	Nom d'une variable, comme dans \$VARIABLE.
option	Option d'une commande, comme dans « l'option -a de la commande ls ».
<i>argument</i>	Argument d'une commande, comme dans « voir man ls ».
<code>commande options arguments</code>	Synopsys ou usage général d'une commande.
nom de fichier	Nom d'un fichier ou d'un répertoire, par exemple « Aller dans le répertoire /usr/bin. »
Touche	Touches du clavier, comme par exemple « tapez Q pour quitter ».
Bouton	Bouton graphique à cliquer, comme le bouton OK. Mais bon, les boutons, ce n'est pas la spécialité maison :)
Menu # Choix	Choix à sélectionner dans un menu graphique, par exemple : « Sélectionnez Aide # A propos de Firefox dans votre navigateur. » Même remarque que précédemment...
<i>Terminologie</i>	Terme ou concept important. Terme étranger. « Le <i>kernel</i> Linux »
Voir Chapitre 1, <i>Principes généraux</i>	Lien interne au document.
Kernel.org [http://kernel.org]	Lien cliquable vers un site ou une ressource externe.

11. Organization de ce document

Liste de chapitres et appendices :

- Chapitre 1, *Principes généraux*
- Chapitre 2, *Post-configuration du système d'exploitation*
- Chapitre 3, *Déploiement et guide des opérations OpenSSH*
- Chapitre 4, *Déploiement et guide des opérations Apache*
- Chapitre 5, *Déploiement et guide des opérations PHP*
- Chapitre 6, *Déploiement et guide des opérations MySQL*
- Chapitre 7, *Déploiement et guide des opérations ProFTPD*
- Chapitre 8, *Déploiement et guide des opérations Postfix*
- Chapitre 9, *Déploiement et guide des opérations Samba*

- Chapitre 10, *Chiffrement SSL/TLS*
- Chapitre 11, *Déploiement et guide des opérations OpenVPN*
- Annexe A, *Firewall de base*
- Annexe B, *GNU Free Documentation License (en VOST)*
- Glossary
- Bibliographie

Chapitre 1. Principes généraux

\$Revision: 1.11 \$

\$Date: 2007/07/03 17:00:14 \$

Aujourd'hui, le foisonnement d'outils, souvent disponibles en OpenSource, permet de déployer rapidement et relativement facilement un grand nombre de services internet. Cette facilité ne doit pas faire oublier les dangers inhérents à la mise en ligne d'un service quel qu'il soit. On devra donc accompagner ces déploiements avec la rigueur nécessaire afin que le service ne se transforme en vecteur de nuisance, aussi bien pour sa propre infrastructure que pour celle des autres.

1.1. Pourquoi ces principes ?

Déployer un serveur n'est généralement pas anodin. Par définition, un serveur fournit un service à un ou plusieurs utilisateurs. La rupture, le détournement, le vol ou la dégradation de ce service ont donc forcément des conséquences.

Sans entrer dans une paranoïa excessive, il convient de garder à l'esprit un certain nombre de grands principes. Même s'ils peuvent parfois paraître démesurés, ils sont tout de même applicables quelle que soit la taille du système d'information déployé. Si l'on garde à l'esprit ces principes pour encadrer le déploiement et la gestion d'un serveur, on réduit considérablement ses possibilités de défaillance. Les *bonnes pratiques* en administration système découlent de ces quelques principes. Ils ne sont pas exhaustifs. La sécurité est un processus; ce n'est pas un état du système. Les menaces et le système d'information évoluent. La « sécurité » doit suivre.

1.2. L'OSS est un avantage

Des discussions sans fin opposent régulièrement les spécialistes en sécurité afin de déterminer si les logiciels Open Source (OSS, Open Source Software) apportent un « plus » par rapport aux autres logiciels. La situation en ce domaine est résolument en faveur de l'OSS.

Tout d'abord, du code OSS c'est du code librement disponible : n'importe qui peut avoir accès au code source. Cela a de multiples implications. Les auteurs sont souvent plus soucieux d'écrire du code lisible. Non que les logiciels commerciaux soient mal codés, mais les logiciels OSS ne sont pas soumis à la pression commerciale, aux dates de release, etc... La lisibilité qui en résulte est probablement plus grande.

Un code disponible, c'est aussi un code qui peut être *audité*. Le projet OpenBSD [<http://www.openbsd.org>] a une équipe dédiée qui, depuis plus de 10 ans, audite le code des applications utilisées par OpenBSD. Tous ces développements s'effectuent donc en public, sous le regard des pairs, ce qui ne peut qu'améliorer la qualité du code produit. Bien sûr, dans la mesure où la disponibilité du code source permet d'identifier plus facilement les défauts, il peut aussi permettre aux personnes malveillantes de repérer des failles directement dans le code. Mais de ce point de vue, l'historique des failles de sécurité montre que « l'obscurité » et la non divulgation du code source n'ont visiblement jamais été des solutions à ce problème.

Bruce Schneier [<http://www.schneier.com>], cryptographe éminent et spécialiste incontesté de la sécurité informatique l'a dit lui-même : « Demand open source code for anything related to security » [Schneier1999]¹.

Enfin, même si l'OSS représente un plus important en terme de sécurité, ce n'est *absolument pas* une condition suffisante qui à elle seule peut garantir la sécurité d'un système d'information quel qu'il soit. Le pire ennemi de la sécurité est le *sentiment* de sécurité.

¹« Exigez du code open source pour tout ce qui touche à la sécurité. »

1.3. La sécurité, partie intégrante

La sécurité doit être considérée comme faisant partie intégrante de chaque élément du système. Chaque service, chaque brique logicielle, chaque choix configuration ont un impact sur la sécurité globale du système. C'est donc un élément constituant de toutes les parties. Il faut le traiter ainsi, et non comme une couche supplémentaire du système. La sécurité, c'est apache bien configuré sur un système bien configuré avec un firewall implémenté correctement. Ce n'est pas apache + firewall.

Exprimé autrement, la sécurité n'est pas un logiciel, ce n'est pas la dernière *appliance* à la mode achetée à grands frais². C'est un ensemble de bonnes pratiques que l'on applique à l'ensemble des éléments du système. Vu ainsi, lorsque la sécurité sera appréhendée comme étant partie de chaque élément, le système sera beaucoup plus robuste, plus résilient et l'impact d'une attaque réussie sera moins important. C'est le fondement de la *sécurité par/en couches*, qui ne cherche pas à supprimer complètement la menace (c'est impossible), mais à ralentir la progression de l'attaquant et réduire les conséquences de son attaque (concept de « défense en profondeur [http://en.wikipedia.org/wiki/Defense_in_depth] »).

1.4. Privilégier la simplicité

La simplicité doit guider toutes les modifications et déploiements sur un serveur. Le principe de simplicité, baptisé *KISS* (Keep It Simple, Stupid) pourrait être paraphrasé en « la solution la plus simple est la meilleure. », ou, selon les termes d'Albert Einstein, « tout devrait être conçu le plus simplement possible, mais pas plus. ».

Il est en effet facile de concevoir que les usines à gaz seront plus sujettes aux fuites que des systèmes simples. L'enchevêtrement de composants, de services, de protocoles nuit à la compréhension globale du système. Et en l'absence de compréhension, impossible d'exploiter le système dans des conditions satisfaisantes de sécurité. En cas de défaillance du système, comment identifier le composant responsable ? Et quel sont les effets de bord à envisager si le composant X tombe en panne ? Ou si le démon Y est compromis ? Ou encore si je mets à jour la librairie Z ? Ou bien si je migre vers la dernière version du système d'exploitation ?

Il faudra donc, autant que possible, spécialiser les différents serveurs afin de ne pas créer de « serveur à tout faire » qui, lorsqu'il sera en panne, impactera un nombre important de services, ou qu'il sera impossible de faire évoluer. L'utilisation de la virtualisation peut aider à résoudre ce problème, même si cette technologie n'est pas encore bien appréhendée en ce qui concerne la sécurité.

1.5. Principe du privilège minimum et séparation des pouvoirs

Lorsqu'un démon se voit compromis, par exemple, par un débordement de pile [http://fr.wikipedia.org/wiki/Buffer_overflow], l'attaquant peut exécuter du code (des ordres) avec les droits de ce démon. Si ce serveur fonctionne avec les droits de l'utilisateur *root*, le code « injecté » par l'attaquant aura lui aussi les droits de *root*. On comprend donc l'intérêt d'exécuter ce logiciel (par exemple, le serveur web *apache*) avec des droits minimum. En général, lorsqu'un logiciel type démon est installé sur un serveur, un utilisateur spécial est créé à cet effet afin de limiter la casse en cas d'intrusion.

Ce principe de *privilège minimum* s'applique aussi aux utilisateurs. Il faut restreindre les droits affectés aux utilisateurs en fonction de leur besoin. Ce n'est pas uniquement une question de *confiance* : le compte de l'utilisateur peut être compromis, l'utilisateur peut ne pas avoir conscience des problèmes liés à la sécurité, etc... Il faut donc là aussi fournir le minimum de privilèges à l'utilisateur afin d'éviter une catastrophe démesurée par rapport aux droits réellement requis pour cet utilisateur.

²et généralement bâtie avec des logiciels Open Source.

A titre d'exemple, imaginons qu'un utilisateur doit envoyer des fichiers sur un serveur web. Si l'administrateur du serveur se contente de créer un compte à cet utilisateur, il possèdera par défaut un *shell*. Ce shell pourra éventuellement être utilisé via ssh, ou au travers d'une application Web mal programmée, ou en local sur la console. Si le seul besoin est d'envoyer des fichiers, l'utilisateur ne doit pouvoir faire que cela.

Privilège minimum

Appliquer le principe du privilège minimum, aussi bien aux applications qu'aux utilisateurs

Le corollaire au privilège minimum est la séparation des pouvoirs. A l'instar des pouvoirs législatifs et exécutifs que nous connaissons ailleurs, une machine ne doit pas offrir des services, qui, cumulés, lui donneraient un pouvoir trop important sur l'activité du site. C'est une question de résilience du système d'information, mais c'est surtout un prérequis permettant d'éviter une éventuelle escalade de privilèges.

Il est à déconseiller, par exemple, d'installer une autorité de certification (CA)³ sur une machine fournissant un autre service. Idem pour les serveurs DNS, LDAP, SQL, qui contiennent ou servent souvent des éléments utilisés dans la sécurité d'autres machines (authentification, noms d'hôtes, ...) revêtant ainsi une importance toute particulière. Il en va de même pour les utilisateurs à pouvoir : il faut séparer les pouvoirs afin d'avoir des gardes fous sur l'utilisation qui est faite du système d'information. Un chef d'entreprise ne devrait pas, par exemple, gérer la messagerie de ses employés, afin d'éviter toute tentation offerte par l'accès aux boîtes.

Séparation des pouvoirs

Appliquer la séparation des pouvoirs, aussi bien aux machines qu'à leurs administrateurs

1.6. Déployer le strict nécessaire

Un des nombreux avantages des systèmes GNU/Linux est de pouvoir sélectionner la quasi totalité des composants installés. Bien sûr, certains de ces composants sont obligatoires (le *noyau*, la *glibc*), mais dans la grande majorité des cas il est possible d'influer sur l'installation ou non d'un composant, de lui préférer un équivalent ou une autre version. Cette souplesse doit être mise à profit en n'installant que le strict nécessaire sur la machine. Pourquoi installer un serveur web si l'on n'a pas besoin de serveur web ? Pourquoi démarrer un serveur web si l'on en a qu'un besoin occasionnel ? Pourquoi installer un compilateur C si l'on a rien à compiler ? En résumé, « Small is beautiful » : il ne faut pas installer ce qui est inutile afin d'éviter que ce composant inutile ne se transforme en vecteur permettant de compromettre la machine.

1.7. Procédures

Les incidents arrivent. Quels que soient les investissements réalisés pour sécuriser son système d'information, il faudra un jour restaurer un serveur, fonctionner en mode dégradé, gérer une intrusion, etc... A cet instant, rien ne sera plus inefficace que l'improvisation.

Il faut donc, avant d'en avoir besoin, développer des procédures qui seront appliquées le moment venu. Ces procédures devront être régulièrement testées (que le backup marche, c'est bien; si la restauration fonctionne, c'est mieux...), évaluées, et remises à jour.

1.8. Dimensionner les mesures de sécurité

Sécuriser un système d'information est une quête sans fin. La question qui se pose en général est : jusqu'où aller ? Afin de pouvoir répondre, il est nécessaire d'évaluer le système d'information : Quel est mon bien ? Quel est sa valeur ? Quels coûts seront engendrés par un acte malveillant ?

³ il est d'ailleurs fortement déconseillé de mettre une machine servant de CA en réseau

Ces coûts peuvent être directs et directement quantifiables (équipements, coûts de réinstallation d'un serveur compromis, coûts d'une action légale, ...) ou indirects, liés à des biens immatériels (coûts d'image, vol de secrets, ...).

En regard de ces différents coûts, il faudra dimensionner l'infrastructure de sécurité. Inutile, à priori, de déployer des milliers d'euros en équipements de protection si vous devez uniquement protéger votre accès internet. En revanche, quand il s'agit d'un réseau plus important et de quelques serveurs, il faut peut être envisager la chose.

Chapitre 2. Post-configuration du système d'exploitation

\$Revision: 1.24 \$

\$Date: 2007/07/05 01:18:23 \$

L'installation du système d'exploitation n'est pas couverte dans ce document. En revanche, les choix par défaut effectués par les distributions Linux sont rarement parfaits. Aussi faut-il généralement faire un peu de nettoyage et ajuster quelques paramètres juste après l'installation. Nous partirons de l'hypothèse que l'utilisateur ayant les droits d'administration est `oper` et que le nom du serveur est `ubuntu`. D'autre part, il est fortement conseillé d'effectuer l'installation :

- *sans* réseau, et de ne pas le brancher avant d'avoir configuré le filtrage (Section 2.4.3, « Filtrage de base ») afin d'éviter tout risque,
- *sans* sélectionner de paquetage particulier (notamment « LAMP » ou « Serveur DNS »).

Figure 2.1. Premier boot

```
* Activating swap... [ OK ]
* Checking root file system...
fsck 1.40-WIP (14-Nov-2006)
/dev/sda1: clean, 21926/243360 files, 97844/485958 blocks
[ OK ]
* Checking file systems...
fsck 1.40-WIP (14-Nov-2006)
[ OK ]
* Mounting local filesystems... [ OK ]
* Activating swapfile swap... [ OK ]
* Configuring network interfaces... [ OK ]
* Setting up console font and keymap... [ OK ]
* Starting system log daemon... [ OK ]
* Starting kernel log... [ OK ]
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Running local boot scripts (/etc/rc.local) [ OK ]
Ubuntu 7.04 ubuntu tty1
ubuntu login: _
```

Dans ce chapitre, nous évoquerons les sujets suivants :

- Suppression des services et des paquetages inutiles
- Ajustements système
- Configuration de la pile IP
- Mise en place d'OSSEC (TBD)

2.1. Modifier un fichier de configuration

Avant de se lancer dans la modification du système installé, il est important de connaître au moins cette bonne pratique essentielle : il est impératif, avant tout changement sur un fichier de configuration, d'en faire une copie *sur place*. La raison est simple et évidente : quand on aura copieusement modifié le fichier de configuration et que plus rien ne fonctionnera correctement, on sera bien heureux de pouvoir

récupérer la situation par un simple `cp`. On pourra même raffiner un peu la méthode en copiant le fichier original en `fichieroriginal_YYYYMMDDNN` (ou YYYY représente l'année, MM le mois, YY le jour et NN un numéro de série) avant de l'éditer. Cela permettra de revenir à une ancienne configuration si l'on s'aperçoit du problème plus tard. On veillera à changer les droits de ces fichiers afin que seul `root` y ait accès (`chmod 600 ...`).

2.2. Suppression des services et paquetages inutiles

2.2.1. Objectifs

Une machine ayant vocation à offrir un service donné ne devrait contenir que les éléments logiciels nécessaires à la fourniture de ce service. Quel est l'intérêt d'avoir le compilateur `gcc` sur un serveur Web ? Pourquoi démarrer `portmap` si aucun service n'a besoin du portmapper ?

Effectuer un nettoyage en règle du système ne permet pas seulement un gain de place. Les réels intérêts sont ailleurs. Supprimer les services inutiles c'est d'abord supprimer le risque que la machine soit *compromise* grâce à ce service. Même si le service exécuté est exempt de bugs et de failles de sécurité le jour J, il ne le sera peut être pas le jour J+1. Grâce au « banner fingerprinting » (détermination de logiciel et de version grâce aux bannières), les délinquants informatiques scannent massivement des blocs d'adresses à la découverte des services situés sur les machines. Ces services sont ensuite *identifiés* si possible grâce à leur bannière d'accueil. Ils peuvent ainsi se constituer une base de données des logiciels/versions en fonctionnement sur une des machines données. Si, le jour J+1, une vulnérabilité exploitable est découverte dans version X du logiciel L, il ne reste qu'à rechercher dans la base de données la liste des machines immédiatement vulnérables. Ces machines n'ont alors aucune chance de mettre en place un correctif avant d'être attaqués.

Enfin, supprimer les services inutiles permet une économie de stockage, de RAM et de CPU.

Même si sur les distributions Ubuntu serveur sont très propres à l'installation, il convient de savoir effectuer ce nettoyage. Ce pourra être utile sur d'autres distributions, ou sur des version d'Ubuntu serveur peut être moins abouties.

2.2.2. Dénicher les services

Lorsque l'on cherche des services inutiles sur un système, il est souvent plus simple de regarder quelles sont les socket ouvertes en écoute. Cela ne garantit pas de trouver tous les services (dæmons) qui tournent sur la machine (tous n'ont pas vocation à écouter sur des ports) mais permet d'éliminer les plus importants en premier lieu. On pourra s'attacher à éliminer le reste ensuite.

`netstat` permet d'afficher la liste des sockets ouvertes sur le système. Nous ne sommes intéressés que par les socket TCP en écoute (`LISTEN`) et par les sockets UDP. La sortie de `netstat` sera donc filtrée par un `egrep "LISTEN|udp"` :

```
oper@ubuntu:~$ sudo netstat -tunap | egrep "LISTEN|udp"
udp      0      0 0.0.0.0:69          0.0.0.0:*        3745/dhclient3
```

```
oper@ubuntu:~$
```

Le seul démon (réseau) en cours de fonctionnement semble être le client dhcp. Lorsque l'adressage IP sera correctement configuré (Section 2.4.1, « Adressage »), ce démon ne sera plus démarré au boot.

2.2.3. Dénicher les paquetages

Sous Ubuntu serveur, seule une poignée de packages peut être supprimée. Ces packages sont installés grâce au « méta-package » `ubuntu-standard`. Un méta-package est une package « virtuel » vide, qui, grâce à ses dépendances, implique l'installation (ou la désinstallation) d'autres packages.

Tous les paquetages dont dépend `ubuntu-standard` sont optionnels. en désinstallant l'un d'entre eux, on forcera aussi la désinstallation du package `ubuntu-standard`. Réinstaller ce dernier forcera la réinstallation de toutes ses dépendances.

Pour connaître les paquets installés comme dépendance d' `ubuntu-standard`, on peut faire appel à la commande **`apt-cache show ubuntu-standard`** :

```
oper@ubuntu:~$ apt-cache show ubuntu-standard
Package: ubuntu-standard
Priority: standard
Section: metapackages
Installed-Size: 44
Maintainer: Matt Zimmerman <mdz@ubuntu.com>
Architecture: i386
Source: ubuntu-meta
Version: 1.43
Depends: at, cpio, cron, dmidcode, dnsutils, dosfstools,
  dselect, ed, fdutils, file, ftp, hdparm, info, inputattach,
  iptables, iputils-arping, iputils-tracepath, logrotate, lshw,
  lsof, ltrace, man-db, manpages, memtest86+, mime-support, nano,
  parted, popularity-contest, ppp, pppconfig, pppoeconf, psmisc,
  reiserfsprogs, rsync, strace, tcpdump, telnet, time, w3m, wget
Recommends: command-not-found, mtr-tiny, openssh-client,
  update-manager-core
Filename: pool/main/u/ubuntu-meta/ubuntu-standard_1.43_i386.deb
Size: 17158
MD5Sum: 4b9cfdce3972c9d6e4752d1fc4bd42ff
Description: The Ubuntu standard system
  This package depends on all of the packages in the Ubuntu standard
  system.
  This set of packages provides a comfortable command-line Unix-like
  environment.
.
  It is also used to help ensure proper upgrades, so it is
  recommended that
  it not be removed.
Bugs: mailto:ubuntu-users@lists.ubuntu.com
Origin: Ubuntu
Task: ubuntu-standard
```

```
oper@ubuntu:~$
```

Cette commande renvoie, à la suite du paramètre `Depends :`, une liste des paquetages *dont dépend* `ubuntu-standard` et qui sont supprimables.

Par exemple, si le serveur n'a pas vocation à utiliser `ppp` ou `pppoe`, il est possible de supprimer les packages `ppp`, `pppconfig` et `pppoeconf`.

```
oper@ubuntu:~$ sudo apt-get remove --purge ppp pppconfig pppoeconf
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les paquets suivants seront ENLEVÉS :
  ppp* pppconfig* pppoeconf* ubuntu-standard*
0 mis à jour, 0 nouvellement installés, 4 à enlever et 4 non mis à
jour.
Il est nécessaire de prendre 0o dans les archives.
Après dépaquetage, 2339ko d'espace disque seront libérés.
Souhaitez-vous continuer [O/n] ? O
```

```
(Lecture de la base de données... 14372 fichiers et répertoires
déjà installés.)
Suppression de ubuntu-standard ...
Suppression de pppoeconf ...
Purge des fichiers de configuration de pppoeconf ...
Suppression de pppconfig ...
Purge des fichiers de configuration de pppconfig ...
rmdir: /var/cache/pppconfig: No such file or directory
Suppression de ppp ...
Stopping all PPP connections...done.
Purge des fichiers de configuration de ppp ...
oper@ubuntu:~$
```

On remarquera, au passage, que le package `ubuntu-standard` est désinstallé comme prévu.

2.3. Ajustements système

2.3.1. Sysctl

`sysctl` permet de modifier le comportement du système en ajustant certains paramètres du noyau. Ces paramètres sont disponibles :

- soit sous `/proc/sys/`
- soit grâce à l'outil `sysctl`

A titre d'exemple, mettre le paramètre `ip_forward` à la valeur 1 peut se faire de deux manières :

- via `sysctl` : `sysctl -w net.ipv4.ip_forward=1`
- en modifiant le fichier : `echo 1 > /proc/sys/net/ipv4/ip_forward`

Pour que ces paramètres soient appliqués au démarrage du système, on met généralement les valeurs à configurer dans le fichier `/etc/sysctl.conf`.

`sysctl` permet aussi d'afficher la valeur d'un paramètre actuellement en vigueur (`sysctl param`), voire la totalité des paramètres (`sysctl -a`).

```
oper@ubuntu:~$ sudo sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
oper@ubuntu:~$ sudo sysctl -a
...
net.ipv4.ip_forward = 0
...
oper@ubuntu:~$
```

Certaines valeurs sont parfois disponibles sous les variables `all`, `default`, ou un nom d'interface. Par exemple, `rp_filter` existe dans les variables `net.ipv4.conf.all.rp_filter`, `net.ipv4.conf.default.rp_filter`, `net.ipv4.conf.eth0.rp_filter`, etc... `all` permet de fixer la valeur pour toutes les interfaces. Si un interface reçoit une valeur explicite via, par exemple, `net.ipv4.conf.eth0.xyz`, c'est cette dernière qui sera utilisée. `default` permet de définir une valeur par défaut qui sera utilisée si une nouvelle interface est créée dynamiquement (carte PCMCIA ou dispositif USB par exemple).

D'une manière générale, il conviendra, lorsque une valeur est donnée pour `all`, d'appliquer la même valeur à `default`.

Quelques optimisations intéressantes sont décrites ci-dessous. Même si la valeur par défaut est donnée pour chacune d'entre elles, il vaut mieux ne pas présumer des valeurs par défaut et renseigner systématiquement `/etc/sysctl.conf`.

2.3.1.1. SYN protection

Pour établir une connexion TCP, les deux parties doivent passer par le « three-way handshake » : le client envoie un paquet TCP vide avec le drapeau SYN, le serveur répond avec un paquet contenant les flags ACK (pour acquitter le premier SYN) et SYN (son propre drapeau de SYNchronisation). Le client achève l'établissement de la connexion en envoyant le troisième et dernier paquet contenant simplement un acquittement (ACK) du SYN serveur.

Tout cela fonctionne bien tant que le processus d'établissement de connexion va à son terme. Si le « 3-way handshake » n'est pas complété, la connexion reste à moitié ouverte. Aux débuts de TCP/IP, ce n'était pas perçu comme un problème : il n'y avait pas de malveillance. La plupart des piles TCP/IP se contentaient alors d'avoir une queue de SYN (« SYN backlog ») de quelques entrées, généralement autour de 8. En clair, le système ne pouvait maintenir que 8 connexions en cours d'établissement.

Il est apparu rapidement que cette manière de fonctionner était problématique. Effectuer un déni de service était particulièrement trivial en remplissant le « backlog » de connexions à moitié ouvertes (« SYN flooding »).

Pour résoudre ce problème, plusieurs techniques sont utilisées sous Linux. Par défaut, Linux utilise une FIFO (« SYN backlog »). Lorsque la file de connexions en attente est pleine, les plus anciennes sont supprimées. Si la réponse (ACK) du client arrive après que la demi-connexion ait été supprimée, un paquet RST sera émis et la connexion sera (normalement) retentée par le client. Le backlog sous Ubuntu est de 1024 entrées.

L'autre possibilité est d'activer les « syncookies ». Les « syncookies » désignent un mécanisme permettant de générer un numéro de séquence initial (*ISN*) dans la réponse au client (SYN+ACK). Cet ISN est construit de telle sorte qu'il n'est pas nécessaire de conserver l'information du SYN initial afin de valider que le dernier paquet du 3-way handshake est correct (voir [SynDJB] et [SynWiki]).

Beaucoup d'encre et de sang (voir « SYN cookie monsters » dans [SynWiki]) ont coulé autour des syncookies, en particulier concernant leur utilisation sur des serveurs chargés. Cependant, il vaut probablement mieux les activer en toutes circonstances. A part dans des cas de congestion extrême, il sont probablement la meilleure réponse au problème du « syn flooding ».

Les syncookies sont pilotés par la variable `net.ipv4.tcp_syncookies`. Elle est à 0 par défaut, il faut donc ajouter la ligne suivante dans `/etc/sysctl.conf`.

```
net.ipv4.tcp_syncookies = 1
```

2.3.1.2. ICMP echo

Il est possible d'empêcher la pile IP de répondre aux ICMP echo requests. Il est probablement préférable d'utiliser **netfilter** pour *limiter* le volume d'icmp echo requests (c'est ce qui sera implémenté dans la section dédiée au filtrage, Section 2.4.2, « Résolution DNS »). En effet Supprimer complètement l'émission d'icmp echo replies viole la RFC 792 (voir [RFC792]) :

3.2.2.6 Echo Request/Reply

Every host MUST implement an ICMP Echo server function that receives Echo Requests and sends corresponding Echo Replies.

ainsi que la RFC 1812 ([RFC1812]) au choix, selon que `net.ipv4.ip_forward` soit à 1 ou non :

4.3.3.6 Echo Request/Reply

A router MUST implement an ICMP Echo server function that receives Echo Requests sent to the router, and sends corresponding Echo Replies.

Par ailleurs, cela n'apporte absolument *rien* en terme de sécurité. L'époque des « ping de la mort » est révolue. Ignorer les paquets ICMP ne règle pas les problèmes de saturation réseau, de pile IP bogguée ou de trou de la couche d'ozone... Enfin, se cacher ne résoud pas les problèmes de sécurité... Si malgré tout, le blocage complet est requis, il suffit de mettre la variable `net.ipv4.icmp_echo_ignore_all` à la valeur 1.



Variable `net.ipv4.icmp_echo_ignore_all`

Mettre `net.ipv4.icmp_echo_ignore_all` à 1 viole les RFC 792 et 1812 ! Il est très fortement conseillé de recourir au rate limiting ICMP à la place.

2.3.1.3. ICMP redirects

Les paquets ICMP redirects permettent aux routeurs de prévenir des hôtes qu'une meilleure route est disponible et qu'ils peuvent l'utiliser. Un paquet ICMP redirect contient la nouvelle gateway que l'hôte doit utiliser.

Sur un réseau bien configuré, l'ICMP redirect est inutile. De plus, l'utilisation de l'ICMP redirect pourrait permettre d'injecter des routes dans les tables de routages des hôtes. Il est donc conseillé de supprimer l'acceptation (`net.ipv4.conf.all.accept_redirects`) et l'émission (`net.ipv4.conf.all.send_redirects`) de paquets ICMP redirect.

Enfin, la variable `net.ipv4.conf.all.secure_redirects`, lorsqu'elle est à 1, permet d'accepter les ICMP redirect depuis une gateway listée comme telle. Il est aussi conseillé de passer cette valeur à 0. Attention : la configuration par défaut fait exactement l'inverse pour toutes ces variables.

2.3.1.4. ICMP echo broadcasts

Les paquets ICMP echo request envoyés en broadcast permettent de facilement « pinguer » toutes les machines d'un sous réseau. Même si cette fonctionnalité (qui tient tout de même de l'effet de bord) peut sembler pratique, il est déconseillé de l'activer. L'effet d'amplification induit peut poser des problèmes de déni de service « smurf attacks » (voir http://en.wikipedia.org/wiki/Smurf_attack). Il est donc préférable de mettre la variable `net.ipv4.icmp_echo_ignore_broadcasts` à 1.

2.3.1.5. Bogus ICMP

Le noyau, par défaut, loggue les paquets ICMP malformés (code d'erreur inconnu). Logguer ces paquets induit une écriture sur le disque. Cela peut permettre à un attaquant d'effectuer un déni de service en remplissant les logs de messages

```
kernel: XX.XX.XX.XX sent an invalid ICMP type T, code C...
```

L'accumulation de ces messages peut finir par remplir le filesystem et accueillant les logs. Il est donc préférable de simplement ignorer ces erreurs en affectant 1 à `net.ipv4.icmp_ignore_bogus_error_responses`. Si nécessaire, on pourra traiter ces messages avec d'autres applications plus spécialisées.

2.3.1.6. Source routing

Le « source routing » permet de spécifier (à l'intérieur même d'un paquet IP) les passerelles par lesquelles doit transiter ce paquet. C'est à *proscrire*. Les implications en termes de sécurité sont multiples (spoofing, network ingerring, etc...). `net.ipv4.conf.all.accept_source_route` doit donc être à 0.



Source routing

Il n'y a aucune raison valable d'utiliser le « source routing ». Le laisser activé est extrêmement dangereux. Si l'on désire vraiment s'affranchir de la topologie de routage, on utilisera un VPN.

2.3.1.7. Défragmentation IP

Si la machine configurée doit servir de passerelle NAT (masquerading ou SNAT), il faut activer l'option `net.ipv4.ip_always_defrag`. Cette dernière force la passerelle à effectuer le réassemblage de paquets pour ses clients. Comme la passerelle maintient une table de correspondance qui s'appuie sur les ports source/destination des paquets, un fragment (en dehors du premier) ne pourra pas être dé-NATé, puisqu'il contient une portion du paquet qui ne contient plus ces informations.

2.3.1.8. Spoofing

La variable `net.ipv4.conf.all.rp_filter` permet de vérifier qu'un paquet arrive bien par l'interface sur laquelle il devrait arriver. Par exemple, si mon interface `eth0` est `192.168.0.1/24` et que je reçois un paquet avec l'IP source `192.168.0.34` sur `eth1`, ce paquet sera rejeté puisqu'il ne peut arriver sur cette interface. De même, `127.0.0.1` est une adresse source invalide pour un paquet arrivant sur une interface autre que la loopback (`lo`).

Par défaut, cette fonction est désactivée. Il faut donc activer `net.ipv4.conf.all.rp_filter` en mettant sa valeur à 1 dans `/etc/sysctl.conf`.

2.3.1.9. Logguer les martiens

Les « martiens » sont des paquets ayant des adresses sources ou destination invalides (voir [RFC1812], « 5.3.7 Martian Address Filtering », p.96). Les paquets source routés (Section 2.3.1.6, « Source routing ») et spoofés (Section 2.3.1.8, « Spoofing ») refusés seront aussi loggés grâce à cette règle. Linux permet de logguer les martiens en mettant la valeur 1 dans la variable `net.ipv4.conf.all.log_martians`.

2.3.1.10. Fichier `sysctl.conf`

Le fichier ci-dessous donne une exemple de base de fichier `sysctl.conf`, à ajuster au besoin en fonction de la machine sur laquelle il sera implanté.

```
## On loggue le trafic bizarre
#
# Log des martiens
#
net.ipv4.conf.default.log_martians = 1
net.ipv4.conf.all.log_martians = 1

## Source routing
#
# Refus des paquets source-routés
#
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.all.accept_source_route = 0

#
# On n'envoie pas de paquets avec
# des options de source routing
#
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.all.mc_forwarding = 0

## ICMP
#
# Ignorer les paquets ICMP redirect
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.all.accept_redirects = 0
#
```

```
# Pas d'envoi d'ICMP redirect
#
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.all.send_redirects = 0

## IP forwarding
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# !! Ajuster au besoin          !!
# !! Un routeur doit forwarder !!
# !! Un hôte non                !!
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#
net.ipv4.conf.default.forwarding = 0
net.ipv4.conf.all.forwarding = 0
net.ipv4.ip_forward = 0

## Divers
#
# Reverse path : vérification de la cohérence
# interface d'entrée / table de routage
#
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1

#
# Par de relais bootp/dhcp
#net.ipv4.conf.default.bootp_relay = 0
net.ipv4.conf.all.bootp_relay = 0

## ARP
#
# Pas de proxying ARP
#
net.ipv4.conf.default.proxy_arp = 0
net.ipv4.conf.all.proxy_arp = 0

#
# Taille maximum de la table ARP
#
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# !! Ajuster au besoin          !!
# !! En fonction de la taille  !!
# !! Du subnet occupé          !!
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#
net.ipv4.neigh.default.gc_thresh3 = 256

#
# Limite au dela de laquelle le nettoyage de la table ARP
# sera engagé
#
net.ipv4.neigh.default.gc_thresh2 = 256

#
# Limite en dessous de laquelle le nettoyage de la table ARP
# sera stoppé
#
net.ipv4.neigh.default.gc_thresh1 = 32
```

```
#  
# Intervalle de nettoyage de la table ARP  
#  
net.ipv4.neigh.default.gc_interval = 30
```

2.3.2. IPv6

2.3.2.1. Un peu tôt ?

IPv6 est installé et activé par défaut sur toutes les distributions. Son intérêt est cependant limité. Il n'y a aucune raison d'activer IPv6 tant que le site n'a pas de connectivité IPv6 ou que ce site n'expérimente pas IPv6 explicitement.



Épuisement des adresses IPv4

L'épuisement des adresses IPv4 est annoncé depuis longtemps. En particulier par les promoteurs d'IPv6. Mais aujourd'hui, la date semble plus proche que jamais : plusieurs études ([Potaroo2007], [Hain2007]) s'accordent sur mars 2010. D'ici là, il faudra maîtriser les masques sur 64 bits et les adresses sur 128 !

Laisser IPv6 activé c'est prendre le risque de s'exposer à de nouveaux problèmes. Ce protocole étant relativement nouveau, les problèmes d'implémentation ou même de design apparaissent progressivement ([v6SrcRoute]).

Si malgré tout IPv6 reste activé, il faudra se poser la question de son filtrage périmétrique (sur le périmètre réseau) et sur chaque machine. En effet, par défaut, il n'y a aucun filtrage. Chaque interface IPv6 ayant (ou pouvant avoir) plusieurs adresses (link local, site scope, global scope) le filtrage est plus délicat à mettre en œuvre.

2.3.2.2. Désactiver IPv6

IPv6 peut être désactivé dans le fichier `/etc/modprobe.d/aliases`. Il faut remplacer la ligne :

```
alias net-pf-10 ipv6
```

par

```
alias net-pf-10 off
```

IPv6 ne sera alors plus activé au prochain redémarrage.

Une autre possibilité consiste à ajouter la ligne

```
blacklist ipv6
```

dans un fichier que l'on mettra dans `/etc/modprobe.d/`. Cette option est plus propre : si nous devons effectuer un upgrade du package `module-init-tools`, le fichier `/etc/modprobe.d/aliases` ne sera pas mis à jour si nous l'avons modifié. Cela peut être ennuyeux si la mise à jour nécessite d'introduire des alias pour les pilotes.

```
oper@ubuntu:~$ sudo -i  
Password:****  
root@ubuntu:~# cat > /etc/modprobe.d/blacklist-perso  
blacklist ipv6  
blacklist pcspkr^D  
root@ubuntu:~# cat /etc/modprobe.d/blacklist-perso  
blacklist ipv6  
blacklist pcspkr
```

```
root@ubuntu:~# logout
oper@ubuntu:~$
```

2.4. Configuration de la pile IP

2.4.1. Adressage

La configuration de l'adressage IP sous Ubuntu est concentrée dans le fichier `/etc/network/interfaces`.

Ce fichier contient la liste des interfaces du système et leur configuration IP. Dans sa forme la plus simple, et la plus fréquente, la configuration d'une interface se présente sous cette forme :

```
iface <interface> inet static
address <adresse_ip>
netmask <masque>
gateway <passerelle_par_defaut>
```

La *passerelle_par_defaut* ne devrait apparaître que sur la configuration d'une seule interface.

Dans ce fichier, on trouve aussi la directive `auto`, suivie de noms d'interfaces. Elle permet d'indiquer au système d'activer cette interface au *boot*.

On privilégiera, autant que faire se peut, l'adressage statique (`iface eth0 inet static`) à l'adressage DHCP (`iface eth0 inet dhcp`). Certains serveurs (Samba) ne nécessitent pas forcément une adresse statique, mais utiliser des adresses statiques permet de filtrer plus efficacement les flux entre machines.

La directive `pre-up` peut être adjointe à la configuration d'une interface. Nous l'utiliserons plus bas pour charger les règles de filtrage `iptables`.

Le détail et toutes les possibilités du fichier de configuration `/etc/network/interfaces` sont donnés dans la page de man d'interfaces (`man 5 interfaces`).

```
#
## Interface de loopback
# démarrage au boot des interfaces
auto lo eth0

# définition loopback
iface lo inet loopback

# définition eth0
iface eth0 inet static
    address 192.168.0.33
    netmask 255.255.255.0
    gateway 213.245.116.99
```

Le fichier `/etc/iftab` a une importance lorsque la machine possède plusieurs cartes réseau. Il permet au système de savoir quel nom (`eth0`, `eth1`, ...) il doit affecter à une interface physique.

Le fichier est composé de lignes sous la forme

```
<device> mac <mac>
```

Par exemple :

```
eth0    mac 00:12:79:59:8D:38
```

```
eth1    mac 00:12:79:59:8D:56
```

Grâce à cette association statique entre l'adresse MAC de la carte et le device, on peut être sûr que les périphériques réseau seront conformes à ce qui est prévu. Si ce fichier n'est pas renseigné, on ne peut *absolument pas* être sûr qu'une carte réseau prendra un nom d'interface particulier. L'ordre de détection de ces interfaces pouvant varier au boot, on sera alors sur une règle premier arrivé, premier servi.

2.4.2. Résolution DNS

La résolution DNS est gérée sous les systèmes Unix par une librairie particulière : le *resolver*. Ce *resolver* est principalement configuré dans deux fichiers : un fichier de configuration générale dans lequel on liste les DNS, et un fichier contenant les associations statiques.

2.4.2.1. Configuration générale : `/etc/resolv.conf`

Ce fichier contient la liste des serveurs DNS que le *resolver* interrogera pour effectuer une résolution. Le *resolver* commencera par interroger le premier. S'il ne reçoit aucune réponse, il interrogera le deuxième, etc... Il ne faudra pas confondre *aucune* réponse et réponse *négative* (i.e. « Ce nom d'hôte n'existe pas. »). Une réponse négative *est* une réponse, les autres serveurs ne seront donc pas interrogés.

On pourra spécifier jusqu'à trois serveurs DNS avec le mot clef `nameserver`.

Le mot clef `search` permet de spécifier les *domaines de recherche* qui seront utilisés comme suffixe pour qualifier un nom d'hôte incomplet. Par exemple, si l'on demande au *resolver* de trouver l'adresse IP de `machine`, il devra transformer ce nom en nom d'hôte valide, complètement qualifié. Pour cela, il ajoutera successivement (et uniquement si nécessaire) la valeur configurée dans `search` et effectuera une résolution, jusqu'à ce qu'une réponse positive soit reçue. Afin d'éviter les confusions malheureuses, notamment en utilisant un navigateur Web¹, on évitera de recourir à cette valeur et on utilisera des noms complètement qualifiés.

La directive `domain` permet de spécifier le domaine de l'hôte local et de qualifier automatiquement les noms ne contenant pas de « . ». Si cette directive est absente, le résolveur essaiera de déterminer automatiquement cette valeur à partir du nom d'hôte complet de la machine. `domain` et `search` étant mutuellement exclusifs, on préférera l'utilisation de `domain`, mais on essaiera si possible de *toujours* utiliser des noms d'hôte complètement qualifiés.

```
#
# /etc/resolv.conf
#
# search exemple.lan example.org
# domain exemple.lan.
nameserver 190.62.57.12
nameserver 174.70.56.72
#
#
```

2.4.2.2. Associations statiques : `/etc/hosts`

Ce fichier contient les associations statiques IP/nom. Il est en général interrogé *avant* le serveur DNS². Il est constitué d'entrées sous la forme :

```
adresse_ip    nom
```

Un des usages les plus intéressants de ce fichier est d'y insérer les adresses des hôtes qui sont sous notre responsabilité administrative. Cela permet d'éviter de recourir à un serveur DNS pour résoudre les

¹Les navigateurs utilisent en général leur propre algorithme de requalification de nom *avant* le résolveur.

²L'ordre d'interrogation est donné par la ligne `hosts:` du fichier `/etc/nsswitch.conf`.

noms de nos machines et ainsi de réduire les possibilités d'attaque de type « *Man In The Middle* ». Beaucoup d'autres usages sont possibles : masquer un vrai nom DNS, créer des noms DNS uniquement accessibles à la machine (pour tester des *virtualhosts* (Section 4.5.3, « Site par défaut et VirtualHosts ») sur le serveur web local par exemple), etc...

```
#
# /etc/hosts
#
127.0.0.1      localhost

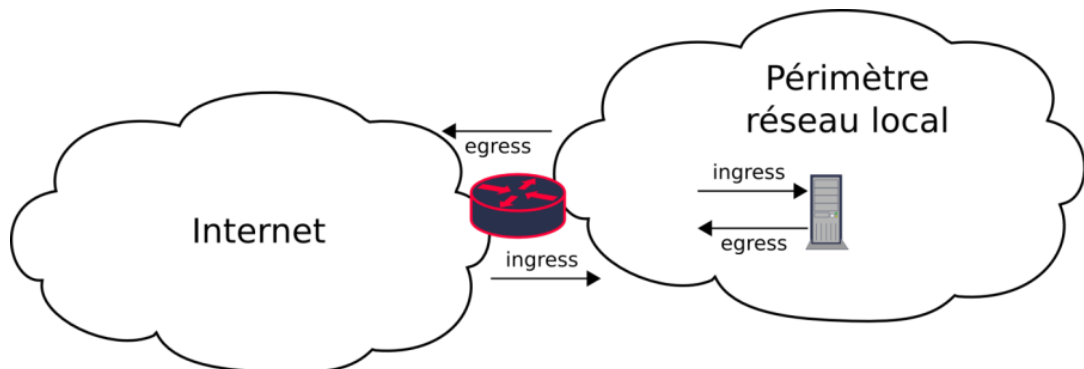
192.168.17.139 ubuntu.exemple.lan vhost1.exemple.lan
                vhost2.exemple.lan
192.168.17.254 gw passerelle gw.exemple.lan

192.168.17.1   alice alice.exemple.lan
#
#
```

2.4.3. Filtrage de base

Le filtrage s'entend généralement par l'élimination des paquets entrants sur un réseau ou un dispositif. Ce n'est qu'en partie exact. Le filtrage, c'est l'application d'une politique à la totalité du trafic : le trafic entrant (qu'il soit à destination de la machine ou devant être relayé par elle) appelé *ingress*, ainsi que le trafic sortant (qu'il soit émis ou simplement transmis par la machine), nommé *egress*.

Figure 2.2. Trafic *egress* et *ingress*



On met souvent l'emphase sur le filtrage du trafic *ingress*, à destination de la machine ou devant être routé par elle vers un réseau interne. A l'évidence, les paquets entrant contiennent du trafic potentiellement dangereux. Mais il ne faut pas sous-estimer l'importance du filtrage *egress* : le trafic que l'on envoie à destination du reste du monde. Ce trafic est aussi potentiellement dangereux : des machines de mon réseau local peuvent être infectées par des vers ou compromises. Le filtrage *egress* permet de limiter au maximum le périmètre des dégâts.

La configuration ci-dessous est un squelette de base pour protéger un serveur. Il faudra l'adapter (en ajoutant des règles sur la chaîne FORWARD) dans le cas d'un routeur.

La politique par défaut utilisée est DROP pour les trois chaînes INPUT, OUTPUT, et FORWARD. Les règles utilisées découpent le trafic sur des chaînes représentant les protocoles ICMP, UDP et TCP (respectivement ICMP_*, UDP_* et TCP_*) ainsi que le sens du trafic (*_IN pour le trafic entrant, *_OUT pour le trafic sortant). Par exemple, la chaîne TCP_OUT contient les règles autorisant le trafic TCP en sortie.

```
#
# On ne traite que filter
#
```

```

*filter
#
# Création et remise à zéro des chaînes
#
:INPUT DROP [0:0]
:FORWARD DROP [0:0] ❶
:OUTPUT DROP [0:0]
:DROP_ME - [0:0] ❷
:ICMP_IN - [0:0] ❸
:ICMP_OUT - [0:0] ❹
:STATEFUL - [0:0] ❺
:TCP_IN - [0:0] ❻
:TCP_OUT - [0:0] ❼
:TCP_INLIMITS - [0:0] ❸
:TCP_SYNLIMITS - [0:0] ❹
:UDP_IN - [0:0] ❶
:UDP_OUT - [0:0] ❶
#

❶ Politiques par défaut des chaînes INPUT, OUTPUT et FORWARD.
❷ Chaîne prenant en charge le rejet du paquet et envoi vers syslog
❺ Cette chaîne acceptera les paquets liés à une connexion existante (plus exactement, à une entrée existante dans la table conntrack).
❸❹❶Création des chaînes dédiées au traitement en entrée des paquets tcp, udp et icmp.
❹❷❶Création des chaînes dédiées au traitement en sortie des paquets tcp, udp et icmp.
❸ Chaîne qui recevra les règles de limitation de trafic TCP.
❹ Chaîne qui recevra les règles de limitation d'ouvertures de connexions TCP entrantes.

#
# #####
# INPUT Dispatch ❶
# #####
#
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state INVALID -j DROP_ME
-A INPUT -s 127.0.0.0/255.0.0.0 -i ! lo -j DROP_ME
-A INPUT -p tcp -j TCP_IN
-A INPUT -p udp -j UDP_IN
-A INPUT -p icmp -j ICMP_IN
#
# #####
# OUTPUT Dispatch ❷
# #####
#
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -p udp -j UDP_OUT
-A OUTPUT -p tcp -j TCP_OUT
-A OUTPUT -p icmp -j ICMP_OUT
-A OUTPUT -j STATEFUL
-A OUTPUT -j REJECT ❸
#
# #####
# STATEFUL : accepte les paquets liés à une connexion existante ❹
# #####
#

```

```
-A STATEFUL -m state --state RELATED,ESTABLISHED -j ACCEPT
-A STATEFUL -j RETURN
...
# #####
# DROP_ME : la chaîne qui jette en laissant des traces dans les
# logs ⑤
# Par défaut, cette chaîne poubellise les paquets en silence
# En changeant la dernière règle par les deux commentées, on
# notifie la source
# du rejet du paquet, c'est plus conforme à la norme. Après, chacun
# décide s'il
# vaut mieux se conformer à la norme avec du trafic qui n'a pas
# lieu d'être...
# --limit permet d'éviter de mettre la machine à genoux en cas de
# déni
# de service
# #####
#
-A DROP_ME -p tcp -m limit --limit 10/min -j LOG --log-prefix
  "DROP:" --log-level 6
-A DROP_ME -p udp -m limit --limit 10/min -j LOG --log-prefix
  "DROP:" --log-level 6
## On pourra utiliser REJECT si l'on souhaite être poli
##-A DROP_ME -p tcp -j REJECT --reject-with tcp-reset
##-A DROP_ME -p udp -j REJECT --reject-with icmp-port-unreachable
-A DROP_ME -j DROP
#
```

- ⑤ Chaîne prenant en charge le rejet du paquet et envoi vers syslog
- ④ Cette chaîne acceptera les paquets liés à une connexion existante (plus exactement, à une entrée existante dans la table *conntrack*).
- ① Dispatch des paquets entrants vers les chaînes de traitement.
- ② Dispatch des paquets sortants vers les chaînes de traitement.
- ③ Permet de rejeter les paquets avant qu'ils n'atteignent la politique par défaut. Cette dernière étant DROP, les délais seraient trop long pour les connexions initiées localement et rejetées. Afin d'accélérer le rejet, on appelle explicitement REJECT.

```
#
# #####
# ICMP entrant ①
# #####
#
-A ICMP_IN -j STATEFUL
-A ICMP_IN -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A ICMP_IN -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A ICMP_IN -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A ICMP_IN -p icmp -m icmp --icmp-type 8 -m limit --limit 5/sec -j
  ACCEPT
#
# #####
# ICMP sortant ②
# #####
#
-A ICMP_OUT -j STATEFUL
-A ICMP_OUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
#
# #####
```



```
# TCP entrant ③
# Il faudra ouvrir des ports au fil de l'eau
# lors de la mise en place de
# services TCP (ssh, apache, ...).
# #####
#
-A TCP_IN -j TCP_INLIMITS
-A TCP_IN -j STATEFUL
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_IN:" --log-level 6
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
# Ajouter les règles ici lors de l'installation de services TCP si
# ces services
# doivent être ouverts
#
# #####
# TCP sortant ④
# Cette machine initie des connexions HTTP vers
# fr.archive.ubuntu.com
# et security.ubuntu.com pour les mises à jour
# #####
#
-A TCP_OUT -j STATEFUL
-A TCP_OUT -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_OUT:" --log-level 6
-A TCP_OUT -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP ⑤
-A TCP_OUT -p tcp -d 194.2.0.36 --dport 80 -j ACCEPT
-A TCP_OUT -p tcp -d 82.211.81.138 --dport 80 -j ACCEPT
-A TCP_OUT -p tcp -d 91.189.88.31 --dport 80 -j ACCEPT
#
# #####
# UDP entrant ⑥
# L'appel à STATEFUL suffit pour accepter les réponses DNS
# Il faudra cependant ouvrir des ports au fil de l'eau lors de la
# mise en place
# de services UDP (DNS, NTP par exemple).
# #####
#
-A UDP_IN -j STATEFUL
# Ajouter les règles ici lors de l'installation de services UDP si
# ces services
# doivent être ouverts
#
# #####
# UDP sortant ⑦
# -remplacer SERVEUR_DNS par le serveur DNS et répéter la ligne
# pour chacun des
# serveurs (primaire, secondaire, etc...)
# -remplacer SERVEUR_NTP par l'adresse IP du serveur NTP si ce
# protocole est
# utilisé
# #####
#
-A UDP_OUT -p udp -m udp -j STATEFUL
-A UDP_OUT -d 192.168.0.254 -p udp -m udp --dport 53 -j ACCEPT
## Remplacer SERVEUR_DNS1, SERVEUR_DNS2 et SERVEUR_NTP par les
# bonnes valeurs
```

```
##-A UDP_OUT -d SERVEUR_DNS1 -p udp -m udp --dport 53 -j ACCEPT
##-A UDP_OUT -d SERVEUR_DNS2 -p udp -m udp --dport 53 -j ACCEPT
##-A UDP_OUT -d SERVEUR_NTP -p udp -m udp --dport 123 -j ACCEPT
## Dans le cas surprenant ou le serveur serait en DHCP
## -A UDP_OUT -p udp -m udp --sport 68 --dport 67 -j ACCEPT #
COMMIT
#
```

- ①③⑥Ajouter dans cette partie les règles permettant respectivement d'accepter les paquets icmp/tcp/udp en entrée. Les paquets retour de connexions initiées par la machine sont normalement pris en charge par la chaîne stateful et ne nécessitent pas d'être explicitement autorisés ici.
- ②④⑦Ajouter dans cette partie les règles permettant respectivement d'accepter les paquets icmp/tcp/udp en sortie. Les paquets retour de connexions initiées par des clients sont normalement pris en charge par la chaîne stateful et ne nécessitent pas d'être explicitement autorisés ici.
- ⑤ Les paquets TCP qui ne sont pas liés à une connexion existante (donc qui ne sont pas matchés par la chaîne STATEFUL) doivent forcément avoir uniquement le flag SYN. Dans le cas contraire, on loggue la situation anormale (avec une limite afin d'éviter un *DoS* par remplissage du système de fichiers).

La chaîne TCP_INLIMITS permet d'appliquer des règles de limitation de trafic avant tout traitement. On l'utilisera plus loin pour protéger des services mais on peut déjà limiter globalement le taux des connexions entrantes.

Nous utiliserons le module iptables `ipt_limit` pour cela. Ce système de limitation fonctionne comme un sac de billes. Dans les règles ci-dessous, chaque ouverture de connexion consomme une bille. Lorsque le sac sera vide, la connexion sera rejetée pour prévenir l'expéditeur et il faudra attendre que le sac se remplisse avant de pouvoir en accepter une autre. Les paramètres `--limit-burst` et `--limit` d'`ipt_limit` contrôlent respectivement le nombre de billes maximum que le sac contient (on commence avec un sac plein) et la vitesse de remplissage de ce sac. Par exemple, si l'on souhaite avoir un « sac » de 10 connexions, se rechargeant à la vitesse d'un par seconde, nous utiliserions :

```
#
# #####
# Limitation des connexions TCP entrantes
# Les connexions trop nombreuses sont rejetées.
# #####
#
# Si la connexion est dans les limites fixées, on retourne d'où
# l'on vient
-A TCP_SYNLIMITS -p tcp -m tcp --syn -m limit --limit 1/sec
--limit-burst 10 -j RETURN
#
# Sinon, on loggue
#
-A TCP_SYNLIMITS -m limit --limit 1/min -j LOG --log-prefix
"TCP_SYNLIMITS:" --log-level 6
-A TCP_SYNLIMITS -j REJECT
#
```

Ces règles pourront être sauveées dans le fichier `/etc/network/iptables` et chargées avant la configuration de l'interface au boot grâce à la directive **pre-up** du fichier de configuration `/etc/network/interfaces` :

```
iface eth0 inet static
address 192.168.0.33
netmask 255.255.255.0
gateway 213.245.116.99
pre-up iptables-restore < /etc/network/iptables
```

2.4.4. Modifier les règles

Pour modifier les règles, il suffit d'éditer le fichier `/etc/network/iptables` et de les réappliquer avec `iptables-restore < /etc/network/iptables`. Attention, si les règles sont modifiées à distance (via ssh par exemple), le pire est à craindre en cas d'erreur, d'autant plus que la probabilité qu'une erreur se glisse dans vos règles est proportionnelle à la distance qui vous sépare de la machine.

Il vaut donc mieux prendre quelques précautions si l'on est pas sur place. Dans ce cas, il sera préférable d'éditer un fichier temporaire (`/etc/network/iptables.test` par exemple). Avant de tester ces règles avec `iptables-restore < /etc/network/iptables.test`, il faudra programmer un reboot de la machine. Si l'application des règles ne nous enferme pas à l'extérieur de la machine, nous pourrions annuler le reboot et renommer le fichier en `/etc/network/iptables`. Dans le cas contraire, nous devons attendre quelques minutes que la machine redémarre avec son ancienne configuration de pare-feu avant de nous reloguer et d'examiner le fichier pour y trouver l'erreur.

```
oper@ubuntu:~$ sudo bash
root@ubuntu:~# shutdown -r +2 &
[1] 4227
root@ubuntu:~#
Broadcast message from oper@ubuntu
      (/dev/pts/0) at 0:39 ...
```

The system is going down for reboot in 2 minutes!

```
root@ubuntu:~# iptables-restore < /etc/network/iptables.test ❶
```

```
root@ubuntu:~# shutdown -c
shutdown: Shutdown cancelled
[1]+  Done                  shutdown -r +5
root@ubuntu:~# mv /etc/network/iptables.test /etc/network/iptables
root@ubuntu:~# logout
oper@ubuntu:~$
```

❶ Après cette commande, si nous sommes figés ou éjectés de la machine, il ne reste plus qu'à attendre...

2.5. Intégrité des fichiers

Malgré toutes les précautions prises (restrictions des services, mise en place d'un firewall, etc...) rien ne garantit l'inviolabilité du serveur. Il convient donc de vérifier périodiquement :

- l'activité du serveur dans les logs
- l'intégrité des binaires et des fichiers de configuration
- vérifier la présence de rootkits

Plusieurs applications occupent ces niches : Samhain [<http://www.la-samhna.de/samhain/>], Integrit [<http://integrit.sourceforge.net/>], AIDE [<http://sourceforge.net/projects/aide/>] pour la vérification d'intégrité. Swatch [<http://swatch.sourceforge.net/>], Logcheck [<http://logcheck.org/>], Splunk< [<http://splunk.org/>] pour l'analyse de logs, chkrootkit [<http://www.chkrootkit.org/>], rkhunter [http://www.rootkit.nl/projects/rootkit_hunter.html] pour la détection de rootkits.

Cette liste n'est pas exhaustive et il est assez difficile de faire un choix dans ce domaine. D'autres outils plus larges, comme PIKT [<http://pikt.org/>] ne sont pas évoqués, car trop complexes et ayant un champ applicatif beaucoup trop large par rapport à la tâche à accomplir. Certains outils sont multiplateformes,

d'autres plus orienté pour des déploiements centralisés, etc... Il faudra évaluer le besoin, débroussailler le terrain, et faire un choix.

Un outil peut néanmoins apporter une solution tout à fait acceptable dans les domaines évoqués : OSSEC [<http://ossec.net>]. Il possède cependant un inconvénient de taille : il n'existe pas encore de package Debian disponibles. Cela va à l'encontre de la règle recommandant de n'utiliser que des paquetages sur le système, et OSSEC est l'archetype du logiciel impossible à désinstaller sans gestion de packaging. Mais les avantages d'OSSEC sont suffisamment importants pour pouvoir éventuellement transgresser le règlement sur ce point.

Ce chapitre sera développé lorsqu'un paquetage OSSEC sera disponible (en cours de création). En attendant, on pourra se référer à ce tutoriel [<http://ubuntuforums.org/showthread.php?t=213445>] pour l'installation d'OSSEC (plutôt simple et intuitive). Il faut garder à l'esprit que ce mode d'installation rendra toute désinstallation difficile à mener à bien complètement.

2.5.1. Installation du serveur OSSEC

TBD

2.5.2. Installation d'un client OSSEC

TBD

Chapitre 3. Déploiement et guide des opérations OpenSSH

\$Revision: 1.20 \$

\$Date: 2007/07/04 13:41:18 \$

OpenSSH est probablement le premier service à configurer sur un serveur. S'il doit y avoir une installation qui doit être plus soignée que d'autres, c'est probablement celle là. Plus de *ssh* implique plus aucun accès distant à la machine. Suivant où se trouve la machine, cela peut être très problématique. Ce chapitre tentera de détailler la mise en place d'OpenSSH dans les meilleures conditions possibles afin de réduire les possibilités de perte ou de dégradation du service, en détaillant les points suivants :

- l'installation d'OpenSSH
- la configuration du serveur et modification des règles iptables
- la création de clef sur les clients et installation sur le serveur.

3.1. Qu'est ce qu'OpenSSH ?

Le besoin qui émerge immédiatement après avoir installé un serveur est de pouvoir y accéder à distance. Historiquement, des logiciels comme les « commandes r » (rsh, rlogin, rexec, ...) ou telnet ont résolu ce problème.

Mais ces outils ont été conçus à une époque où l'Internet n'était peuplé que par quelques utilisateurs bien intentionnés : ils n'effectuent aucun chiffrement sur les données transmises. Cela signifie que n'importe quel utilisateur capturant une connexion qui utilise l'un de ces protocoles pouvait voir la totalité des échanges entre le client et le serveur.

En 1995, Tatu Ylönen, un universitaire finlandais, développa la première version. L'objectif était de pouvoir se connecter à un système en chiffrant le trafic client-serveur. Il fonda une société afin de commercialiser son produit et quelques années plus tard une version libre fût développée par une équipe du projet OpenBSD.

3.2. Installation

L'installation, comme pour la plupart des logiciels packagés, se résume à un **apt-get install openssh-server**. L'installateur se charge même de créer les clefs serveur. On ne proposera pas ici d'exécuter OpenSSH via un super-serveur (inet ou xinetd) comme on le fera pour ProFTPD par exemple (Chapitre 7, *Déploiement et guide des opérations ProFTPD*). La raison est purement technique : même si les profils d'usage sont à peu près identique (inutile, à priori, de laisser tourner OpenSSH en permanence vu la faible fréquentation du service), OpenSSH a besoin de générer une clef au démarrage du serveur (la « clef serveur », la clef d'hôte étant statique). Cette génération étant assez longue (une poignée de secondes), elle induit une réelle latence qui pénalise chaque établissement de connexion.

```
oper@ubuntu:~$ sudo apt-get install openssh-server
Password:*****
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les paquets supplémentaires suivants seront installés :
  libwrap0
Paquets suggérés :
  ssh-askpass xbase-clients rssh molly-guard
Paquets recommandés :
  tcpd
```

```
Les NOUVEAUX paquets suivants seront installés :
  libwrap0 openssh-server
0 mis à jour, 2 nouvellement installés, 0 à enlever et 4 non mis à
jour.
Il est nécessaire de prendre 0o/265ko dans les archives.
Après dépaquetage, 700ko d'espace disque supplémentaires seront
utilisés.
Souhaitez-vous continuer [O/n] ? O
Préconfiguration des paquets...
Sélection du paquet libwrap0 précédemment désélectionné.
(Lecture de la base de données... 14335 fichiers et répertoires
déjà installés.)
Dépaquetage de libwrap0 (à partir de
  ../libwrap0_7.6.dbs-11build1_i386.deb) ...
Sélection du paquet openssh-server précédemment désélectionné.
Dépaquetage de openssh-server (à partir de
  ../openssh-server_4.3p2-8ubuntu1_i386.deb) ...
Paramétrage de libwrap0 (7.6.dbs-11build1) ...

Paramétrage de openssh-server (4.3p2-8ubuntu1) ...
Creating SSH2 RSA key; this may take some time ...
Creating SSH2 DSA key; this may take some time ...
* Restarting OpenBSD Secure Shell server...
  [ OK ]

oper@ubuntu:~$
```

3.3. Configuration

Le serveur OpenSSH se configure via le fichier `/etc/ssh/sshd_config`. Même si la configuration par défaut pour Ubuntu serveur est plutôt bonne, quelques ajustements peuvent améliorer la sécurité du service.

3.3.1. Port d'écoute

Le serveur ssh écoute par défaut sur le port `22/tcp`. Laisser ce port ouvert à n'importe qui peut sembler anodin, puisqu'il qu'une authentification est requise. Mais c'est en fait un risque qui n'a rien de négligeable.

Tout d'abord, le risque est d'être victime d'attaque de force brute. Dans ce cas, le serveur est assailli de tentatives de connexion ssh avec des identifiants/mot de passe tirés de dictionnaire de mots ou générés aléatoirement. En choisissant des mots de passe décents, il y a peu de risque. C'est tout de même ennuyeux puisque cela consomme des ressources que l'on voudrait allouer ailleurs (CPU, remplissage des logs). Mais le risque principal est plutôt de se retrouver sous le coup d'un exploit *zéro-day* : une faille inconnue du public (et des développeurs d'OpenSSH) qui serait exploitée par des attaquants. Si l'on considère le fait que les attaquants se constituent des bases de données service/version à l'avance grâce à des scans massifs, l'apparition d'un tel *0-day* peut potentiellement compromettre le serveur instantanément.

Plusieurs possibilités existent afin de mettre le port 22 à l'abri. Comme d'habitude, ces techniques ont leur avantages et leurs inconvénients, certaines sont meilleures que d'autres et il faudra choisir la plus adaptée en fonction de la situation et de l'appréciation du risque. Il est tout à fait possible de combiner ces solutions entre-elles : changer le port d'écoute, le protéger par un port knocker qui filtrera les paquets reçus pour n'accepter que quelques blocks d'IP spécifiques, etc...

Il faut cependant noter que *plus de sécurité* n'est pas forcément une *meilleure sécurité*, et que mettre en place une usine complexe juste pour défendre l'accès à un port peut conduire à des erreurs ou une

incompréhension du système final qui vont au contraire desservir la sécurité. Il faut garder à l'esprit que le seul garant de la sécurité est OpenSSH lui-même et la qualité de sa configuration et le suivi de ses éventuels problèmes de sécurité seront toujours cruciaux.

Il faut donc considérer ces mesures uniquement pour ce qu'elles sont : éliminer l'accès au port pour éliminer les 99.99% des accès frauduleux (brute force, fingerprinting, script kiddies, ...). Le danger réside évidemment dans les 0.01% restants et là, seul OpenSSH lui-même (et sa configuration) feront la différence.

3.3.1.1. Filtrage d'accès

Un des moyens les plus simples consiste à filtrer l'accès au port pour ne laisser passer que des IP ayant le « droit » d'utiliser le service.

- Avantages : très simple à mettre œuvre par la configuration de règles IPtables (Section 2.4.3, « Filtrage de base ») :

```
#
# #####
# TCP entrant
# Il faudra ouvrir des ports au fil de l'eau
# lors de la mise en place de
# services TCP (ssh, apache, ...).
# #####
#
-A TCP_IN -j TCP_INLIMITS
-A TCP_IN -j STATEFUL
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_IN:" --log-level 6
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
# Ajouter les règles ici lors de l'installation de services TCP
# si ces services
# doivent être ouverts
#
-A TCP_IN -s adresse_ip_autorisée -p tcp -m tcp --dport 22 -j
  ACCEPT ⓘ
#
```

- ⓘ Règle autorisant l'accès au port 22/tcp (ssh) pour l'adresse *adresse_ip_autorisée*

- Inconvénients : il n'est pas toujours possible (client en DHCP, ...) ou suffisant (client NATé) de restreindre l'accès au port ssh par IP. Par ailleurs, une adresse IP peut être « spoofée » par un attaquant qui aurait réussi à déterminer la relation de confiance serveur/*adresse_ip_autorisée*.

3.3.1.2. Déplacement du port d'écoute

Il est possible de changer le port d'écoute (22/tcp) vers l'un de n'importe quel autre des 65535 ports possibles (sshd refuse d'utiliser le port 0, pourtant valide).

- Avantages : très simple à mettre œuvre (il suffit de changer le paramètre `Port` dans le fichier de configuration).
- Inconvénients : cette mesure suffit contre les attaques automatisées basiques, mais est largement insuffisante contre les scans. Rien n'interdit à un attaquant de scanner les autres ports de la machine et de retrouver le serveur sshd grâce au banner fingerprinting. **nmap** est tout à fait capable de reconnaître ssh sur un port différent du port standard.

Il est donc souhaitable d'utiliser d'autres techniques de protection du port si possible.

3.3.1.3. Portknockers (carillons)

Les portknockers permettent de n'ouvrir un port donné que lorsqu'une séquence particulière de paquets vient d'être reçue. Par exemple, un portknocker peut être configuré pour ouvrir le port 22 lorsqu'il a auparavant reçu un paquets TCP syn sur le port 33333 suivi d'un paquet UDP sur le port 1234. Il faudra donc connaître la séquence de paquets à envoyer, le « sésame » afin que l'accès au port soit permis (typiquement grâce à une règle iptables ressemblant à `iptables -I INPUT -p tcp -s ip_ayant_correctement_carillonné --dport 22 -j ACCEPT`. Des outils comme knockd, voire même quelques règles iptables [<http://danieldegraaf.afraid.org/info/iptables/mportknock>] sont capables de fournir ce service.

Cette idée peut sembler astucieuse à première vue, mais souffre d'un problème plutôt ennuyeux : le portknocking « de base » est vulnérable aux attaques par rejeu : une personne mal intentionnée peut, si elle se trouve au milieu de la conversation, capturer de trafic et le rejouer afin d'ouvrir le port ssh. Des techniques plus pointues ont donc été mises en avant afin de contrer le rejeu : séquence de ports construite à partir d'un cryptogramme contenant l'IP source et le port à débloquent (<http://www.portknocking.org/view/details>), SPA [<http://www.portknocking.org/view/details>], (Single Packet Authorization), ... Ces implémentations, beaucoup plus intéressantes, souffrent quand à elles d'un autre défaut : leur mise en place nécessite de déployer un démon supplémentaire sur le système. Et mettre en place un démon en plus c'est s'exposer à de nouveaux problèmes introduits par ce dernier, c'est un service de plus à maintenir, à monitorer, à upgrader, pour lequel il faudra suivre les évolutions et les informations liés à la sécurité...

Chacun décidera en fonction de ses besoins si le jeu en vaut la chandelle. Mais il faut donc bien prendre le port knocking comme ce qu'il est : une méthode simple, mais pas infaillible, de masquer la présence d'un serveur ssh. Point.

```
oper@ubuntu:~$ sudo apt-get install knockd
Password:
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les NOUVEAUX paquets suivants seront installés :
  knockd
0 mis à jour, 1 nouvellement installés, 0 à enlever et 4 non mis à
jour.
Il est nécessaire de prendre 25,1ko dans les archives.
Après dépaquetage, 172ko d'espace disque supplémentaires seront
utilisés.
Réception de : 1 http://fr.archive.ubuntu.com feisty/universe
  knockd 0.5-2ubuntu1 [25,1kB]
25,1ko réceptionnés en 5s (4218o/s)
Sélection du paquet knockd précédemment désélectionné.
(Lecture de la base de données... 14236 fichiers et répertoires
déjà installés.)
Dépaquetage de knockd (à partir de
  .../knockd_0.5-2ubuntu1_i386.deb) ...
Paramétrage de knockd (0.5-2ubuntu1) ...
Not starting knockd. To enable it edit /etc/default/knockd

oper@ubuntu:~$
```

3.3.2. Paramètres

Afin de limiter les effets d'une attaque par force brute, il est préférable de n'autoriser que l'authentification par clef publique. Pour cela, il faut configurer `PasswordAuthentication` à la

valeur `no` dans le fichier de configuration. Il est conseillé d'avoir mis en place l'authentification par clef publique (voir Section 3.4, « Authentification par clef publique ») avant de n'autoriser *que* celle-ci...

Il est aussi conseillé de mettre `PermitRootLogin` à `without-password`. Cela n'apporte rien si `PasswordAuthentication` est à la valeur recommandée, s'avère utile dans le cas contraire en forçant les connexions « as root » à s'authentifier sans mot de passe.

Si le proxying de connexions `ssh` et le transfert de sessions X-Window n'est pas requis (ce qui est généralement le cas dans la mesure où les serveurs n'ont pas de bibliothèques X installées), il est recommandé de passer les valeurs `AllowTcpForwarding` et `X11Forwarding` à `no`

Enfin, même si IPv6 a été désactivé sur la machine, il est conseillé de mettre `AddressFamily` à `inet`, ce qui force le démon `sshd` à n'utiliser qu'IPv4. L'intérêt est double : c'est un sécurité en cas d'erreur de désactivation d'IPv6 (voir Section 2.3.2.2, « Désactiver IPv6 »), mais cela permet aussi de ne pas remplir les logs avec des messages du kernel tentant de charger le module `ipv6` alors qu'il est *blacklisté*.

Les autres paramètres pas défaut sous Ubuntu serveur 7.04 sont corrects. Une fois ces paramètres appliqués, il faut redémarrer le serveur pour qu'ils soient pris en compte.

3.4. Authentification par clef publique

OpenSSH supporte plusieurs types d'authentification dont les plus utilisés sont l'authentification par mot de passe et l'authentification par clef publique. Avec la première méthode, le serveur vous demande de saisir un mot de passe lors de la connexion. Comme vu précédemment, utiliser cette possibilité est déconseillé car elle peut laisser la porte ouverte à des attaques par force brute. Ce n'est pas non plus très pratique : il faudra taper un mot de passe à chaque connexion (même si l'on s'est déconnecté juste avant). Si en plus on doit gérer plusieurs serveurs, il faudra taper un mot de passe (potentiellement différent) à chaque connexion sur ces serveurs.

L'authentification par clef publique, en revanche, ne nécessite pas de taper un mot de passe sur le serveur. Il suffit de prouver au serveur que l'on possède une certaine clef (notre clef privée) correspondant à une clef déposée sur le serveur (notre clef publique).

Cette méthode possède plusieurs avantages. Tout d'abord, l'authentification est décorrélée du mot de passe du compte sur lequel nous voulons nous connecter. Par exemple, si je désire me connecter en tant que `root`, il suffira que je dépose ma clef publique à un endroit spécifique dans le répertoire maison de `root`. Si un autre utilisateur a aussi besoin de se connecter *as root* sur la machine, il suffira de déposer aussi sa clef privée. Tout cela se fait donc sans connaître le mot de passe de `root`¹. Ainsi, lorsque *n* personnes ont accès au compte `root` d'une machine, retirer l'accès à l'une d'elle consiste simplement à retirer sa clef. Si l'on avait utilisé l'authentification par mot de passe, il aurait fallu changer le mot de passe et distribuer ce nouveau mot de passe à *n-1* personnes, avec tous les problèmes pratiques que cela peut poser.

Pour « débloquer » ma clef privée (et ainsi pouvoir valider mon identité en correspondance avec la clef publique), il faudra tout de même taper un mot de passe². Mais OpenSSH est capable, grâce à `ssh-agent`, de mémoriser les clefs privées débloquées. Ainsi, une fois la clef débloquée, vous n'aurez généralement plus de mot de passe à taper tant que vous ne quittez pas votre session.

3.4.1. Création de clef client

La création de la clef (plus exactement de la paire de clefs) se fait avec `ssh-keygen`.

```
alice@linus:~$ ssh-keygen -t dsa ①
Generating public/private dsa key pair.
```

¹il faudra évidemment connaître le mot de passe pour se connecter la première fois sur la machine et y déposer la première clef

²il est possible de créer une clef privée sans mot de passe mais c'est déconseillé car extrêmement dangereux

```
Enter file in which to save the key (/home/alice/.ssh/id_dsa):
Enter passphrase (empty for no passphrase): *****
Enter same passphrase again: *****
Your identification has been saved in /home/alice/.ssh/id_dsa.
Your public key has been saved in /home/alice/.ssh/id_dsa.pub.
The key fingerprint is:
42:24:74:1d:f3:f6:e8:74:24:99:06:4b:41:08:b6:8a alice@linus
alice@linus:~$
```

❶ Type de clef à créer. Il est recommandé d'utiliser des clefs DSA

Les clefs publiques et privées générées seront appelées respectivement `id_dsa.pub` et `id_dsa` et seront créées dans le répertoire `$HOME/.ssh`. A moins d'utiliser plusieurs clefs, il vaut mieux éviter de changer le nom des clefs générées ou leur emplacement. La clef `$HOME/.ssh/id_dsa` est donc la clef privée et elle doit être particulièrement protégée. En cas de doute sur sa confidentialité, il ne faut pas hésiter à régénérer ses clefs.

3.4.2. Mise en place des clefs sur le serveur

Un fois la paire de clefs générée, il faut déposer la clef publique sur le serveur (ici 192.168.17.139). Cette clef sera ajoutée dans le `authorized_keys` de l'utilisateur cible situé dans le répertoire `.ssh`. Par exemple, si Alice veut se connecter sur la machine `ubuntu` en tant qu'utilisateur `root`, il faudra ajouter sa clef publique dans le fichier `~root/.ssh/authorized_keys` sur `ubuntu`.

Lorsque l'usager du serveur cible possède un mot de passe (ce qui n'est pas le cas de `root` sous les Ubuntu), il est possible d'effectuer cette opération en une seule commande avec `ssh-copy-id -i .ssh/id_dsa.pub utilisateur@server`

Voici une session dans laquelle `alice` génère une paire de clef afin de se connecter en tant que `root` sur la machine `serveur`. On part du principe que `PasswordAuthentication` n'a pas encore été changé.

La syntaxe de `ssh` est simplissime :

```
ssh {user@host}
```

ou `user` est le nom d'utilisateur sous lequel l'on veut se connecter et `host` est la machine.

```
alice@linus:~$ scp .ssh/id_dsa.pub oper@192.168.17.139: ❶
The authenticity of host '192.168.17.139 (192.168.17.139)' can't be
established.
RSA key fingerprint is
a2:e5:d2:3a:44:be:e6:2a:10:71:dc:2e:d7:80:c1:c7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.17.139' (RSA) to the list of
known hosts.
oper@ubuntu's password:*****
id_dsa.pub                                100% 602      0.6KB/s
    00:00
alice@linus:~$ ssh oper@192.168.17.139 ❷
oper@192.168.17.139's password: *****
Linux ubuntu 2.6.20-15-server #2 SMP Sun Apr 15 07:41:34 UTC 2007
i686
```

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in `/usr/share/doc/*/copyright`.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

```
Last login: Wed Jun 6 09:03:53 2007
```

```
oper@ubuntu:~$ sudo -i ③
Password:
root@ubuntu:~# mkdir .ssh
root@ubuntu:~# cat ~oper/id_dsa.pub >> .ssh/authorized_keys
root@ubuntu:~# chmod 700 .ssh/ ④
root@ubuntu:~# chmod 600 .ssh/*
root@ubuntu:~# logout ⑤
oper@ubuntu:~$ logout
Connection to 192.168.17.139 closed.
alice@michel:~$ ssh root@192.168.17.139 ⑥
Enter passphrase for key '/home/alice/.ssh/id_dsa': ***** ⑦
Linux ubuntu 2.6.20-15-server #2 SMP Sun Apr 15 07:41:34 UTC 2007
i686
```

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
root@ubuntu:~#

- ① **scp** permet de copier (Secure CoPy) de fichiers à travers un canal ssh sécurisé. Ici on copie `~alice/.ssh/id_dsa.pub` vers le répertoire `$HOME` de l'utilisateur *oper* de la machine *serveur*. Nous sommes obligés de passer par ce compte utilisateur puisque *root* n'a pas de mot de passe. Attention à ne pas oublier les « : » dans la commande, sinon n'effectuera qu'une copie locale.
- ⑦ Alice se connecte en tant que *oper* sur le serveur.
- ③ *oper* est dans le groupe *admin*, et il est donc autorisé à passer *root* grâce à **sudo**.
- ④ **ssh** est très regardant concernant les permissions des fichiers. Il n'autorisera les connexions que si les fichiers d'autorisation et les clefs sont correctement protégées des regards indiscrets. Alice aurait pu aboutir au même résultat en faisant un `umask 077` après le **sudo**.
- ⑤ Une fois le fichier mis en place et les permissions corrigées, Alice peut se déloguer du compte *root* (elle revient du **sudo**), puis du serveur (elle revient du **ssh**).
- ⑥ Le moment de vérité... Alice essaie de se connecter en tant que *root* sur le serveur.
- ⑦ Maintenant Alice utilise le mot de passe qu'elle a utilisé lors de la création de sa clef privée.

Pour faire mémoriser une clef débloquée à **ssh-agent**, il suffit d'invoquer **ssh-add** et de taper le mot de passe de déblocage de la clef.

```
alice@linus:~$ ssh-add
Enter passphrase for /home/alice/.ssh/id_dsa: *****
Identity added: /home/alice/.ssh/id_dsa (/home/alice/.ssh/id_dsa)
Identity added: /home/alice/.ssh/identity (alice@linus)
alice@linux:~$ ssh root@192.168.17.139
Last login: Wed Jun 6 17:54:41 2007 from 192.168.253.70
[root@ubuntu root]#
```

3.4.3. Régénération des clefs sur le serveur

On peut parfois avoir besoin de régénérer les clefs du serveur. Dans le cas de machines clonées par exemple, si rien n'est fait les clefs seront identiques sur les clones. Il est préférable de générer une nouvelle clef de serveur afin que chaque serveur ait son propre jeu de clefs.

Pour régénérer les clefs serveurs, on peut invoquer le script de post-installation Ubuntu avec `dpkg-reconfigure openssh-server` :

```
alice@michel:~$ ssh root@192.168.17.139
...
root@ubuntu:~# rm /etc/ssh/*key*
root@ubuntu:~# dpkg-reconfigure openssh-server
Creating SSH2 RSA key; this may take some time ...
Creating SSH2 DSA key; this may take some time ...
 * Restarting OpenBSD Secure Shell server...
                                [ OK ]
root@ubuntu:~#
```

Une autre possibilité consiste à les générer à la main :

```
alice@michel:~$ ssh root@192.168.17.139
...
root@ubuntu:~# ssh-keygen -f "/etc/ssh/ssh_host_rsa_key" -N '' -t
rsa
Generating public/private rsa key pair.
/etc/ssh/ssh_host_rsa_key already exists.
Overwrite (y/n)? y
Your identification has been saved in /etc/ssh/ssh_host_rsa_key.
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub.
The key fingerprint is:
eb:46:c3:9c:6f:90:06:b6:c9:f6:3a:9b:11:28:87:41 root@ubuntu
root@ubuntu:~# ssh-keygen -f "/etc/ssh/ssh_host_dsa_key" -N '' -t
dsa
Generating public/private dsa key pair.
/etc/ssh/ssh_host_dsa_key already exists.
Overwrite (y/n)? y
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
78:46:c7:97:7d:2f:a7:ad:21:77:9c:01:20:06:29:61 root@ubuntu
root@ubuntu:~#
```

Dans les deux cas, Alice devra aussi effectuer une modification dans sa configuration. En effet, ssh étant plutôt strict, il refusera de vous laisser vous connecter à un serveur dont la clef à changé. Ces clefs serveurs sont parfois stockés globalement (/etc/ssh/ssh_known_hosts) mais le plus souvent resident dans ~/.ssh/known_hosts. Ce fichier est en général rempli par ssh avec la clef d'un serveur lors de la première connexion (sauf si le paramètre *StrictHostKeyChecking* est à *yes* dans /etc/ssh/ssh_config).

Il faudra alors supprimer manuellement l'ancienne clef serveur de ~/.ssh/known_hosts. Comme le message d'erreur comporte le numéro de ligne (NN ici) contenant l'ancienne clef, l'astuce `vi +NNd +x /home/alice/.ssh/known_hosts` permet de supprimer la mauvaise clef en une seule commande. Il suffit ensuite de relancer la connexion et d'accepter la nouvelle clef.

```
alice@linus:~$ ssh root@192.168.17.139
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!    @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle
attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
eb:46:c3:9c:6f:90:06:b6:c9:f6:3a:9b:11:28:87:41.
Please contact your system administrator.
```

```
Add correct host key in /home/alice/.ssh/known_hosts to get rid of
this message.
Offending key in /home/alice/.ssh/known_hosts:43
RSA host key for 192.168.17.139 has changed and you have requested
strict checking.
Host key verification failed.
alice@linus:~$ vi +43d +x /home/alice/.ssh/known_hosts
alice@linus:~$ ssh root@192.168.17.139
The authenticity of host '192.168.17.139 (192.168.17.139)' can't be
established.
RSA key fingerprint is
eb:46:c3:9c:6f:90:06:b6:c9:f6:3a:9b:11:28:87:41.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.17.139' (RSA) to the list of
known hosts.
Enter passphrase for key '/home/alice/.ssh/id_dsa': *****
Last login: Wed Jun 6 22:41:50 2007 from 192.168.0.228
Linux ubuntu 2.6.20-15-server #2 SMP Sun Apr 15 07:41:34 UTC 2007
i686
```

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted
by
applicable law.
root@ubuntu:~#

3.4.4. Modification du mot de passe d'une clef privée

On pourrait critiquer de l'utilité d'une telle fonctionnalité. En effet, pour quelle raison voudrait-on changer de mot de passe ? Soit le mot de passe est compromis (ou est susceptible de l'être). Dans ce cas, la clef privée l'est aussi. Soit on désire utiliser un mot de passe plus robuste. C'est donc que l'ancien ne l'était pas et que la clef n'est pas conséquent plus digne de confiance.

La aussi, chacun d'entre nous devra peser le « pour » et le « contre » et décider d'utiliser ou non cette fonctionnalité. C'est l'option `-p` de la commande **ssh-keygen** qui permet de modifier le mot de passe de la clef.

Chapitre 4. Déploiement et guide des opérations Apache

\$Revision: 1.30 \$

\$Date: 2007/07/07 19:53:15 \$

Apache est sans aucun doute le serveur HTTP le plus configurable du marché. Sa vaste panoplie de directives et surtout la multiplicité de leur combinaison peut rebuter au premier abord. Ce chapitre tente d'introduire les grands principes qui permettront ensuite de configurer des serveurs Web sous Apache sans difficulté.

4.1. Historique et description

Le serveur HTTP Apache a été créé au départ afin de palier à l'arrêt du développement du serveur NCSA. Il est, depuis 1996, le serveur le plus répandu dans le monde ([Netcraft]). Il a été créé au départ afin de palier à l'arrêt du développement du serveur NCSA. Trois versions majeures sont aujourd'hui maintenues :

- la branche 1.3, ancienne, mais très éprouvée et donc particulièrement stable,
- la branche 2.0, qui apporte le support des threads et un meilleur support des plateformes non Unix,
- la branche 2.2, apporte une configuration plus « modulaire » (mais pas forcément plus simple...), des améliorations sur les modules de proxy et d'authentification.

Ubuntu Serveur 7.04 est livrée avec les version 1.3.34 et 2.2.3. Nous utiliserons cette dernière version (package `apache2`), plus intégrée. Mais la branche 2.2 étant plus récente et (relativement) moins testée il faudra être vigilant sur les éventuels correctifs à appliquer en exploitation.

4.2. Architecture

A chaque version majeure, Apache est devenu plus modulaire. Les version 1.x autorisaient l'utilisation de modules pour déléguer une partie du traitement (les fameux `mod_*`); la version 2 a introduit la notion de « Multi-Processing Module » (MPM).

4.2.1. Modèles MPM

Les MPM permettent à Apache de traiter plusieurs requêtes simultanément. Traditionnellement, Apache traitait ce problème en *pré-forkant* : il se dupliquait lui-même N fois (paramétrable) au démarrage et ainsi pouvait traiter un nombre plus important de requêtes. L'inconvénient de cette méthode est l'empreinte mémoire : quand on se réplique, on réplique aussi la mémoire du processus initial.

Les versions 2.0 et supérieures introduisent d'autres modèles, principalement *worker* (ou *thread-pool*) et *event*. Le premier modèle consiste en un seul processus parcouru par plusieurs fils d'exécution (des *threads*). On pourrait faire l'analogie avec une route. Au lieu de construire plusieurs routes entre A et B afin de pouvoir faire circuler plusieurs véhicules (*pre-fork*), on met simplement plusieurs voitures sur une seule route. Il faudra juste veiller aux collisions. C'est plus efficace et cela nécessite moins de bitume. Le modèle *event* en revanche permet de traiter plusieurs connexions dans un seul processus en traitant les connexions les unes après les autres mais partiellement, afin d'avoir un traitement parallèle des connexions (plusieurs voitures mais un seul conducteur).

Le choix de modèle de fonctionnement conditionne le paquetage à installer. Par défaut, un `apt-get install apache2` installera le modèle *worker*. Si l'on désire un autre modèle, il suffira de l'installer explicitement.

```
root@ubuntu:~# apt-get install apache2-mpm-prefork
```

```
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les paquets suivants seront ENLEVÉS :
  apache2-mpm-worker
Les NOUVEAUX paquets suivants seront installés :
  apache2-mpm-prefork
0 mis à jour, 1 nouvellement installés, 1 à enlever et 4 non mis à
jour.
Il est nécessaire de prendre 0o/429ko dans les archives.
Après dépaquetage, 8192o d'espace disque seront libérés.
Souhaitez-vous continuer [O/n] ? O
dpkg : apache2-mpm-worker : problème de dépendance, mais
suppression comme demandé :
  apache2 dépend de apache2-mpm-worker (>= 2.2.3-3.2build1) |
  apache2-mpm-prefork (>= 2.2.3-3.2build1) | apache2-mpm-event (>=
  2.2.3-3.2build1) ; cependant :
  Le paquet apache2-mpm-worker doit être supprimé.
  Le paquet apache2-mpm-prefork n'est pas installé.
  Le paquet apache2-mpm-event n'est pas installé.
(Lecture de la base de données... 14790 fichiers et répertoires
déjà installés.)
Suppression de apache2-mpm-worker ...
 * Stopping web server (apache2)...
[ OK ]
Sélection du paquet apache2-mpm-prefork précédemment désélectionné.
(Lecture de la base de données... 14785 fichiers et répertoires
déjà installés.)
Dépaquetage de apache2-mpm-prefork (à partir de
.../apache2-mpm-prefork_2.2.3-3.2build1_i386.deb) ...
Paramétrage de apache2-mpm-prefork (2.2.3-3.2build1) ...
 * Starting web server (apache2)...
[ OK ]

root@ubuntu:~#
```

Apache recommande le MPM *worker*. En revanche, si PHP est requis, le MPM *pre-fork* doit être utilisé (PHP n'étant pas *thread-safe*).

4.2.2. Modules

Les modules d'Apache permettent de gérer un aspect spécifique du traitement d'une requête http : authentification (*mod_auth*), chiffrement (*mod_ssl*), interprétation du PHP (*mod_php*), etc... Ces modules peuvent être chargés ou non en fonction de la configuration souhaitée. On évitera évidemment d'activer des modules inutiles pour l'usage souhaité d'Apache. Les versions 2+ d'Apache fournissent les outils **a2enmod** et **a2dismod** pour respectivement demander le chargement ou non d'un module au boot.

Par exemple, si l'on désire activer SSL, il suffira d'exécuter la commande `a2enmod ssl` :

```
root@ubuntu:~# a2enmod ssl
Module ssl installed; run /etc/init.d/apache2 force-reload to
enable.
root@ubuntu:~# /etc/init.d/apache2 force-reload
 * Forcing reload of web server (apache2)...
[ OK ]

root@ubuntu:~#
```

Ces scripts (ce sont des scripts shell) fonctionnent un peu à la manière de **update-rc.d** ou **chkconfig** : ils créent un lien symbolique dans `/etc/apache2/mods-enabled/` depuis le fichier de configuration chargeant le module (`/etc/apache2/mods-available/`).

4.3. Gérer le service

4.3.1. Démarrage et arrêt

La gestion du service s'effectue avec un script SysV habituel (`/etc/init.d/apache2`) ou avec la commande d'invocation de ces scripts (**invoke-rc.d**).

```
invoke-rc.d apache2 {[start] | [stop] | [restart] | [reload] | [force-reload]}  
ou
```

```
/etc/init.d/apache2 apache2 {[start] | [stop] | [restart] | [reload] | [force-reload]}
```

Les arguments possibles sont :

- `start` : démarre le serveur,
- `stop` : arrête le serveur; les éventuelles connexions en cours sont brutalement coupées,
- `reload` : arrête le serveur sans couper les connexions en cours (il refuse les nouvelles connexions et attend que les connexions en cours soient terminées),
- `force-reload` : stoppe le serveur avec `stop` puis redémarre avec `start`
- `restart` : comme `force-reload`

Le script `/etc/init.d/apache2` est en fait un « wrapper » (une surcouche) de la commande **apache2ctl**. Il est recommandé d'utiliser le wrapper dans la plupart des cas. Mais cette commande offre quelques fonctionnalités supplémentaires utiles : `graceful-stop` et `configtest` qui permettent, respectivement de stopper le serveur sans couper les connexions en cours et de tester la configuration. Cette dernière possibilité est particulièrement utile sur des serveurs en production. Le script SysV utilise d'ailleurs `configtest` lorsqu'il est invoqué avec l'argument `reload`.

```
root@ubuntu:~# apache2ctl configtest  
Syntax OK  
root@ubuntu:~#
```

4.4. Filtrage

Rien de très compliqué pour ouvrir l'accès au port 80 avec les règles établies précédemment (Section 2.4.3, « Filtrage de base »). En revanche, nous pouvons y ajouter quelques règles afin d'élaborer une stratégie de protection contre les dénis de services.

4.4.1. Filtrage

Exemple 4.1. Apache: configuration du filtrage TCP en entrée

```
#
# #####
# TCP entrant
# Il faudra ouvrir des ports au fil de l'eau
# lors de la mise en place de
# services TCP (ssh, apache, ...).
# #####
#
-A TCP_IN -j TCP_INLIMITS
-A TCP_IN -j STATEFUL
-A TCP_IN -j TCP_SYNLIMITS
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_IN:" --log-level 6
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
# Ajouter les règles ici lors de l'installation de services TCP si
# ces services
# doivent être ouverts
#
-A TCP_IN -s adresse_ip_autorisée -p tcp -m tcp --dport 80 -j
  ACCEPT ❶
# on peut aussi débloquer le port 80 pour tout le monde
-A TCP_IN -p tcp -m tcp --dport 80 -j ACCEPT ❷
#
```

- ❶ Règle autorisant l'accès au port 80/tcp (http) pour l'adresse *adresse_ip_autorisée* (qui peut aussi être un subnet). Cette règle peut être répétée autant de fois que nécessaire.
- ❷ Règle autorisant l'accès au port 80/tcp (http) pour tout le monde.

4.4.2. « Protection » contre les dénis de service (DoS)

Les dénis de service sont une forme d'attaque visant à mettre un service hors d'état de fonctionner. L'idée générale est de noyer le serveur sous des requêtes en provenance d'une machine ou d'un groupe de machines (on parle alors dans ce dernier cas de déni de service distribués, *DDoS*). Cela se traduit en général par un serveur hors-service (incapable de traiter les requêtes), une machine avec un CPU à 100%, une bande passante saturée, des routeurs dépassés par les événements, voire une combinaison de ces facteurs.

La stratégie déployée ici vise uniquement à protéger le serveur HTTP. Si un flot important de paquets arrive au serveur, il est trop tard pour la bande passante. Il faudra donc, dans la mesure du possible, prévoir d'autres mécanismes en amont (*upstream*) afin de protéger la disponibilité de son réseau contre ces attaques et leurs variantes (le *DRDoS* par exemple).

4.4.2.1. Utilisation de LIMIT

Un première idée consiste à limiter (avec `libipt_limit`), comme dans l'extrait suivant :

Exemple 4.2. Apache : Limitation de connexions avec ipt_limit

```
...
# #####
# Limitation TCP entrant
# Limitation du trafic TCP entrant
# Le trafic hors limite est droppé.
# #####
##
# On limite le nombre de paquets entrant sur le port http a 100
# ce nombre sera rechargé de 10 par seconde.
#
-A TCP_INLIMITS -p tcp --dport 80 -m limit --limit 10/sec
  --limit-burst 100 -j ACCEPT ❶
-A TCP_INLIMITS -p tcp --dport 80 -j DROP ❷
-A TCP_INLIMITS -j RETURN
...
```

- ❶ cette règle autorise un 'pool' de 100 paquets entrants. Ce pool sera rechargé au rythme de 10 par seconde
- ❷ ce qui dépasse ces limites (et donc qui n'a pas pu être pris par `-j ACCEPT`) sera jetté.

Un test simple permet de vérifier si ces règles fonctionnent.

```
alice@linux:~$ time (while true; ❶
> do
> echo -e "GET /
HTTP/1.1\nHost:192.168.17.139\nKeep-Alive:300\nConnection:
keep-alive\n\n";
> done) | nc 192.168.17.139 80 > /dev/null

real    0m0.162s
user    0m0.140s
sys     0m0.020s

...
root@ubuntu:~# /sbin/iptables-restore < /etc/network/iptables ❷

...
alice@linux:~$ time (while true; ❸
> do
> echo -e "GET /
HTTP/1.1\nHost:192.168.17.139\nKeep-Alive:300\nConnection:
keep-alive\n\n";
> done) | nc 192.168.17.139 80 > /dev/null

real    0m10.195s
user    0m0.244s
sys     0m0.036s
```

- ❶ test effectué avant l'application des règles LIMIT. La commande se termine car apache ferme la connexion *keep-alive* après 100 requêtes.
- ❷ application des règles sur le serveur
- ❸ test effectué après l'application des règles LIMIT. Le temps nécessaire pour faire aboutir les (100) requêtes et beaucoup plus long.

Cette politique semble donc fonctionner, mais elle a un gros inconvénient : elle produit exactement l'inverse de l'effet souhaité. En effet, ces règles limitent fortement les paquets entrants à destination du port 80. En cas de déni de service, les connexions légitimes auront encore plus de mal à aboutir.

Il faudra donc, si possible, insérer des règles permettant d'accepter du trafic en provenance d'adresses que l'on désire servir quoi qu'il arrive. Cela n'est malheureusement pas toujours possible. Il faudra donc utiliser une stratégie plus évoluée avec par exemple `libipt_recent`.

4.4.2.2. Utilisation de RECENT

Le module `libipt_recent` permet une gestion plus fine par adresse IP. On pourra appliquer des restrictions en paquets par seconde individuellement à chaque adresse. On devra d'abord précéder le chaîne `TCP_IN` par un saut vers une nouvelle chaîne dédié à la gestion des limites :

Exemple 4.3. Apache : Limitation de connexions avec `ipt_recent`

```
#
*filter
#
# Création et remise à zéro des chaînes
#
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:DROP_ME - [0:0]
:ICMP_IN - [0:0]
:ICMP_OUT - [0:0]
:STATEFUL - [0:0]
:TCP_IN - [0:0]
:TCP_INLIMITS - [0:0] ❶
...

# #####
# TCP entrant
# Il faudra ouvrir des ports au fil de l'eau
# lors de la mise en place de
# services TCP (ssh, apache, ...).
# #####
#
-A TCP_IN -j TCP_INLIMITS ❷
-A TCP_IN -j STATEFUL
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_IN:" --log-level 6
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
...

❶ Création de la chaîne et mise à zéro.
❷ Saut vers la chaîne de limitation dès l'arrivée du paquet sur TCP_IN.

...
-A TCP_IN -p tcp --dport 80 -m recent --name HTTP_DOS --set ❶
-A TCP_IN -p tcp --dport 80 -m recent --name HTTP_DOS --update
  --hitcount 15 --seconds 30 -j DROP ❷
...
```

- ❶ cette règle met dans la liste `HTTP_DOS` l'adresse IP source des paquets arrivant sur le port 80
- ❷ si une IP source a été vue plus de 15 fois dans les 30 dernières secondes, il est jeté.

Le même test que précédemment permet de s'assurer que le filtrage fonctionne. Ce n'est pas parfait. On pourra par exemple y adjoindre des listes blanches avec d'autres règles permettant de ne pas limiter le trafic en provenance de certaines sources de confiance. Pour une plus grande efficacité, on devra utiliser des mécanismes de QoS permettant de limiter le trafic sortant du serveur (voir « Chapter 9. Queueing Disciplines for Bandwidth Management » dans [LARTC]).

4.5. Configuration

Apache a vu ses fichiers de configuration complètement remaniés entre les versions 1.x et les versions 2.x. Les anciennes configuration étaient très monolithiques par défaut. Il y avait généralement un seul fichier de configuration (`/etc/httpd/httpd.conf`), accompagné parfois de quelques inclusions correspondant à des virtualhosts. Dans sa version 2.x, c'est une toute autre histoire... En voulant faire plus « modulaire », les développeurs n'ont pas forcément fait plus simple. Chacun appréciera.

4.5.1. Fichiers de configuration

La totalité de la configuration d'Apache 2.x se trouve dans une arborescence sous `/etc/apache2`. Plusieurs sous-répertoires à cet emplacement ont des vocations particulières.

- `conf.d` : contient des « morceaux » de configuration générale non inclus dans le fichier principal. Cela permet, lors de l'installation de modules, d'ajouter des morceaux de configuration qui leur sont propres sans avoir à modifier le fichier de configuration principal.
- `mods-available` : fichiers de chargement de tous les modules Apache disponibles sur le système.
- `mods-enabled` : liens vers les fichiers de chargement requis
- `sites-available` : fichiers de configuration de tous les sites web sur le système.
- `sites-enabled` : liens vers les fichiers de configuration des sites web activés pour le serveur.

Le fichier `ports.conf`, aussi dans ce répertoire, contient les numéros de ports sur lesquels apache doit écouter. Enfin, la configuration générale du serveur se trouve dans le fichier `apache2.conf` de ce même répertoire.

4.5.2. Configuration générale

La configuration générale définit les grands paramètres régissant le fonctionnement du serveur. La section `mpm*_module` par exemple, définira les paramètres appliqués au MPM en vigueur.

Pour le MPM `pre-fork`, on peut par exemple définir combien de processus Apache doit initialement lancer pour traiter les requêtes (`StartServers`). Apache pourra lancer d'autres processus (à concurrence de `MaxClients`) si les processus sont tous occupés. Apache fera en sorte d'avoir toujours `MinSpareServers` libre d'avance, et commencera à supprimer des processus lorsque plus de `MaxSpareServers` seront au repos. Accessoirement, apache peut décider de supprimer un processus lorsqu'il a traité `MaxRequestsPerChild`.

Le nombre de requêtes servies en `keep-alive` évoqué plus haut est défini par le paramètre `MaxKeepAliveRequests`, tandis que la durée maximum d'une connexion est définie dans `KeepAliveTimeout`. Les paramètres permettent de conserver une seule connexion ouverte et d'y faire transiter plusieurs requêtes. Cela permet de réduire le nombre d'établissement de connexions, voire de mettre en oeuvre du pipelining HTTP [http://en.wikipedia.org/wiki/HTTP_pipelining].

Mais l'utilisation de cette fonctionnalité (elle est activée par défaut grâce à `KeepAlive On`) a une conséquence : elle mobilise les processus plus longtemps que nécessaire. En effet, si le client HTTP demande de conserver la connexion ouverte, elle le restera jusqu'à l'expiration de `KeepAliveTimeout` secondes (ou après que `MaxKeepAliveRequests` aient été satisfaites, ce qui a peu de chances de se produire avec la valeur par défaut et en conditions normales). La valeur configurée par défaut étant de 15 secondes, cela implique que chaque client en `keep-alive` va mobiliser

un processus pendant 1/4 de seconde. Il est donc recommandé de réduire fortement la valeur de `KeepAliveTimeout` pour la passer à 5 secondes maximum. Il faudra peut être augmenter la valeur de `MaxClients` si le serveur est chargé, afin de compenser l'effet du keep-alive.

Les séries de directives `Limit*` et `RLimit*` permettent respectivement de régler finement un certain nombre de paramètres HTTP et système d'Apache. Dans la plupart des cas, on pourra laisser les valeurs à leur défaut. `LimitRequestBody`, qui limite la taille d'une requête HTTP et n'a aucune valeur par défaut pourra être positionné à une valeur proche de celle définie pour la taille maximale d'un POST PHP (voir `post_max_size`, Section 5.2.2, « Limites »).

Deux autres paramètres doivent être modifiés par rapport à la configuration par défaut. `ServerTokens` permet de modifier les informations renvoyées par le serveur dans les en-têtes réponse HTTP. Il est préférable de mettre sa valeur à `Prod`, qui se contente de répondre Apache dans les en-tête.

```
alice@linus:~$ telnet 192.168.17.139 80
Trying 192.168.17.139...
Connected to 192.168.17.139.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 10 Jun 2007 15:18:13 GMT
Server: Apache/2.2.3 (Ubuntu) ❶

Connection: close
Content-Type: text/html; charset=UTF-8

Connection closed by foreign host.
```

... ❷

```
alice@linus:~$ telnet 192.168.17.139 80
Trying 192.168.17.139...
Connected to 192.168.17.139.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 10 Jun 2007 15:18:56 GMT
Server: Apache ❸
Connection: close
Content-Type: text/html; charset=UTF-8
```

```
Connection closed by foreign host.
alice@michel:~$
```

- ❶ En-tête renvoyé par Apache avec la configuration par défaut (`ServerTokens Full`).
- ❷ Changement de configuration et redémarrage d'Apache.
- ❸ En-tête renvoyé par Apache avec la configuration `ServerTokens Prod`.

Afin d'être complet sur le masquage de version, il faudra aussi modifier la valeur par défaut de `ServerSignature` en la passant à `Off`. Cette variable détermine l'affichage du nom du serveur et de sa version sur les pages générées automatiquement par le serveur (erreurs, listings de répertoires, etc...).

Le fichier principal contient ensuite des directives pour quelques modules nécessitant une configuration globale :

- `mod_alias` : lien entre une URL spécifique et un répertoire

- `mod_autoindex` : generation d'index de répertoire automatique
- `mod_mime` : gestion et surcharge des types mime du système
- `mod_negotiation` : négociation de la priorité de la langue
- `mod_setenvif` : changement de « variables d'environnement » en fonction de paramètres
- `mod_status` : état du démon et de l'occupation des différents threads/processus
- `mod_info` : informations sur la configuration

4.5.3. Site par défaut et VirtualHosts

Dans sa version 1.1, le protocole HTTP transmet l'en-tête `Host` : dans les requêtes. Cela permet d'héberger des serveurs Web « virtuels » avec des noms différents (`www.exemple.com`, `www2.exemple.com`, `www.autreexemple.com`, ...) sur une seule machine physique ayant une seule adresse IP. Sans cet en-tête, impossible de savoir à quel serveur « virtuel » (*virtualhost*) le client désire accéder. Dans ce cas, c'est le serveur par défaut qui répond.

Les paramètres du serveur par défaut sont définis dans `/etc/apache2/sites-available/default`. Ce fichier est principalement constitué des directives :

```
NameVirtualHost *
<VirtualHost *>
...
</VirtualHost>
```

`NameVirtualHost` permet d'indiquer à Apache que nous allons configurer des serveurs virtuels par *nom* (en analysant l'en-tête client `Host` :), et ce, sur n'importe quelle adresse IP de la machine (« * »).

Tous les paramètres inclus en suite entre les balises `<VirtualHost *` et `</VirtualHost>` définissent les paramètres du serveur par défaut (représenté par « * » la aussi). Ce serveur sera utilisé lorsque l'en-tête « `Host` : » reçu par un client ne correspondra à aucun hôte virtuel défini. Par exemple, si les hôtes virtuels `www.exemple.com` et `www.autreexemple.com` sont définis et qu'une requête HTTP est reçue avec l'en-tête `Host` : `www.serveur.net`, c'est le serveur par défaut qui sera utilisé.

Le serveur par défaut est défini dans un serveur virtuel, mais ce n'est pas une obligation. Ses directives de configuration auraient très bien pu apparaître dans le contexte général (hors `<VirtualHost *` ... `</VirtualHost>`).

Si des serveurs additionnels virtuels doivent être définis, il faudra créer d'autres entrées `<VirtualHost *`. Par exemple :

```
<VirtualHost www.exemple.com>
...
</VirtualHost>
```

Il est préférable de mettre chaque configuration d'hôte virtuel dans un fichier propre sous `/etc/apache2/sites-available` puis d'indiquer à Apache que ce site doit être activé avec **a2ensite** (Cf. Section 4.5.4.3, « Exemple »).

4.5.4. Création d'un VirtualHost

Quelques paramètres suffisent à mettre en œuvre un serveur virtuel basique. Les différents paramètres utilisés dans l'hôte virtuel « default » sont décrits ici.

4.5.4.1. Balises

- Les balises `<Directory>` et `</Directory>` encadrent des paramètres de configuration qui vont s'appliquer au répertoire du système de fichiers donné en paramètre, ainsi qu'à tous ses sous

répertoires. Par exemple, ce qui est inclus dans la section ci-dessous sera appliqué à la totalité du système de fichiers, sauf si une autre directive plus précise est appliquée par la suite.

```
<Directory />
  Options FollowSymLinks
  AllowOverride None
</Directory>
```

- Les balises `<Location>` et `</Location>` sont quasiment identiques à `Directory` à la différence qu'elles représentent un chemin de l'URL demandée par le client. `<Location />` n'est donc *pas* la même chose que `<Directory />` : dans le premier cas nous appliquons des directives à la racine du stie, dans l'autre cas à la racine du filesystem !
- Les balises `<Files>` et `</Files>` permettent de restreindre l'application de directives à certains fichiers. On pourra utiliser des expressions régulières dans la spécification du nom de fichier en ajoutant « ~ » dans la balise (`<Files ~ . . .>`). Un exemple est donné dans le chapitre consacré à PHP (Section 5.2.2, « Limites »). Pour plus de clarté, on préférera probablement la directive `<FilesMatch>`, traitant exclusivement des expressions régulières.

4.5.4.2. Directives

- `ServerAdmin` permet de définir l'email du webmaster du site. Lors de l'affichage d'une page d'erreur, cette adresse email sera affichée (en fonction du paramètre utilisé pour `ServerSignature`, voir Section 4.5.2, « Configuration générale »).
- `DocumentRoot` indique l'emplacement de la *racine* du site Web sur le système de fichier. Lorsqu'un client demandera `http://www.exemple.com/page.html`, Apache essayera de lire le fichier `page.html` dans le répertoire pointé par `DocumentRoot`.
- `ScriptAlias` indique à Apache l'emplacement du répertoire `cgi-bin`, dans lequel on autorisera l'exécution de scripts. Les directives placées dans la balise `<Directory "/usr/lib/cgi-bin">` permettent de préciser la configuration du répertoire contenant les scripts.
- `ErrorLog` indique à Apache le chemin du fichier contenant les logs d'erreur pour cet hôte virtuel. Le niveau de sensibilité est paramétré par `LogLevel`.
- `CustomLog` permet lui d'utiliser un log personnalisé (défini dans le fichier de configuration principal) afin de logger les requêtes client.
- `Alias` est un directive qui est gérée par `mod_alias`. Elle permet de faire correspondre le chemin de la page dans l'URL à un emplacement physique sur le serveur. Ainsi, dans la configuration par défaut, l'accès à l'URL `http://serveur/doc/` renverra le contenu du répertoire `/usr/share/doc/`.

Au sein de chaque section des paramètres permettent de configurer le comportement d'Apache.

- La directive `Options` indique les fonctionnalités disponibles dans un répertoire particulier. `FollowSymLinks` indique à Apache qu'il peut suivre les liens symboliques; `Indexes` qu'il peut générer un index du répertoire si aucun fichier `index.html` n'est présent. D'autres options moins importantes existent et sont détaillées dans le manuel d'Apache.
- `AllowOverride` indique quels sont les paramètres que l'on peut surcharger en plaçant un fichier `.htaccess` dans un répertoire. Cette fonctionnalité permet de laisser le contrôle sur certains paramètres à des utilisateurs n'ayant pas accès à la configuration Apache.

Il conviendra de désactiver cette possibilité de surcharge (avec `AllowOverride none`) sauf si elle est vraiment requise. En effet, si on autorise la surcharge, Apache va devoir vérifier dans chaque sous répertoire si un fichier `.htaccess` est présent. Par exemple, si un navigateur demande un objet situé dans le répertoire `/var/www/documents/apache/documentation/2/2/3/francais/` etc qu'`AllowOverride` est positionné à une autre valeur que `none` pour le répertoire `/`, Apache va devoir chercher un fichier `.htaccess` dans chacun des répertoires du chemin et appliquer les éventuelles directives trouvées au passage.

4.5.4.3. Exemple

```
alice@ubuntu:~$ mkdir /home/alice/{monsieuebe,meslogs}
alice@ubuntu:~$ echo "<html><head><title>hello</title></head><body>
\
<h1>En construction</h1></html>" >
/home/alice/monsieuebe/index.html
alice@ubuntu:~$ sudo -i
Password:
root@ubuntu:~# cat /etc/apache2/sites-available/site_alice
<VirtualHost *>
    ServerAdmin alice@example.org
    ServerName alice.example.org
    DocumentRoot /home/alice/monsieuebe/

    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
    </Directory>

    ErrorLog /home/alice/meslogs/error.log

    # Possible values include: debug, info, notice, warn, error,
    crit,
    # alert, emerg.
    LogLevel info

    CustomLog /home/alice/meslogs/access.log combined
    ServerSignature Off
</VirtualHost>
root@ubuntu:~# echo 192.168.17.139 alice.example.org >> /etc/hosts
root@ubuntu:~# a2ensite site_alice
Site site_alice installed; run /etc/init.d/apache2 reload to
enable.
root@ubuntu:~# /etc/init.d/apache2 reload
* Reloading web server config...
[OK]
root@ubuntu:~#
```

4.5.5. Contrôle d'accès

Apache permet de paramétrer finement l'accès aux ressources hébergées. La grande diversité de modules permet d'utiliser des types d'authentification variés : fichier de mots de passe, base de données, LDAP, radius, certificats, ... Le serveur possède aussi des fonctionnalités de filtrage afin de restreindre l'accès aux ressources à certaines IP autorisées. Ces possibilités sont combinables entre-elles (authentification et filtrage).

4.5.5.1. Authentification

L'authentification peut être mise en œuvre très simplement dans Apache à l'aide de fichiers de mots de passe. C'est tout à fait acceptable pour un petit site gérant un petit nombre d'utilisateurs qui changent peu. Pour des configurations nécessitant un plus grand nombre d'utilisateurs il vaudra mieux d'utiliser des bases de données.

Deux type d'authentifications sont possibles : l'authentification *basic* et *digest*. La première se contente d'envoyer la chaîne de caractères utilisateur:mot_de_passe encodé en Base64 ([RFC2617]) au serveur. Ce mode est peu sécurisé : n'importe quelle personne pouvant capturer la conversation client serveur pourra retrouver l'identifiant et le mot de passe.

L'authentification *digest* en revanche fonctionne différemment : lorsque le client cherche à se connecter au serveur, le serveur lui indique qu'il faut s'authentifier et lui transmet un *challenge* qui servira au client, grâce à une fonction de hashage, à construire une chaîne d'authentification. Le serveur pourra vérifier sans ambiguïté cette chaîne puisqu'il possède les mêmes informations. ([RFC2617], « 3.3 Digest Operation », p. 17). Dans ce mode, le mot de passe et l'identifiant ne circulent jamais en clair sur le réseau et un éventuel attaquant ne pourra obtenir ces informations. L'exemple d'implémentation ci-dessous force le serveur à demander une authentification pour les accès aux documents situés sous le répertoire « *private* » :

```
<Location /private/>
  AuthType Digest
  AuthName "Zone privée" ❶
  AuthDigestDomain /private/ http://www.exemple.com/private/ ❷

  AuthDigestProvider file ❸
  AuthUserFile /var/www/.users ❹
  Require valid-user ❺
</Location>
```

- ❶ Le nom du « realm » (royaume !). C'est une zone logique pour laquelle on utilise la même authentification.
- ❷ Définit l'URL de base pour laquelle s'applique l'authentification
- ❸ Indique que les données d'authentification sont dans un fichier (valeur par défaut). `mod_auth_digest` peut aussi utiliser des bases de données.
- ❹ Fichier contenant la liste des utilisateurs et leurs paramètres d'authentification.
- ❺ Un utilisateur valide et authentifié est obligatoire !

Il faudra créer le fichier `/var/www/.users` avec l'utilitaire **htdigest** afin de constituer la liste des utilisateurs.

```
root@ubuntu:/var/www# htdigest -c .users "Zone privée" alice ❶
Adding password for alice in realm zoneprivée.
New password:
Re-type new password:
root@ubuntu:/var/www# cat .users
alice:Zone privée:cca4760367f40dafae44b6ae98d65498
root@ubuntu:/var/www#
```

- ❶ « Zone privée » doit correspondre au *realm* défini dans la configuration.

4.5.5.2. Restrictions

Apache permet d'implémenter un « filtrage » d'accès par adresse IP avec les directives `Order`, `Allow` et `Deny`. Ces directives peuvent apparaître dans les sections `Location` ou `Directory`.

`Order` peut prendre les valeurs `Deny`, `Allow` ou `Allow,Deny`. La formulation de la directive `Order` n'est pas particulièrement intuitive. Le tableau ci-dessous aidera à y voir plus clair.

Tableau 4.1. Apache : détermination de la restriction

Correspondance	Résultat pour Allow,Deny	Résultat pour Deny,Allow
Allow correspond	Requête autorisée	Requête autorisée
Deny correspond	Requête refusée	Requête refusée
Aucune correspondance	Defaut sur la 2ème directive: Refusé	Defaut sur la 2ème directive: Autorisé
Allow et Deny correspondent	La dernière correspondance décide : Refusée	La dernière correspondance décide : Autorisée

A titre d'exemple, les règles ci-dessous demandent à Apache de ne renvoyer les objets qu'aux requêtes en provenance du réseau 192.168.17.0/24. Les autres recevront une réponse 403 - Forbidden.

```
<Location />  
  Order Deny,Allow  
  Deny from all  
  Allow from 192.168.17.0/24  
</Location>
```

4.5.5.3. Modules divers

Apache, toujours grâce aux modules, peut faire beaucoup plus : réécriture d'URL (*mod_rewrite*), proxy HTTP (*mod_proxy*), détection d'attaques applicatives (*mod_security*), correction de fautes d'orthographe dans les URL (*mod_speling*¹), affichage de pages automatiquement dans la langue préférée du visiteur (*mod_negotiation*), etc... On se reportera à la documentation [<http://httpd.apache.org/docs/2.2/mod/>] d'Apache pour la liste des modules officiels et leur directives de configuration.

¹Orthographié correctement, il devrait s'appeler *mod_spelling*. On appréciera l'humour des développeurs d'Apache :)

Chapitre 5. Déploiement et guide des opérations PHP

\$Revision: 1.19 \$

\$Date: 2007/07/07 19:53:15 \$

Rares sont les serveurs Apache qui ne sont pas accompagnés du module PHP permettant d'interpréter des *scripts* sur le serveur. Après avoir lu le chapitre traitant des questions générales de sécurité (Chapitre 1, *Principes généraux*), les mots « scripts » et « serveur » dans une même phrase doivent inquiéter. A juste titre. PHP est probablement l'un des vecteurs de nuisances les plus utilisés aujourd'hui. Même s'il est parfois impossible de contrôler la qualité du code PHP déposée sur votre serveur, on pourra prendre un certain nombre de mesures afin de « limiter les dégâts » en cas d'exploitation de ce code. Ce chapitre en présente quelques unes.

5.1. Installation

PHP fonctionne comme *module* d'Apache. L'installation de PHP consiste donc à installer ce module et à configurer Apache pour l'utiliser.

```
root@ubuntu:~# apt-get install libapache2-mod-php5
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les paquets supplémentaires suivants seront installés :
  libxml2 php5-common
Paquets suggérés :
  php-pear
Paquets recommandés :
  xml-core
Les NOUVEAUX paquets suivants seront installés :
  libapache2-mod-php5 libxml2 php5-common
0 mis à jour, 3 nouvellement installés, 0 à enlever et 8...
Il est nécessaire de prendre 2754ko/3515ko dans les archives.
Après dépaquetage, 7782ko d'espace disque supplémentaire...
Souhaitez-vous continuer [O/n] ? O
Réception de : 1 http://security.ubuntu.com feisty-security/main php5-...
Réception de : 2 http://security.ubuntu.com feisty-security/main libap...
2754ko réceptionnés en 1m5s (41,9ko/s)

Sélection du paquet libxml2 précédemment désélectionné.
(Lecture de la base de données... 16145 fichiers et répertoires déjà install...
Dépaquetage de libxml2 (à partir de ../libxml2_2.6.27.dfsg-lubuntu3_i386.de...
Sélection du paquet php5-common précédemment désélectionné.
Dépaquetage de php5-common (à partir de ../php5-common_5.2.1-0ubuntu1.2_i38...
Sélection du paquet libapache2-mod-php5 précédemment désélectionné.
Dépaquetage de libapache2-mod-php5 (à partir de ../libapache2-mod-php5_5.2....
Paramétrage de libxml2 (2.6.27.dfsg-lubuntu3) ...

Paramétrage de php5-common (5.2.1-0ubuntu1.2) ...
Paramétrage de libapache2-mod-php5 (5.2.1-0ubuntu1.2) ...

Creating config file /etc/php5/apache2/php.ini with new version
* Forcing reload of web server (apache2)... [ OK ]

root@ubuntu:~#
```

Lors de l'installation, les fichiers de configuration Apache permettant de charger le module PHP sont installés dans répertoire `/etc/apache2/modules-available/`. Il suffit donc d'activer le module comme vu précédemment :

```
root@ubuntu:~# a2enmod php5
Module php5 installed; run /etc/init.d/apache2 force-reload to
enable.
root@ubuntu:~# /etc/init.d/apache2 force-reload
```

```
* Forcing reload of web server (apache2)...  
[ OK ]  
root@ubuntu:~#
```

On pourra vérifier simplement que PHP est bien installé et fonctionne correctement en créant une page dynamique PHP à la racine du site avec la simple commande :

```
echo "<? phpinfo(); ?>" > /var/www/index.php
```

Il suffira ensuite de faire pointer un navigateur sur l'URL `http://<ip_serveur>/info.php` pour obtenir une page d'information sur le serveur Apache/PHP.

Figure 5.1. PHP Info

System	Linux ubuntu 2.6.20-15-server #2 SMP Sun Apr 15 07:41:34 UTC 2007 i686
Build Date	May 22 2007 18:51:12
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
additional .ini files parsed	/etc/php5/apache2/conf.d/pdo.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	zip, php, file, data, http, ftp, compress.bzip2, compress.zlib, https, ftps
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend Technologies

Powered By

Il conviendra de supprimer cette page (ou d'en restreindre l'accès). Les données renvoyées par `phpinfo()` sont très prisées par les personnes à la recherche de vecteurs de nuisances. Pour la même raison, on évitera de créer une URL `info.php` contenant un appel à `phpinfo()`, car cette URL est souvent testée afin d'obtenir des informations sur les version et configurations déployées.

5.2. Configuration

PHP lit sa configuration depuis `/etc/php5/apache2/php.ini`. Par défaut, un certain nombre de paramètres sont plutôt dangereux et méritent d'être revus.

5.2.1. Restrictions

5.2.1.1. Couper PHP

La première et la plus radicale des restrictions possibles est de *désactiver* l'interprétation des scripts PHP globalement sur le serveur avec la directive **engine off**. On pourra les autoriser ensuite dans les *VirtualHosts* par des directives spéciales comme décrit dans Section 5.2.4, « Exceptions ».

5.2.1.2. Masquer PHP

Même si ce n'est pas un moyen de protection valable, on peut toujours masquer la présence de PHP afin de rendre les choses plus difficiles pour les éventuels attaquants. PHP est plutôt verbeux sur sa présence. On trouve même dans les versions récentes, des URL *easter-eggs*¹. On pourra les supprimer en mettant la variable `expose_php`² à `Off`. Cela ne suffit bien sûr pas : quand on a affaire à des fichiers se terminant par `.php`, cela lève le doute assez rapidement... Il faudra donc aussi modifier le *handler* PHP, définit dans `/etc/apache2/mods-available/php5.conf` pour y ajouter d'autres extensions moins « bavardes ». On pourra même demander à Apache de renvoyer tous les fichiers `.html` au moteur PHP. C'est discret, mais le coût en performance peut être important si la proportion pages statiques/pages dynamiques est défavorable sur le serveur.

5.2.1.3. Limiter l'injection de code

Lorsque PHP est activé (globalement, ou sur un *virtualhost*), la dernière chose que nous voulons le voir faire est de charger du code à notre insu. Il est donc conseillé de mettre la valeur de `enable_dl` à `Off`. `allow_url_fopen` permet de charger du code PHP distant en ouvrant un URL (par **include** ou **fopen**). Il faudra aussi désactiver cette possibilité en mettant la directive à `Off`.

5.2.1.4. Variables globales

PHP a toujours été handicapé par la nécessité de maintenir une certaine compatibilité ascendante. Les variables `register_globals`, `register_long_arrays` et `register_argc_argv` font partie de cet héritage. Mais pour déployer des scripts « modernes » et surtout bien écrits, il n'est pas nécessaire (et même dangereux) d'activer ces paramètres. Il faudra donc tous les mettre à `Off`.

5.2.1.5. Supprimer des fonctions

PHP sait faire beaucoup. Parfois un peut trop dans le contexte de scripts web. Il est heureusement possible de désactiver certaines fonctions du langage avec `disable_functions`. Il est probablement plus sûr de désactiver les fonctions suivantes : `chgrp`, `chmod`, `chown`, `diskfree`, `dl`, `escapeshellcmd`, `exec`, `get_current_user`, `getmypid`, `getmyuid`, `getrusage`, `highlight_file`, `ignore_user_abort`, `ini_alter`, `ini_restore`, `ini_set`, `leak`, `link`, `listen`, `login`, `passthru`, `php_undefine_function`, `popen`, `posix_mkfifo`, `posix_setegid`, `posix_seteuid`, `posix_setgid`, `posix_setpgid`, `posix_setsid`, `posix_setuid`, `proc_get_status`, `proc_nice`, `proc_open`, `proc_terminate`, `set_time_limit`, `shell_exec`, `show_source`, `socket_bind`, `socket_listen`, `system` et `virtual`. Attention toutefois : on ne peut configurer ce paramètre

¹Par exemple celui-ci donnant affichant les différents crédits [[http://192.168.17.139/index.php?=php echo md5\('PHPB885F2A0-3C92-11d3-A3A9-4C7B08C10000'\)</a\] concernant le développement de PHP.](http://192.168.17.139/index.php?=<?php echo md5('PHPB885F2A0-3C92-11d3-A3A9-4C7B08C10000');)

²cette variable n'est utilisable que dans `php.ini`.

que dans `php.ini`, et non dans la configuration d'Apache (principale ou *virtualhost*). La valeur affectée sera donc globale.

5.2.1.6. Limiter l'accès au système de fichiers

Sur des serveurs ayant plusieurs *virtualhosts*, il est préférable de restreindre l'accès au filesystem aux secteurs strictement nécessaires. L'utilisateur *alice* qui met ses pages, par exemple, dans `/home/alice/www/` n'a pas besoin d'ouvrir des fichiers se trouvant au dessus de son répertoire web dans le filesystem. Le paramètre `open_basedir` permet de configurer un répertoire au dessus duquel il ne sera pas possible d'ouvrir de fichier. Il conviendra de le spécifier dans chaque *virtualhost*.

`doc_root` permet de spécifier la racine des scripts php, à ne pas confondre avec `open_basedir` qui donne la racine de tous les fichiers accessibles par PHP (éventuellement des fichiers de données ou des fichiers de session).

5.2.1.7. Protection des sessions

Pour des raisons de sécurité, on veillera aussi à déplacer le répertoire utilisé pour stocker les fichiers de session (par défaut, `/var/lib/php5/` sous Ubuntu). Il faudra créer un répertoire propre à chaque *virtualhost*, afin de limiter les éventuels problèmes de vol de session. On utilisera la variable `session.save_path` pour spécifier le nouveau répertoire et on veillera à mettre une valeur située *sous* celle d'`open_basedir`. Dans le cas contraire, PHP ne pourrait sauver les fichiers de session.

5.2.1.8. Upload de fichiers

Selon la destination du serveur, on pourra être tenté de positionner la variable `file_uploads` à `off`. Cela désactivera la possibilité d'envoyer des fichiers en POST vers des scripts PHP.

Si l'on désire conserver cette possibilité, il vaut mieux modifier la variable `upload_tmp_dir` et lui affecter un répertoire accessible depuis le *virtualhost* (donc sous le répertoire pointé par `open_basedir`).

5.2.1.9. safe_mode

Un paramètre regroupe à lui seul plusieurs restrictions améliorant la sécurité : `safe_mode`. Il permet notamment de faire respecter les droits associés aux fichiers (un fichier appartenant à X ne pourra être utilisé par un script appartenant à Y) et d'empêcher la modification des variables d'environnement listées dans le paramètre `safe_mode_protected_env_vars`.

Le paramètre `safe_mode` désactive aussi certaines fonctions, listées [<http://www.php.net/manual/fr/features.safe-mode.functions.php>] dans le manuel de PHP, donc la plupart sont déjà couvertes par la valeur recommandée pour `disable_functions`. Il faudra donc mettre cette valeur à `On` sans hésiter.

5.2.2. Limites

PHP permet de fixer quelques limites pour l'exécution des scripts. Ces limites permettent d'éviter d'avoir un serveur complètement surchargé par l'exécution d'un seul script. En revanche, il n'y a pas de provision possible sur le nombre de scripts exécutables en parallèle, qui potentiellement peut être égal au nombre de processus Apache. Il faut donc garder à l'esprit que ce n'est pas un mécanisme qui permet de se protéger contre les attaques, mais plutôt contre des erreurs de programmations.

`memory_limit` permet de limiter la mémoire maximum utilisable par un script PHP. Il est vivement recommandé de ne pas laisser la valeur par défaut de 128M. Il est difficile d'estimer la bonne valeur pour un script, mais dans la plupart des cas, 10M devraient amplement suffire.

La durée d'exécution d'un script peut être fixée à l'aide de `max_execution_time`. Par défaut, sa valeur est 30 (secondes).

`max_input_time` permet de limiter la durée d'une requête client. Par défaut, il n'y a aucune valeur maximum. Des clients mal configurés pourraient donc mobiliser des scripts trop longtemps. En positionnant cette variable à une valeur différente de -1, on peut limiter ce temps. Il faut cependant noter que ce temps comprend la totalité de la requête cliente, y compris des fichiers « uploadés » en POST. Donc si le script reçoit des fichiers importants (ou sert à faire des tests de bande passante par exemple), il faudra prévoir une valeur suffisamment grande pour les clients lents.

`max_input_time` est donc liée à la variable `upload_max_filesize` qui limite la taille maximale d'un fichier uploadé (2M par défaut).

De même, la taille maximum d'un envoi par POST, fixé dans `post_max_size` (8M par défaut), devra être supérieur à `upload_max_filesize` (pour couvrir l'envoi de fichier) et servira à calibrer la durée maximum de la requête client.

Pour clore la série des limitations, on pourra aussi en citer une qu'il est peut être souhaitable d'appliquer à la configuration Apache. Les scripts PHP incluent souvent des portions de code depuis d'autres fichiers, se terminant en général par `.inc`. Bien que rien n'oblige un développeur à nommer ces fichiers `.inc`, c'est une pratique assez courante³. Le problème avec cette pratique est qu'Apache, par défaut, n'a pas de *handler* (gestionnaire) associé aux extensions `.inc`. Un fichier avec une telle extension fichier sera donc affiché comme n'importe quel autre fichier texte, donnant des informations potentiellement sensibles⁴. On peut demander à Apache de ne pas servir ces fichiers : ils sont inclus directement depuis les scripts côté serveur; inutile donc d'offrir la possibilité aux clients HTTP de voir le contenu de ces fichiers. Il en va de même pour les fichiers avec les extensions `~`, `.bak`, `.swp`, `.old`, `.sav[e]`, etc... qui sont parfois envoyés en FTP en même temps que des fichiers légitimes ou laissés sur place par des développeurs éditant directement sur le serveur.

```
<Files ~ "\.(inc|bak|swp|old|sav[e]?|~)$">
  Order allow,deny
  Deny from all
</Files>
```

5.2.3. Gestion d'erreurs

PHP a un système de gestion d'erreurs qui lui permet de les afficher dans les pages HTML retournées ou de les envoyer vers un fichier spécifique. Il vaudra mieux utiliser cette dernière méthode pour les applications en production : cela permet de donner un minimum d'informations via la page Web tout en permettant aux développeurs de voir les messages d'erreurs. Si l'on désire masquer la présence de PHP c'est aussi une modification recommandée. `log_errors` à On permet d'envoyer les erreurs vers le fichier défini dans `error_log`. On désactivera ensuite les messages d'erreur avec `display_errors` et `display_startup_errors` à Off.

5.2.4. Exceptions

Dans le cas où un paramètre doit avoir une valeur spéciale pour un *VirtualHost* particulier, il est possible d'affecter des valeurs aux variables de configuration PHP dans une section `<VirtualHost ...>` avec la directive `php_admin_value`. Par exemple, si le *VirtualHost* d'Alice doit pouvoir utiliser l'upload de fichiers et nécessite 50 Mb de mémoire pour son exécution, on pourra corriger la valeur de `file_uploads` et `memory_limit` uniquement pour lui :

```
root@ubuntu:~# cat /etc/apache2/sites-available/site_alice
<VirtualHost *>
  ServerAdmin alice@example.org
  ServerName alice.example.org
```

³Cette pratique pourrait être avantageusement remplacée par une autre consistant à nommer ces fichiers en `.php` et à les placer dans un sous répertoire `/include` par exemple.

⁴On pourra s'en convaincre en visitant, par exemple, cette adresse <http://www.google.com/search?q=filetype%3Ainc+inurl%3Aconfig+%22dbname%22>

```
DocumentRoot /home/alice/monsieuebe/
```

```
php_admin_flag engine on  
php_admin_flag file_uploads on  
php_admin_value memory_limit 50M  
...
```

On privilégiera systématiquement **php_admin_value** au lieu de **php_value** pour changer les valeurs de configuration de PHP dans la configuration Apache. Cette dernière directive autorise le changement des valeurs dans un fichier `.htaccess` tandis que **php_admin_value** la fixe sans aucune possibilité de modification.

Seules les valeurs `expose_php`, `disable_functions` et `disable_classes` ne peuvent pas être modifiées dans la configuration d'Apache (voir <http://fr.php.net/manual/fr/ini.php>).

Chapitre 6. Déploiement et guide des opérations MySQL

\$Revision: 1.16 \$

\$Date: 2007/07/07 14:26:00 \$

MySQL a commencé comme une toute petite base de données, avec un sous ensemble du langage SQL assez limité. Mais sa simplicité d'utilisation en regard des autres bases de données du marché (PostgreSQL pour l'OpenSource, Oracle/DB2/Informix/Sybase pour les SGBD commerciaux) et ses performances sur les petites bases en ont fait un produit de choix. Une décennie plus tard, MySQL possède toutes les caractéristiques d'une base de données professionnelle, et notamment les fonctionnalités qui lui manquaient cruellement pour y parvenir : triggers, procédures stockées et transactions. Ce chapitre détaille les points clef du déploiement d'un serveur MySQL :

- installation,
- configuration,
- gestion des droits,
- sauvegarde des bases

6.1. Installation

L'installation de MySQL s'accompagne généralement de toute une suite de bibliothèques clientes, de bibliothèques perl, de dépendances... Le paquetage MySQL d'Ubuntu s'illustre particulièrement dans ce domaine, puisque l'installation du serveur provoque l'arrivée de 8 paquets en tout, pour une taille totale approchant les 100 Mo :

```
root@ubuntu:~# apt-get install mysql-server-5.0
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les paquets supplémentaires suivants seront installés :
  libdbd-mysql-perl libdbi-perl libmysqlclient15off
  libnet-daemon-perl
  libplrpc-perl mysql-client-5.0 mysql-common
Paquets suggérés :
  dbshell libcompress-zlib-perl tinyca
Paquets recommandés :
  mailx
Les NOUVEAUX paquets suivants seront installés :
  libdbd-mysql-perl libdbi-perl libmysqlclient15off
  libnet-daemon-perl
  libplrpc-perl mysql-client-5.0 mysql-common mysql-server-5.0
0 mis à jour, 8 nouvellement installés, 0 à enlever et 8 non mis à
jour.
Il est nécessaire de prendre 0o/35,9Mo dans les archives.
Après dépaquetage, 93,6Mo d'espace disque supplémentaires seront
utilisés.
Souhaitez-vous continuer [O/n] ? O
Préconfiguration des paquets...
Sélection du paquet mysql-common précédemment désélectionné.
(Lecture de la base de données... 16199 fichiers et répertoires
déjà installés.)
Dépaquetage de mysql-common (à partir de
.../mysql-common_5.0.38-0ubuntul_all...
```

```
Sélection du paquet libnet-daemon-perl précédemment désélectionné.  
Dépaquetage de libnet-daemon-perl (à partir de  
.../libnet-daemon-perl_0.38-1....  
Sélection du paquet libplrpc-perl précédemment désélectionné.  
Dépaquetage de libplrpc-perl (à partir de  
.../libplrpc-perl_0.2017-1.1_all.de...  
Sélection du paquet libdbi-perl précédemment désélectionné.  
Dépaquetage de libdbi-perl (à partir de  
.../libdbi-perl_1.53-1build1_i386.deb...  
Sélection du paquet libmysqlclient15off précédemment désélectionné.  
Dépaquetage de libmysqlclient15off (à partir de  
.../libmysqlclient15off_5.0.3...  
Sélection du paquet libdbd-mysql-perl précédemment désélectionné.  
Dépaquetage de libdbd-mysql-perl (à partir de  
.../libdbd-mysql-perl_3.0008-1b...  
Sélection du paquet mysql-client-5.0 précédemment désélectionné.  
Dépaquetage de mysql-client-5.0 (à partir de  
.../mysql-client-5.0_5.0.38-0ubu...  
Paramétrage de mysql-common (5.0.38-0ubuntu1) ...  
Sélection du paquet mysql-server-5.0 précédemment désélectionné.  
(Lecture de la base de données... 16415 fichiers et répertoires  
déjà installés.)  
Dépaquetage de mysql-server-5.0 (à partir de  
.../mysql-server-5.0_5.0.38-0ubu...  
Paramétrage de libnet-daemon-perl (0.38-1.1) ...  
Paramétrage de libplrpc-perl (0.2017-1.1) ...  
Paramétrage de libdbi-perl (1.53-1build1) ...  
Paramétrage de libmysqlclient15off (5.0.38-0ubuntu1) ...  
  
Paramétrage de libdbd-mysql-perl (3.0008-1build1) ...  
Paramétrage de mysql-client-5.0 (5.0.38-0ubuntu1) ...  
Paramétrage de mysql-server-5.0 (5.0.38-0ubuntu1) ...  
* Stopping MySQL database server mysqld  
  [ OK ]  
* Starting MySQL database server mysqld  
  [ OK ]  
* Checking for corrupt, not cleanly closed and upgrade needing  
tables.  
  
root@ubuntu:~#
```

6.2. Gérer le service

A l'instar de tous les services du système, on peut invoquer le script de démarrage (`/etc/init.d/mysql`) directement ou via la commande **invoke-rc.d**.

MySQL version 5.0 installe aussi les services nécessaires à la gestion d'un cluster (**ndbd** et **ndb_mgmd**). Même si les scripts de démarrage pour ces deux services sont activés pour le runlevel par défaut (2), les conditions pour qu'ils démarrent ne sont pas remplies¹. Si ces deux services ne sont pas requis, il vaut de toutes façon mieux les désactiver au boot avec l'option `-f` et le paramètre `remove` de la commande **update-rc.d** :

```
root@ubuntu:~# update-rc.d -f mysql-ndb remove  
Removing any system startup links for /etc/init.d/mysql-ndb ...  
  /etc/rc0.d/K22mysql-ndb  
  /etc/rc1.d/K22mysql-ndb
```

¹Il manque en effet le fichier de configuration pour **ndb_mgmd** et une ligne de configuration `ndb-connectstring` pour **ndbd**.

```
/etc/rc2.d/S18mysql-ndb
/etc/rc3.d/S18mysql-ndb
/etc/rc4.d/S18mysql-ndb
/etc/rc5.d/S18mysql-ndb
/etc/rc6.d/K22mysql-ndb
root@ubuntu:~# update-rc.d -f mysql-ndb-mgm remove
Removing any system startup links for /etc/init.d/mysql-ndb-mgm
...
/etc/rc0.d/K23mysql-ndb-mgm
/etc/rc1.d/K23mysql-ndb-mgm
/etc/rc2.d/S17mysql-ndb-mgm
/etc/rc3.d/S17mysql-ndb-mgm
/etc/rc4.d/S17mysql-ndb-mgm
/etc/rc5.d/S17mysql-ndb-mgm
/etc/rc6.d/K23mysql-ndb-mgm
root@ubuntu:~#
```

6.3. Notions de base

MySQL est un SGBD particulièrement simple à mettre en œuvre, mais quelques notions sont nécessaires afin d'en saisir son fonctionnement.

6.3.1. Fichier de configuration

Le fichier de configuration de MySQL est traditionnellement `my.cnf` qui, sous Ubuntu, est situé dans `/etc/mysql`. Dans les versions récentes (> 5.0.4b), MySQL peut aussi inclure des fichiers de configuration depuis le fichier de configuration principal. A l'instar d'Apache, on trouve donc un sous-répertoire `conf.d` dans lequel on peut déposer des « bouts » de configuration complémentaires.

Le fichier de configuration contient des paramètres système, mais les différentes bases de données et les utilisateurs MySQL sont directement stockés dans une base de données spécifique appelée *mysql*.

6.3.2. Utilisateurs

MySQL possède ses propres utilisateurs, complètement distincts du système. Même si MySQL possède par défaut un utilisateur *root*, il n'a rien avoir avec le *root* du système. L'intérêt est de pouvoir décorreler la responsabilité du système de celle de la base de données, qui est habituellement gérée par un *DBA*.

6.3.3. Bases de données

Lorsque des données doivent être stockées sur un SGBD, elles le sont dans des tables représentant des données d'un type particulier. Par exemple, une table pourra contenir une liste de *communes*, une autre une liste de *personnes*. Dans ce cas, une commune occupera une « ligne » (appelé aussi un *enregistrement*) dans la table des communes et un individu occupera un enregistrement dans la table des personnes.

Toutes les données qui ont un *lien* entre elles seront regroupées au sein d'une même base de données. Par exemple, les entrées de la table *personnes* peuvent être liées à une entrée de la table *communes* par leur adresse et seront donc logiquement dans la même base de données.

MySQL stocke ses bases de données sous `/var/lib/mysql/`. Si des optimisations du système de fichier (journalisation, `noatime`, utilisation du RAID, etc...) sont requises, il faudra donc cibler la branche `/var/lib/mysql/`. Il n'est en revanche pas recommandé de sauvegarder les bases telles qu'elles : on obtiendrait probablement des données sauvegardées corrompues car modifiées en cours de backup. On privilégiera la sauvegarde à froid (service MySQL arrêté) ou la sauvegarde d'un *dump* (export SQL) de la base (voir Section 6.7, « Sauvegarde et restauration de bases »).

6.3.4. Outils

Par défaut, les outils `mysql` (**mysql**, **mysqladmin**, **mysqldump**, ...) vont tenter de se connecter à la base de données « en tant que » l'utilisateur unix courant. Par exemple, si l'utilisateur Unix *root* utilise la commande **mysqladmin**, cette dernière utilisera les droits de l'utilisateur MySQL *root*.² On pourra spécifier un nom d'utilisateur à tous les outils MySQL en utilisant l'option `-u`.

Deux autres options sont à noter. Tout d'abord, `-p` permet de spécifier le mot de passe pour la connexion. C'est à éviter en ligne de commande, mais parfois obligatoire dans les scripts. On veillera alors à mettre des droits restreints afin que n'importe qui ne puisse pas lire le script et obtenir le mot de passe. Si `-p` n'est pas suivi par le mot de passe, l'outil le demandera de manière interactive. On privilégiera cet usage afin de ne pas laisser de trace dans l'historique de commandes³.

Enfin `-h` permet de spécifier le nom de l'hôte auquel l'on veut se connecter.

- **mysql** : ligne de commande `mysql` qui permet d'exécuter des requêtes SQL dans la base de données. S'utilise soit comme un *shell* (on l'invoke, puis on saisit des commandes), soit en lui envoyant des ordres SQL depuis l'entrée standard (via l'opérateur de redirection du shell « < » ou par le biais d'un pipe).
- **mysqladmin** : utilitaire d'administration `mysql`, permettant de créer/supprimer des utilisateurs, changer des mots de passe, créer/supprimer des bases de données, démarrer/stopper le serveur, ...
- **mysqldump** : permet de faire un export d'une base de données (et/ou de sa structure).
- **mysqlshow** : permet d'afficher des informations sur une base, les tables d'une base, etc...

6.3.5. SQL par l'exemple en 3 minutes

Le langage *SQL* permet de lire, modifier et supprimer des données dans une base de données relationnelle. Quelques opérations de base sont données ici. On se reportera à la masse de tutoriaux disponibles en ligne pour approfondir le sujet. C'est un langage relativement simple dans la mesure où il est assez proche du langage naturel.

6.3.5.1. SELECTIONner des données

La construction **SELECT** permet d'obtenir une liste de colonnes d'enregistrements issus d'une table. Prenons par exemple la table suivante :

Tableau 6.1. Exemple SQL : Table des Génies

Nom	Prenom	AnneeNaissance
Cox	Alan	1968
Knuth	Donald	1938
Tovalds	Linus	1969
Turing	Alan	2912
Gates	Bill	1955
Einstein	Albert	1879

Nous pourrions demander la liste de tous les enregistrements de la table avec la requête⁴ :

²Voilà probablement pourquoi les développeurs de MySQL ont choisi de nommer *root* l'utilisateur ayant par défaut le rôle de DBA *mysql* : cela simplifie les opérations de maintenance.

³On pourra dans tous les cas supprimer l'ajout des commandes exécutées dans le shell courant par `history -r`; dans certaines versions antérieures, les curieux pouvaient aussi voir le mot de passe au moment où la commande était exécutée par un **ps** bien placé.

⁴Par convention, les mots clefs SQL sont écrits en majuscules mais ce n'est pas une obligation.

```
SELECT * FROM Genies;
```

Si nous ne voulons que les prénoms, nous pourrions faire :

```
SELECT Prenom FROM Genies;
```

Si nous voulons nom et prénom, il suffit de faire :

```
SELECT Nom,Prenom FROM Genies;
```

Maintenant, nous ne voulons que les génies qui se prénomment Alan, nous utiliserions la clause **WHERE**, qui permet de *filtrer* les enregistrements sélectionnés :

```
SELECT * FROM Genies WHERE Prenom='Alan';
```

Nous pouvons chercher les génies nés avant le XXème siècle :

```
SELECT * FROM Genies WHERE AnneeNaissance <= 1900;
```

Nous pourrions aussi combiner plusieurs « filtres » et demander la liste des 'Alan' nés après 2000 :

```
SELECT * FROM Genies WHERE Prenom='Alan' AND AnneeNaissance >= 2000;
```

Simple !

Le caractère « % » sert de *wildcard* (joker) en SQL : il permet de remplacer n'importe quoi. Mais si on l'utilise, on doit remplacer « = » par **LIKE** :

```
SELECT * FROM Genies WHERE Prenom LIKE 'A%';
```

permet de sélectionner tous les enregistrements dont la première lettre du champ « Prenom » est « A ».

6.3.5.2. Mettre à jour (UPDATE) les données

Le mot clef **UPDATE** permet de mettre à jour les enregistrements correspondant à la clause **WHERE** désirée (ou tous les enregistrements s'il n'y a pas de clause *where*).

Dans la table exemple, nous nous sommes visiblement trompés de siècle pour Alan Turing. Pas de problème :

```
UPDATE Genies SET AnneeNaissance=1912 WHERE Prenom='Alan' AND Nom='Turing';
```

ou encore plus simple, si c'est le seul enregistrement ayant 2142 comme année de naissance :

```
UPDATE Genies SET AnneeNaissance=1912 WHERE AnneeNaissance=2142;
```

Idem pour changer tous les 'Bill' en 'William' :

```
UPDATE Genies SET Prenom='William' WHERE Prenom='Bill';
```

6.3.5.3. Supprimer (DELETE) des enregistrements

Toujours sur le même schéma, on pourra supprimer des enregistrements correspondant à une clause **WHERE** avec le mot clef **DELETE**. Sans clause **WHERE**, comme d'habitude, tous les enregistrements seront affectés.

Nous pouvons très facilement supprimer les intrus de notre table :

```
DELETE FROM Genies WHERE Prenom='William' AND Nom='Gates';
```

6.3.5.4. Autres commandes

D'autres commandes sont utiles en ligne de commande mysql. Ces commandes sont généralement achevées par « ; », indiquant à MySQL que la commande est terminée et qu'il peut l'appliquer. On peut par exemple changer la base de données en cours avec **USE** :

```
USE mysql;
```

On peut aussi effacer complètement une table avec **DROP TABLE** :

```
DROP TABLE Genies;
```

voire une base de données complète avec **DROP DATABASE** :

```
DROP DATABASE test;
```

Pour voir les bases de données existantes, on utilisera **SHOW DATABASES** :

```
SHOW DATABASES;
```

et pour voir les tables déclarées dans la base de données actuellement sélectionnée on utilisera **SHOW TABLES** :

```
SHOW TABLES;
```

Enfin, **FLUSH PRIVILEGES** permet de demande à MySQL d'actualiser les droits des utilisateurs :

```
FLUSH PRIVILEGES;
```

6.4. Configuration initiale

6.4.1. Utilisateurs

Par défaut, MySQL 5 sous Ubuntu est configuré avec 2 utilisateurs : `root` et `debian-sys-maint`. Le premier sert à l'administration du SGBD : création des bases, création des utilisateurs, etc... Le deuxième en revanche sert au démarrage, à l'arrêt et à la mise à jour des packages de la base. `root` n'a pas de mot de passe par défaut. La première des choses à faire après l'installation de la base est donc de lui en affecter un. `mysqladmin` permet de changer des mots de passe, mais aussi de créer/supprimer des bases de données, de démarrer/stopper un serveur, ...

Pour modifier le mot de passe d'un utilisateur MySQL il y a plusieurs possibilités, mais nous n'en verrons que deux. Nous pouvons le faire directement en ligne de commande soit en ligne de commande bash (voir les remarques à ce sujet dans Section 6.3.4, « Outils ») :

```
root@ubuntu:~# mysqladmin password "nouveau mot de passe" ❶
```

```
root@ubuntu:~# mysqladmin password "nouveau mot de passe" ❷
```

```
mysqladmin: connect to server at 'localhost' failed  
error: 'Access denied for user 'root'@'localhost' (using password:  
NO)' ❸
```

```
root@ubuntu:~# history -r
```

```
root@ubuntu:~#
```

- ❶ modification du mot de passe
- ❷ une fois le mot de passe modifié, il faut utiliser `-p` !
- ❸ relecture de l'historique

Une autre solution est de le faire en ligne de commande MySQL :

```
root@ubuntu:~# mysql ❶
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> SET PASSWORD FOR
  root@localhost=PASSWORD('nouveau mot de passe'); ❷
Query OK, 0 rows affected (0.12 sec)

mysql> FLUSH PRIVILEGES; ❸
Query OK, 0 rows affected (0.02 sec)

mysql> QUIT ❹
Bye
root@ubuntu:~# cat /dev/null > ~/.mysql_history ❺

root@ubuntu:~#
```

- ❶ démarrage du shell MySQL.
- ❷ modification du mot de passe pour l'utilisateur *root* se connectant depuis *localhost*.
- ❸ `flush privileges` permet de demander au serveur MySQL de relire les droits.
- ❹ fermeture du shell MySQL
- ❺ le shell `mysql` aussi a son historique... (voir Section 6.4.4, « En finir avec l'historique » pour une solution définitive)

Pour plus de sécurité, on peut aussi modifier le nom du « super-utilisateur » MySQL. Par défaut c'est *root*, mais une fois de plus, rien ne nous oblige à utiliser ce compte. Pour changer le nom de l'utilisateur, il faudra faire les modifications directement en SQL dans la table système (table `mysql`).

```
root@ubuntu:~# mysql -p ❶

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> USE mysql; ❷

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> UPDATE user SET user='dbadmin' WHERE user='root'; ❸

Query OK, 3 rows affected (0.11 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> FLUSH PRIVILEGES; ❹

Query OK, 0 rows affected (0.01 sec)

mysql> QUIT
Bye
```

```
root@ubuntu:~# mysql -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost'
(using password: YES) ⑤

root@ubuntu:~# mysql -u dbadmin -p

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g. ⑥

Your MySQL connection id is 39
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> QUIT
Bye
root@ubuntu:~# cat /dev/null > ~/.mysql_history
```

- ① démarrage du shell MySQL.
- ② `use` permet de changer la base de données en cours. `use mysql;` indique que l'on veut utiliser la base de données `mysql`. On aurait pu aboutir au même résultat en démarrant le shell `mysql` avec `mysql` en option (`mysql mysql -p`).
- ③ `update` est une commande SQL permettant de modifier des enregistrements dans une table. Ici, on demande de changer la valeur de `user` en `dbadmin` pour tous les enregistrements de la table `user` dont le champ `user` vaut `root`.
- ④ `flush privileges` permet de demander au serveur MySQL de relire les droits.
- ⑤ même avec le bon mot de passe, `mysql` nous refuse : l'utilisateur `root` n'existe plus. Nous devons donc spécifier l'utilisateur avec l'option `-u`.
- ⑥ avec le bon utilisateur, pas de problème

Selon les version de MySQL et les OS qui les packagent, on trouve parfois un utilisateur *anonyme* configuré par défaut⁵. Même si cet utilisateur n'a en général pas de droits, il est préférable de le supprimer afin d'éliminer un vecteur potentiel d'intrusion dans la base de données. Il faudra donc se connecter à la base `mysql` et supprimer l'utilisateur " (dont le nom est vide) dans la table `user`.

Attention : les utilisateurs peuvent apparaître plusieurs fois dans la table `user`, puisque la *clef* de cette table est le couple (*utilisateur, hôte de connexion*).

```
root@ubuntu:~# mysql -u dbadmin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> SHOW DATABASES;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| test                    |
+-----+
3 rows in set (0.03 sec)
```

⁵Ce n'est pas le cas sous la Ubuntu 7.04 serveur


```
mysql> USE mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
mysql> DELETE FROM user WHERE user='';
Query OK, 1 row affected (0.00 sec)
```

```
mysql> DELETE FROM db WHERE user='';
Query OK, 2 rows affected (0.01 sec)
```

```
mysql>
```

Enfin, on pourra faire une dernière passe sur la base de données système *mysql* afin de s'assurer que seuls les droits nécessaires sont activés.

```
mysql> DELETE FROM db; ❶
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> DELETE FROM user WHERE NOT (host="localhost" AND
  (user="dbadmin" OR user="debian-sys-maint")); ❷
Query OK, 3 rows affected (0.00 sec)
```

```
mysql> FLUSH PRIVILEGES; ❸
Query OK, 0 rows affected (0.00 sec)
```

```
mysql>
```

- ❶ suppression de toutes les règles de la table *db*.
- ❷ suppression de tous les utilisateurs qui ne sont pas *root* ou *debian-sys-maint*.
- ❸ actualisation des droits.

6.4.2. Bases de données

Les opérations de création et suppression peuvent aussi s'effectuer de deux façons : en ligne de commande bash, ou en ligne de commande mysql. La première méthode utilise encore **mysqladmin** avec les arguments *create* ou *drop* :

```
root@ubuntu:~# mysqladmin create mabase -u dbadmin -p ❶
Enter password:
root@ubuntu:~# mysqlshow -u dbadmin mabase -p ❷
Enter password:
Database: mabase
+-----+
| Tables |
+-----+
+-----+
root@ubuntu:~#
```

- ❶ création de la base en ligne de commande.
- ❷ **mysqlshow** permet d'obtenir des informations sur des bases, des tables et des champs.

En cas de regrets **mysqladmin drop** permet de supprimer la base :

```
root@ubuntu:~# mysqladmin drop mabase -u dbadmin -p
Enter password:
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.
```

```
Do you really want to drop the 'mabase' database [y/N] y
Database "mabase" dropped
root@ubuntu:~# mysqlshow -u dbadmin mabase -p
Enter password:
mysqlshow: Unknown database 'mabase'
root@ubuntu:~#
```

Si l'on préfère la version ligne de commande mysql, on fera :

```
root@ubuntu:~# mysql -u dbadmin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> CREATE DATABASE mabase;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mabase |
| mysql |
| test |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> DROP DATABASE mabase;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.00 sec)
```

```
mysql>
```

On pourra pratiquer en réel sur la base *test*. Cette base est livrée avec toutes les versions de MySQL et apparemment sous tous les OS supportés. Comme pour l'utilisateur *anonyme*, il est préférable de s'en débarrasser : elle n'apporte rien sinon un vecteur potentiel pour un attaquant. Cela pourra éventuellement ennuyer quelques modules Perl liés à MySQL qui testent parfois leur bon fonctionnement en utilisant la base *test* comme cobaye. Dans ce cas, il faudra forcer leur installation ou recréer la base *test*.

```
mysql> DROP DATABASE test;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SHOW DATABASES;
+-----+
```

```
| Database |
+-----+
| information_schema |
| mysql |
+-----+
2 rows in set (0.00 sec)
```

```
mysql>
```

6.4.3. my.cnf

Une commande particulièrement dangereuse sous MySQL est **LOAD DATA INFILE**. Elle permet d'insérer le contenu d'un fichier dans une base de données. Si, via une application PHP, une attaquante arrive à injecter du code SQL, elle pourra éventuellement remplir des tables avec des informations sensibles :

```
root@ubuntu:~# mysql -u dbadmin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> LOAD DATA LOCAL INFILE "/etc/passwd" INTO TABLE mydb.res;
Query OK, 29 rows affected (0.07 sec)
Records: 29  Deleted: 0  Skipped: 0  Warnings: 0
```

```
mysql> SELECT * FROM res WHERE texte LIKE 'root%';
+-----+
| texte |
+-----+
| root:x:0:0:root:/root:/bin/bash |
+-----+
1 row in set (0.00 sec)
```

```
mysql>
```

On voit les problèmes que ce genre de commande peut amener : une exposition de la totalité des fichiers de configuration système : récupération des identifiants utilisateur sur le système afin de préparer une attaque par force brute, escalade de privilèges en récupérant les informations d'authentification dans `/etc/mysql/debian.cnf`, etc... On veillera donc à supprimer l'accès à **LOCAL INFILE** dans le fichier de configuration en ajoutant `set-variable = local-infile=0` dans la section `[mysqld]`. Après avoir redémarré la base de données, il faudra vérifier que la configuration est correctement appliquée :

```
root@ubuntu:~# mysql -u dbadmin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> LOAD DATA LOCAL INFILE "/etc/passwd" INTO TABLE mydb.res;
ERROR 1148 (42000): The used command is not allowed with this MySQL
version
mysql>
```

Par défaut sous Ubuntu la directive `bind-address = 127.0.0.1` est dans le fichier de configuration par défaut. Elle permet de restreindre l'accès au port 3306 à la machine locale (127.0.0.1). On devra commenter cette directive uniquement si l'accès au serveur MySQL est requis depuis une autre machine. Si le serveur possède plusieurs interfaces, on ne « bindera » le serveur que sur l'interface nécessaire. Il est aussi possible de supprimer complètement l'accès au serveur MySQL via le réseau avec la directive `skip-networking`. Dans ce cas les clients devront utiliser la socket unix `/var/run/mysqld/mysqld.sock` pour se connecter à la base.

6.4.4. En finir avec l'historique

Comme on a pu le remarquer plus haut, le shell MySQL conserve un historique, à l'image de `bash`, dans le fichier `~/.mysql_history`. On peut demander à `mysql` d'utiliser un autre fichier grâce à la variable d'environnement `MYSQL_HISTFILE` (tout comme `HISTFILE` dans le cas de `bash`).

Pour supprimer l'historique, il y a deux possibilités :

- mettre « `/dev/null` » dans la variable `MYSQL_HISTFILE` (`export MYSQL_HISTFILE="/dev/null"` dans le fichier `~/.bashrc` par exemple),
- lier `~/.mysql_history` à `/dev/null` (`ln -sf /dev/null ~/.mysql_history`).

6.4.5. Filtrage

La structure du filtrage étant déjà en place, l'ouverture de l'accès à la base MySQL est très simple. On n'ouvrira l'accès au port que si cela est réellement nécessaire, et, si possible, uniquement aux adresses IP qui le nécessitent. Il n'y a pas de raison d'ouvrir l'accès IP au serveur de bases de données à tout le monde.

Exemple 6.1. MySQL : configuration du filtrage TCP en entrée

```
#
# #####
# TCP entrant
# Il faudra ouvrir des ports au fil de l'eau
# lors de la mise en place de
# services TCP (ssh, apache, ...).
# #####
#
-A TCP_IN -j TCP_INLIMITS
-A TCP_IN -j STATEFUL
-A TCP_IN -j TCP_SYNLIMITS
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_IN:" --log-level 6
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
# Ajouter les règles ici lors de l'installation de services TCP si
# ces services
# doivent être ouverts
#
-A TCP_IN -s adresse_ip_autorisée -p tcp -m tcp --dport 3306 -j
  ACCEPT ❶
# on peut aussi débloquer le port 3306 pour tout le monde
-A TCP_IN -p tcp -m tcp --dport 3306 -j ACCEPT ❷
#
```

- ❶ Règle autorisant l'accès au port 3306/tcp (mysql) pour l'adresse `adresse_ip_autorisée` (qui peut aussi être un subnet). Cette règle peut être répétée autant de fois que nécessaire.
- ❷ Règle autorisant l'accès au port 3306/tcp (mysql) pour tout le monde.

6.5. Gestion des droits

La gestion des droits (ajout/suppression d'utilisateur, modification de droits) sous MySQL se fait entièrement en SQL. MySQL stocke les utilisateurs et leurs droits dans la base *mysql*, et les tables *user*, *db* et *host*. La détermination des droits en fonction de ces tables est un peu complexe à détailler mais si l'on utilise les commandes **GRANT** et **REVOKE**, cela reste très simple.

6.5.1. GRANT

La commande **GRANT** permet d'ajouter des droits. Lorsque j'ajoute des droits à un utilisateur inexistant, **GRANT** créera cet utilisateur

Sa syntaxe générale est :

```
GRANT quoi ON sur_quoi TO qui@depuis_ou IDENTIFIED BY mot_de_passe
```

Par exemple, pour donner l'autorisation à *alice* de lire (c'est à dire faire des **SELECT**) sur la table *Genies* de la base *mabase*, on devra saisir :

```
GRANT SELECT ON mabase.Genies TO alice@localhost IDENTIFIED BY  
'lepass'
```

Si l'on souhaite donner un accès lecture/création/mise à jour d'enregistrement, on utilisera :

```
GRANT SELECT,INSERT,UPDATE ON mabase.Genies TO alice@localhost  
IDENTIFIED BY 'lepass'
```

On pourra être moins spécifique si nécessaire. Dans les exemples précédent, les droits alloués concernaient la table *Genies* de la base *mabase*. Si l'on souhaite appliquer ces droits à toutes les tables de *mabase*, on utilisera :

```
GRANT SELECT,INSERT,UPDATE ON mabase.* TO alice@localhost  
IDENTIFIED BY 'lepass'
```

De même, si l'on souhaite appliquer ces droits pour Alice sur toutes les bases du système, on utilisera :

```
GRANT SELECT,INSERT,UPDATE ON *.* TO alice@localhost IDENTIFIED BY  
'lepass'
```

Attention cependant à la table *mysql*... Ces commandes ne permettront à Alice de se connecter que localement (*localhost*). Si Alice doit se connecter depuis une machine distante, il faudra remplacer *localhost* par *nom_machine* ou *ip_machine*. Bien sûr, le filtrage (Section 6.4.5, « Filtrage ») et le *binding* (Section 6.4.3, « *my.cnf* ») devront être adaptés.

```
root@ubuntu:~# mysql -u alice -p ❶  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 7  
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql> USE mabase  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed
```

```
mysql> SELECT * from Genies WHERE Nom='Turing';
+-----+-----+-----+
| Nom    | Prenom | AnneeNaissance |
+-----+-----+-----+
| Turing | Alan   |          2912   |
+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> UPDATE Genies SET AnneeNaissance=1912 WHERE Nom='Turing'; ❷
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> DELETE FROM Genies WHERE Nom='Gates'; ❸
ERROR 1142 (42000): DELETE command denied to user
'alice'@'localhost' for table 'Genies'
mysql>
```

- ❶ Alice peut effectuer des **SELECT**,
- ❷ Alice peut effectuer des **UPDATE**,
- ❸ mais Alice ne peut pas effectuer de **DELETE**.

6.5.2. REVOKE

La commande **REVOKE** permet de supprimer des droits. Lorsque je supprime le dernier droit à un utilisateur, il n'est *pas* supprimé. Il faudra le supprimer explicitement de la table *user* (base *mysql*) si nécessaire. **GRANT** créera cet utilisateur

Sa syntaxe générale, presque symétrique à celle de **GRANT**⁶ est :

```
REVOKE quoi ON sur_quoi FROM qui@depuis_ou
```

Par exemple, pour supprimer à Alice la possibilité de mettre à jour des enregistrements, on utilisera la commande suivante :

```
REVOKE UPDATE ON mabase.Genies FROM alice@localhost
```

Si l'on reprend la connexion ci-dessus, une tentative d'**UPDATE** par Alice se traduirait par un refus :

```
mysql> UPDATE Genies SET AnneeNaissance=1912 WHERE Nom='Turing';
ERROR 1142 (42000): UPDATE command denied to user
'alice'@'localhost' for table 'Genies'
mysql>
```

6.5.3. Visualisation des droits

La commande **SHOW GRANTS** permet de visualiser les droits dont disposent les utilisateurs de la base. Sans arguments, cette commande renvoie les droits de l'utilisateur qui l'invoque. Si on l'accompagne de **FOR**, on pourra spécifier pour quel utilisateur on souhaite visualiser les droits.

```
mysql> SHOW GRANTS;
+-----+
-----+
-----+
-----+
| Grants for dbadmin@localhost
|
|
```

⁶attention cependant à l'erreur classique qui consiste à oublier de substituer **TO** par **FROM**

```
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'dbadmin'@'localhost' IDENTIFIED
| BY PASSWORD '*196BDEDE2AE4F84CA44C47D54D78478C7E2BD7B7' WITH GRANT
| OPTION |
+-----+
-----+
1 row in set (0.00 sec)

mysql> SHOW GRANTS FOR alice@localhost;
+-----+
| Grants for alice@localhost
|
+-----+
| GRANT SELECT, INSERT, UPDATE ON *.* TO 'alice'@'localhost'
| IDENTIFIED BY PASSWORD '*4F5CCA657BD61D1C1127E5C4EA3B0EE4A9841B85'
|
| GRANT SELECT, INSERT ON `mabase`.`Genies` TO 'alice'@'localhost'
|
+-----+
2 rows in set (0.00 sec)
mysql>
```

6.5.4. Droits utilisables avec GRANT et REVOKE

La liste des droits utilisables avec **GRANT** et **REVOKE** (le *quoi*) est assez importante. Chaque droit porte généralement le nom de la commande SQL autorisée (ou révoquée). On se reportera à la documentation MySQL (voir sur [MySQL] le chapitre [<http://dev.mysql.com/doc/refman/5.0/fr/grant.html>] concernant **GRANT**) pour un tableau exhaustif, mais les plus importantes sont :

- **SELECT** : « lire » des enregistrements
- **INSERT** : insérer des enregistrements
- **UPDATE** : mettre à jour des enregistrements existants
- **DELETE** : supprimer des enregistrements
- **CREATE** : créer des tables
- **DROP** : détruire des tables
- **GRANT** : donner ou supprimer des droits

Il faudra ici aussi appliquer le principe du *privilège minimum* : inutile de donner accès à toutes les bases si l'utilisateur n'a besoin d'accéder qu'à la sienne. De même, inutile de donner le droit **DELETE** si l'utilisateur n'a besoin que de lire la base, et inutile de donner des droits sur des tables d'une base quand ce n'est pas nécessaire.

6.6. Perte des identifiants

En cas de perte des identifiants du « super-utilisateur » de la base, la solution la plus simple consiste à utiliser les paramètres d'authentification de *debian-sys-maint* que l'on trouvera dans le fichier `/etc/mysql/debian.cnf` et de changer celui du « super-utilisateur » avec :

```
SET PASSWORD FOR 'dbadmin'@'localhost' =  
PASSWORD('nouveaumotdepasse');
```

Une autre solution, plus portable (la solution précédente ne fonctionne que sur Ubuntu/Debian), consiste à démarrer le serveur avec un fichier SQL d'initialisation dans lequel nous mettrons la commande ci-dessus. Ce fichier d'initialisation sera exécuté sans aucune restriction de droit.

```
root@ubuntu:~# invoke-rc.d mysql stop  
* Stopping MySQL database server mysqld  
[ OK ]  
root@ubuntu:~# echo "SET PASSWORD FOR 'dbadmin'@'localhost' =  
PASSWORD('nouveaumotdepasse');" > /root/resetpass.sql  
root@ubuntu:~# mysqld_safe --init-file=/root/resetpass.sql &  
[1] 7036  
root@ubuntu:~# Starting mysqld daemon with databases from  
/var/lib/mysql  
mysqld_safe[7075]: started  
  
root@ubuntu:~# invoke-rc.d mysql restart  
* Stopping MySQL database server mysqld  
  
STOPPING server from pid file /var/run/mysqld/mysqld.pid  
mysqld_safe[7141]: ended  
  
[ OK ]  
* Starting MySQL database server mysqld  
[ OK ]  
* Checking for corrupt, not cleanly closed and upgrade needing  
tables.  
[1]+ Done mysqld_safe  
--init-file=/root/resetpass.sql  
root@ubuntu:~# mysql -u dbadmin -p  
Enter password: xxxx  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 6  
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql>
```

Le dernière solution consiste à démarrer mysql avec les options `--skip-grant-tables` et `--user=root`, de se connecter à la base avec **mysql** puis de changer le mot de passe. L'option `--skip-grant-tables` demande au serveur de ne pas tenir compte des tables d'authentification. Il fait noter que cette méthode est dangereuse et ne devra être utilisée qu'en dernier recours après avoir supprimé toute autre possibilité d'accès à la base (en filtrant, en stoppant Apache, en coupant SSH, etc...).

6.7. Sauvegarde et restauration de bases

La problématique générale de sauvegarde de bases de données est de rester en production *pendant* cette sauvegarde : c'est ce que l'on appelle la sauvegarde *à chaud*. Sauvegarder *à froid* (serveur arrêté) est aussi simple qu'une copie de fichier classique. En revanche, lorsque la base fonctionne, une sauvegarde traditionnelle ne fonctionnera pas puisque les objets sauvegardés (fichiers de tables) seront *flous* (changeants).

La méthode la plus simple et la plus sûre pour la sauvegarde à chaud consiste à faire un *dump* de la base avec **mysqldump**. Cette commande suit la syntaxe générale et prend en argument le nom de la

base que l'on souhaite « exporter ». Cet export se fera par défaut en SQL, et nous obtiendrons ainsi un export de la base sous forme de commandes permettant de reconstruire cette base.

```
root@ubuntu:~# mysqldump -u dbadmin -p xxxx
Enter password:
-- MySQL dump 10.11
--
-- Host: localhost      Database: mabase
-- -----
-- Server version      5.0.38-Ubuntu_0ubuntu1-log
...
--
-- Table structure for table `Genies`
--

DROP TABLE IF EXISTS `Genies`;
CREATE TABLE `Genies` (
  `Nom` varchar(200) default NULL,
  `Prenom` varchar(200) default NULL,
  `AnneeNaissance` int(11) default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `Genies`
--

INSERT INTO `Genies` VALUES ('Cox','Alan',1968);
INSERT INTO `Genies` VALUES ('Knuth','Donald',1938),
INSERT INTO `Genies` VALUES ('Torvalds','Linus',1969);
INSERT INTO `Genies` VALUES ('Turing','Alan',2912);
INSERT INTO `Genies` VALUES ('Gates','Bill',1955);
INSERT INTO `Genies` VALUES ('Einstein','Albert',1879);

-- Dump completed on 2007-06-17 12:52:13
root@ubuntu:~#
```

En redirigeant la sortie standard (ou avec l'option `-r`), on peut sauver ces instructions de création/remplissage de tables dans un fichier SQL qui pourra plus tard être utilisé simplement avec le shell **mysql** :

```
mysql -u dbadmin -p < fichier_dump.sql
```

Pour faire un *dump* de toutes les bases du serveur, on utilisera l'option `--all-databases` à la place du nom de la base.

L'autre possibilité pour la sauvegarde de bases à *chaud* et la commande **mysqldhotcopy**. Elle permet d'effectuer une copie des *fichiers* de tables tout en s'assurant de l'intégrité de ces fichiers. **mysqldhotcopy** ne supporte pas la lecture interactive du mot de passe (option `-p` des autres commandes). Pour l'utiliser, on devra donc configurer un identifiant/mot de passe temporairement dans la section `[client]` du fichier `my.cnf` :

```
root@ubuntu:~# cat /etc/mysql/my.cnf
#
# The MySQL database server configuration file.
#
...
# This will be passed to all mysql clients
```

```
# It has been reported that passwords should be enclosed with
  ticks/quotes
# especially if they contain "#" chars...
# Remember to edit /etc/mysql/debian.cnf when changing the socket
  location.
[client]
...
user      = dbadmin
password = motdepasse
```

L'utilisation de **mysqlhotcopy** est alors très simple et suit la syntaxe de **cp** :

```
mysqlhotcopy {base} [répertoire]
```

Par exemple, pour copier la base *mabase*, on utilisera :

```
root@ubuntu:~# mysqlhotcopy mabase ~/dbbackup/
Locked 1 tables in 0 seconds.
Flushed tables (`mabase`.`Genies`) in 0 seconds.
Copying 4 files...
Copying indices for 0 files...
Unlocked tables.
mysqlhotcopy copied 1 tables (4 files) in 0 seconds (1 seconds
  overall).
root@ubuntu:~# ls ~/dbbackup/
mabase
root@ubuntu:~# ls ~/dbbackup/mabase/
db.opt  Genies.frm  Genies.MYD  Genies.MYI
root@ubuntu:~#
```

Si un seul argument (*base*) est fourni à la commande, il fera une copie de la base avec le suffixe *_copy* qui sera immédiatement disponible comme une nouvelle base :

```
root@ubuntu:~# mysqlhotcopy mabase
Using copy suffix '_copy'
Locked 1 tables in 0 seconds.
Flushed tables (`mabase`.`Genies`) in 0 seconds.
Copying 4 files...
Copying indices for 0 files...
Unlocked tables.
mysqlhotcopy copied 1 tables (4 files) in 1 second (1 seconds
  overall).
root@ubuntu:~# mysql -u dbadmin
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> SHOW DATABASES;
+-----+
| Database                |
+-----+
| information_schema      |
| mabase                  |
| mabase_copy             |
| mysql                   |
+-----+
4 rows in set (0.00 sec)
```

mysql>

D'autres fonctionnalités sont disponibles, comme la copie de certaines tables uniquement, l'utilisation d'expressions régulières, la copie via **scp**, etc... Attention tout de même, la page de man affiche un avertissement clair : « WARNING: THIS PROGRAM IS STILL IN BETA »...

Chapitre 7. Déploiement et guide des opérations ProFTPD

\$Revision: 1.11 \$

\$Date: 2007/07/06 20:49:09 \$

Après une décennie de disette, pendant laquelle le choix d'un serveur FTP se résumait à `wu-ftp`, plusieurs serveurs FTP alternatifs ont surgi simultanément dans le monde OpenSource. Sur le devant de la scène, on trouve aujourd'hui `vsftpd`, `Pure-FTPd` et `ProFTPD`. Nous nous pencherons sur ce dernier, car ses fonctionnalités sont tout à fait honorables pour notre usage. Autre avantage, sa syntaxe est assez proche de celle d'Apache, donnant ainsi le sentiment d'être en terrain connu.

7.1. Installation

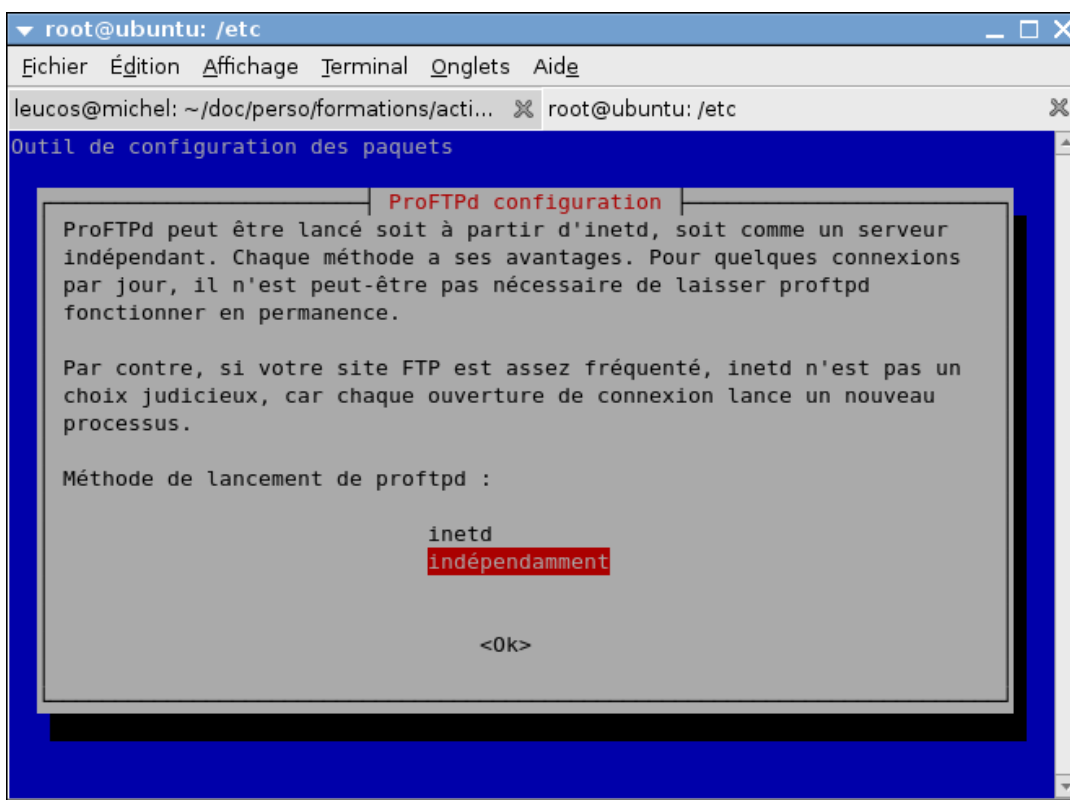
ProFTPD, comme certains services TCP, peut être démarré à la demande en cas de connexion ftp entrante. Cela permet de ne pas avoir un processus qui fonctionne en permanence, consommant de la mémoire, alors que les connexions TCP sont très rares. Dans cette configuration, c'est un *super-serveur* qui écoute le port FTP. Lorsqu'une connexion FTP est ouverte, le super-serveur exécutera ProFTPD et lui transmettra cette connexion.

Le prix à payer, théoriquement, est une certaine latence, puisque le serveur est démarré lors de la connexion. Mais dans la pratique, avec des charges raisonnables, on ne verra aucune différence entre une connexion passant par un super-serveur et une connexion directement prise en charge par ProFTPD.

Le super-serveur utilisé ici est **xinetd**. Il est plus souple que le super-serveur historique (**inetd** du package netkit) et permet une configuration très fine des ressources utilisable par les différents services.

On devra de préférence installer **xinetd** avant ProFTPD. Lors de l'installation de ce dernier, nous pourrons choisir le type d'installation dans une boîte de dialogue.

Figure 7.1. Choix d'installation



L'installation des deux paquetage s'effectue comme d'habitude avec **apt-get** :

```
root@ubuntu:/etc# apt-get install xinetd && apt-get install proftpd
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les NOUVEAUX paquets suivants seront installés :
  xinetd
0 mis à jour, 1 nouvellement installés, 0 à enlever et 8...
Il est nécessaire de prendre 0o/135ko dans les archives.
Après dépaquetage, 369ko d'espace disque supplémentaires...
Sélection du paquet xinetd précédemment désé...
(Lecture de la base de données... 18007 fichiers et répertoires...
Dépaquetage de xinetd (à partir de ../xinetd_2.3.14-lubuntu1_i...
Paramétrage de xinetd (2.3.14-lubuntu1) ...
Stopping internet superserver: xinetd.
Ajout de « diversion of /etc/init.d/inetd to /etc/init.d/inetd.real
  by xinetd »
Starting internet superserver: xinetd.

Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Paquets suggérés :
  proftpd-doc
Les NOUVEAUX paquets suivants seront installés :
  proftpd
0 mis à jour, 1 nouvellement installés, 0 à enlever et 8...
Il est nécessaire de prendre 0o/784ko dans les archives.
Après dépaquetage, 2331ko d'espace disque supplémentaire...
```

```
Préconfiguration des paquets...
Sélection du paquet proftpd précédemment dés...
(Lecture de la base de données... 18036 fichiers et répertoires...
Dépaquetage de proftpd (à partir de ../proftpd_1.3.0-2lubuntu1...
Paramétrage de proftpd (1.3.0-2lubuntu1) ...
----- IMPORTANT INFORMATION FOR XINETD USERS -----
The following line will be added to your /etc/inetd.conf file:
```

```
ftp      stream tcp      nowait  root    /usr/sbin/tcpd
/usr/sbin/proftpd
```

```
If you are indeed using xinetd, you will have to convert the
above into /etc/xinetd.conf format, and add it manually. See
/usr/share/doc/xinetd/README.Debian for more information.
-----
```

```
Adding system user `proftpd' (UID 109) ...
Adding new user `proftpd' (UID 109) with group `nogroup' ...
Not creating home directory `/var/run/proftpd'.
ProFTPD warning: cannot start neither in standalone nor in
inetd/xinetd mode....
```

```
root@ubuntu:/etc#
```

Même en installant xinetd avant ProFTPD, les scripts d'installation ne créent aucune configuration de base pour nous. Il faudra créer un fichier de configuration xinetd pour ProFTPD à la main. Sans rentrer des les détails des la syntaxe xinetd (voir xinetd.conf(5)), on pourra utiliser la configuration minimale ci-dessous. Il faudra placer ce fichier de configuration dans `/etc/xinetd.d/proftpd`.

```
service ftp
{
  disable          = no
  socket_type      = stream
  wait             = no
  nice             = 10
  user             = root
  server           = /usr/sbin/proftpd
  instances        = 4
  log_on_success  += DURATION HOST USERID
}
```

Si l'on désire exécuter ProFTPD en mode « standalone », dans lequel il fonctionne en permanence et prend l'écoute du port 21, on devra couper xinetd (ou mettre la variable `disable` à `yes`) et préciser `ServerType standalone` dans son fichier de configuration.

7.2. Configuration

La configuration de ProFTPD est donnée par le fichier `/etc/proftpd/proftpd.conf`. Ce fichier inclus tout de suite un autre fichier de configuration : `/etc/proftpd/modules.conf` La configuration par défaut de ces deux fichiers devra subir quelques ajustements.

7.2.1. `/etc/proftpd/modules.conf`

Ce fichier contient la liste des modules que ProFTPD doit charger. La plupart d'entre eux sont probablement inutiles. Il faudra bien sûr adapter en fonction de sa situation, mais si l'authentification se fait avec la base des utilisateurs locaux, inutile de charger le modules `mod_sql*`, `mod_ldap` et `mod_radius`. On pourra donc les commenter.

```
#LoadModule mod_sql.c
#LoadModule mod_ldap.c
#LoadModule mod_sql_mysql.c
#LoadModule mod_sql_postgres.c
```

Idem pour les quotas. S'ils ne sont pas utilisés, il faut commenter dans le fichier de configuration :

```
#LoadModule mod_quotatab.c
#LoadModule mod_quotatab_file.c
#LoadModule mod_quotatab_ldap.c
#LoadModule mod_quotatab_sql.c
```

Les *tcpwrappers* n'étant pas utilisés dans notre configuration, on pourra supprimer `mod_wrap`. Idem pour `mod_rewrite` servant à faire de la réécriture d'URL : s'il n'est pas utilisé on le commente.

```
#LoadModule mod_wrap.c
#LoadModule mod_rewrite.c
```

7.2.2. /etc/proftpd/proftpd.conf

Ce fichier contient le reste de la configuration du serveur. On pourra supprimer l'utilisation d'IPv6. Dans la mesure où nous l'avons désactivé au niveau système, inutile que ProFTPD tente d'ouvrir le port 21 en IPv6. Il faudra aussi changer `ServerIdent`, afin d'éviter d'exposer la version du serveur. Avec la valeur par défaut, lors d'une connexion le serveur enverrait :

```
220 ProFTPD 1.3.0 Server (MonServeur) [192.168.17.139]
```

En ajoutant la directive de configuration, nous aurons :

```
220 Serveur prêt
```

La directive `RequireValidShell` sera positionnée à *off* si nous voulons permettre aux utilisateurs d'accéder au serveur en FTP sans forcément leur fournir un shell. Cela permet de les laisser mettre des fichiers en FTP sur un espace Web même si l'on a modifié leur shell par `/bin/false`.

On utilisera aussi « `DefaultRoot ~` » afin de restreindre les utilisateurs dans leur répertoire personnel.

```
UseIPv6                off
ServerName              "MonServeur"
ServerIdent             on "Serveur prêt"
ShowSymlinks           off
DefaultRoot             ~
RequireValidShell      off
IdentLookups            off
```

7.3. Filtrage

Comme pour les autres services, il faudra ouvrir le port correspondant au service ftp (21/tcp).

```
#
# #####
# TCP entrant
# Il faudra ouvrir des ports au fil de l'eau
```

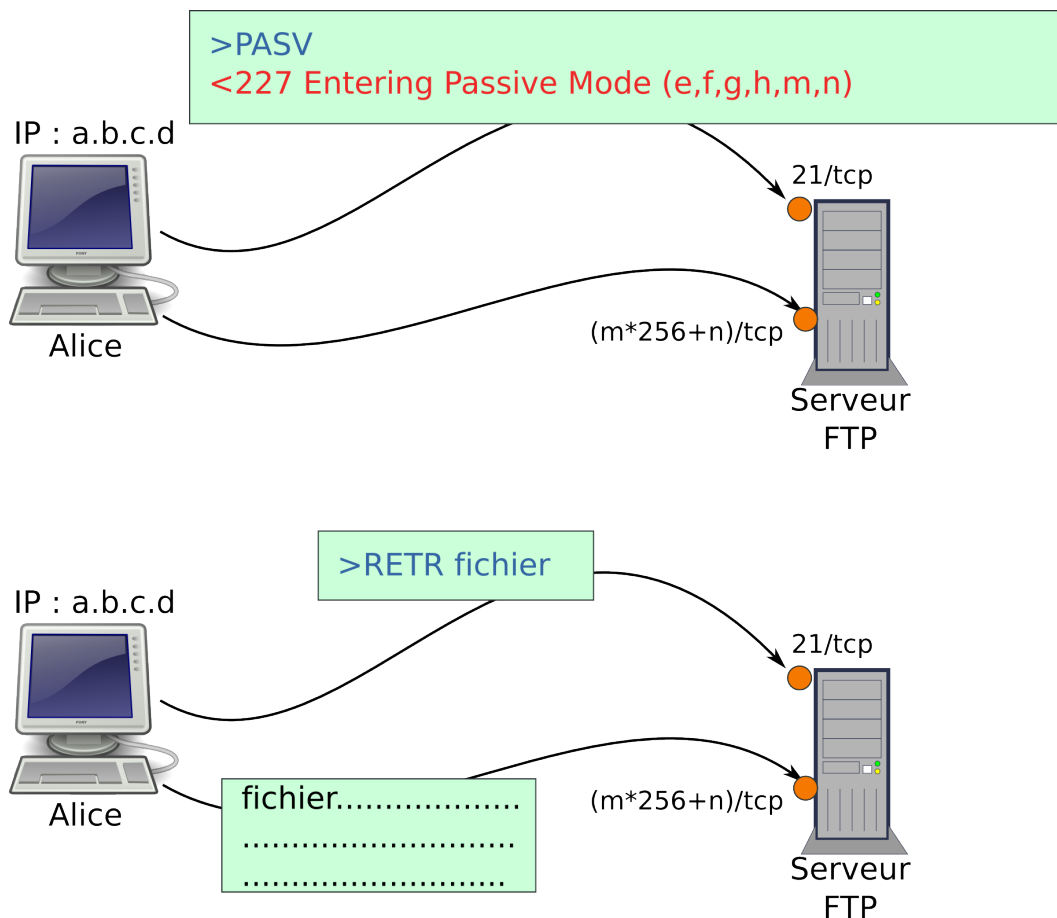
```
# lors de la mise en place de
# services TCP (ssh, apache, ...).
# #####
#
-A TCP_IN -j TCP_INLIMITS
-A TCP_IN -j STATEFUL
-A TCP_IN -j TCP_SYNLIMITS
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_IN:" --log-level 6
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
#
# Ajouter les règles ici lors de l'installation de services TCP si
# ces services
# doivent être ouverts
#
-A TCP_IN -s adresse_ip_autorisée -p tcp --dport 21 -j ACCEPT ❶
# on peut aussi débloquer le port 21 pour tout le monde
-A TCP_IN -p tcp --dport 21 -j ACCEPT ❷
#
```

- ❶ Règle autorisant l'accès au port 21/tcp (ftp) pour l'adresse *adresse_ip_autorisée* (qui peut aussi être un subnet). Cette règle peut être répétée autant de fois que nécessaire.
- ❷ Règle autorisant l'accès au port 21/tcp (ftp) pour tout le monde.

Mais concernant le protocole FTP, il y a une subtilité que l'on ne trouve pas ailleurs. Ce protocole nécessite deux connexions pour fonctionner. Une connexion de *contrôle*, qui est initiée par le client vers le port 21 du serveur. C'est par elle que transite l'authentification et les ordres envoyés par le client. Mais FTP a aussi besoin d'une connexion *data*. C'est par cette connexion que transiteront les données. La situation pour l'établissement de ce canal de données est moins simple, puisque protocole FTP a deux mode de fonctionnement, et que selon ce mode, c'est le client ou le serveur qui va se connecter à l'autre partie pour établir le canal de données.

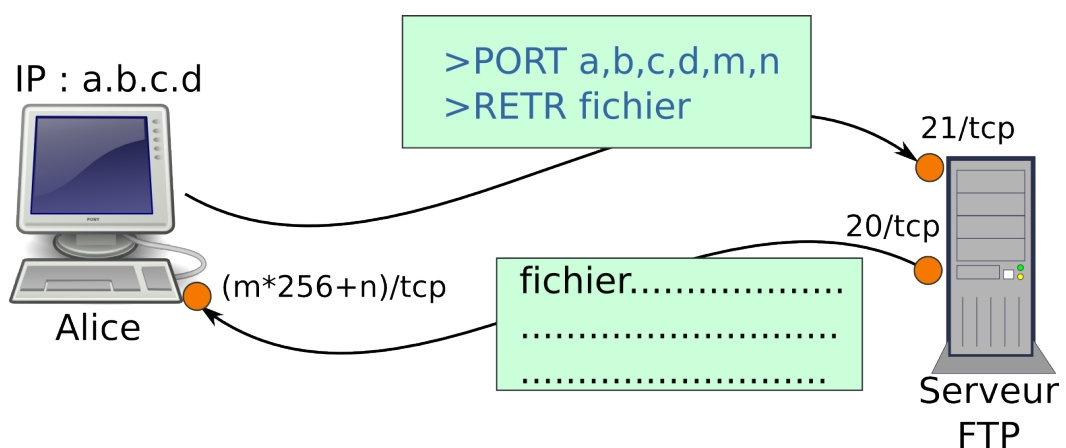
- *mode passif* : dans ce mode, le client établira une deuxième connexion vers le serveur pour le transfert de données (quel que soit l'émetteur de ces données). Le port d'écoute sur le serveur est négocié *dans* le protocole et sera > 1023. Coté client, le port source sera bien sur aussi > 1023.

Figure 7.2. Mode FTP passif



- *mode actif* : dans ce mode, c'est le serveur qui va initier la connexion ftp-data vers le client à partir du port 20.

Figure 7.3. Mode FTP actif



Ces deux modes sont parfaitement supportés par netfilter/iptables. Un module spécialisé (`nf_conntrack_ftp`) scrute les connexion FTP et permet aux canaux de contrôle de s'établir sans problème, aussi bien en mode passif qu'en mode actif. Mais nous reviendrons là dessus plus tard dans le chapitre sur le chiffrement (Section 10.3.2, « FTP/TLS »), car cette particularité du protocole FTP nous posera alors un réel problème.

7.4. Gérer le service

7.4.1. Démarrage et arrêt

Dans la configuration utilisée ici, le service ne démarre pas directement par lui même mais par l'intermédiaire du super-serveur xinetd. L'activation ou la suspension du service est donc gérée dans la configuration de xinetd (voir Section 7.4.3, « Suspendre le service »).

L'arrêt/redémarrage/reconfiguration de xinetd utilise un script SysV standard que l'on peut appeler par **invoke-rc.d** ou directement avec `/etc/init.d/xinetd`.

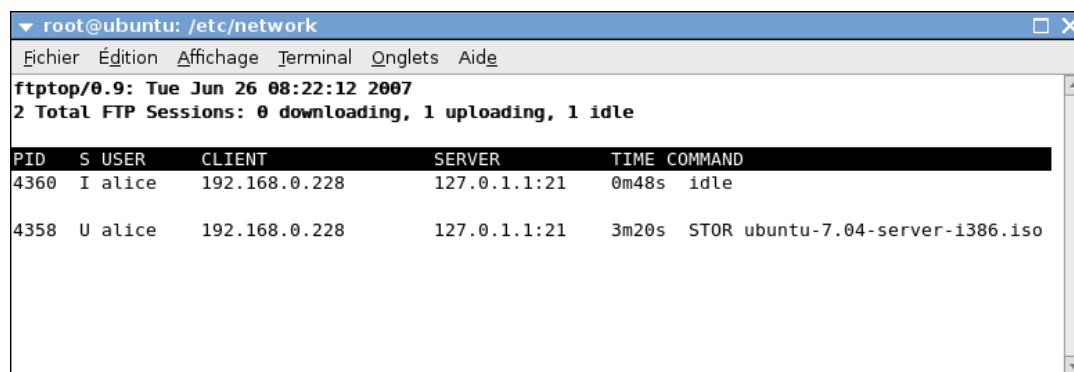
7.4.2. Supervision des connexions

ProFTPD permet de surveiller l'activité du serveur en temps réel grâce à **ftpwho** et **ftptop**. La première permet de lister les clients connectés, leur durée de connexion et la commande en cours d'exécution :

```
root@ubuntu:~# ftpwho
inetd FTP daemon:
no users connected
root@ubuntu:~# ftpwho
inetd FTP daemon:
 4329 alice      [ 0m17s]  0m11s idle
Service class           -    1 user
root@ubuntu:~#
```

ftptop donne les même informations mais rafraichies en continue, à l'image de top(1).

Figure 7.4. ftptop



PID	S	USER	CLIENT	SERVER	TIME	COMMAND
4360	I	alice	192.168.0.228	127.0.1.1:21	0m48s	idle
4358	U	alice	192.168.0.228	127.0.1.1:21	3m20s	STOR ubuntu-7.04-server-i386.iso

7.4.3. Suspendre le service

La commande **ftpsht** permet de couper immédiatement le service ftp. Cette commande crée un fichier spécial `/etc/shutmsg` qui indique à ProFTPD de ne plus accepter de connexions. Il faudra donner à **ftpsht** un paramètre temporel pour indiquer *quand* couper le service. Ces paramètre peut être *now* (coupure immédiate), *+N* (coupure dans N minutes) ou *HHMM* (coupure à HH heures et MM minutes). On pourra aussi préciser en deuxième paramètre un message qui sera renvoyé aux clients. Lorsqu'un client tentera de se connecter au serveur, il sera informé de l'arrêt du serveur :

Coupure du service sur le serveur

```
root@ubuntu:~# ftpshut now "Coupure pour maintenance"
```

Tentative de connexion

```
alice@linus:~$ ftp-ssl 192.168.17.139
```

```
Connected to 192.168.17.139.  
500 FTP server shut down (Coupure pour maintenance) -- please try  
again later  
ftp> quit  
alice@linus:~$
```

On pourra par la suite rétablir le service avec l'argument `-R` :

```
root@ubuntu:~# ftpshut -R  
ftpshut: /etc/shutmsg removed  
root@ubuntu:~#
```

7.4.4. Logs

ProFTPD fournit des logs très complets, regroupés sous `/var/log/proftpd/`.

- `/var/log/proftpd/proftpd.log` : informations générales sur le fonctionnement du serveur, connexions réussies ou échouées, etc... C'est le fichier qu'il faut surveiller pour repérer les attaques par force brute.
- `/var/log/proftpd/xferlog` : ce log contient la liste de tous les fichiers transférés vers ou depuis le serveur, le type des fichiers, etc.. On se réfèrera à `xferlog(5)` pour une description exhaustive du fichier de log.
- `/var/log/proftpd/tls.log` : détaille les opérations liées au module TLS permettant de créer un canal de communication chiffré (voir Section 10.3.2, « FTP/TLS » à ce sujet). Ces informations peuvent être intéressantes pour diagnostiquer des problèmes ponctuels, ou pour vérifier au déploiement que TLS fonctionne correctement. On pourra par contre supprimer la directive `TLSLog` pour éviter de remplir ce fichier lorsque le serveur fonctionne correctement en production.

Chapitre 8. Déploiement et guide des opérations Postfix

\$Revision: 1.12 \$

\$Date: 2007/07/10 22:01:08 \$

La plus vieille application de l'internet est probablement la messagerie ([Tomlinson]). C'est peut être ce qui explique toutes les tares du protocole SMTP¹ qui font qu'au moment où sont écrites ces lignes, 83% des emails transitant dans le monde sont du spam².

Dans notre déploiement de Postfix, nous n'aurons pas à gérer ces problèmes puisque nous mettrons en place une passerelle de messagerie d'expédition ne traitant que des messages sortants. La gestion de mails entrants à des implications en termes de configuration (DNS et adressage notamment) qui ne peuvent entrer dans le cadre de ce document.

Ce document présentera deux possibilités de plateforme de messagerie :

- cliente : la plateforme utilise un relais extérieur
- indépendante : la plateforme envoie les messages directement aux destinataires

8.1. Installation

Contrairement à la plupart des installations précédentes, celle de Postfix est assez bavarde. Le processus d'installation nous affiche une suite d'écrans d'informations et de choix. Le premier d'entre eux nous informe des différentes possibilités d'installation.

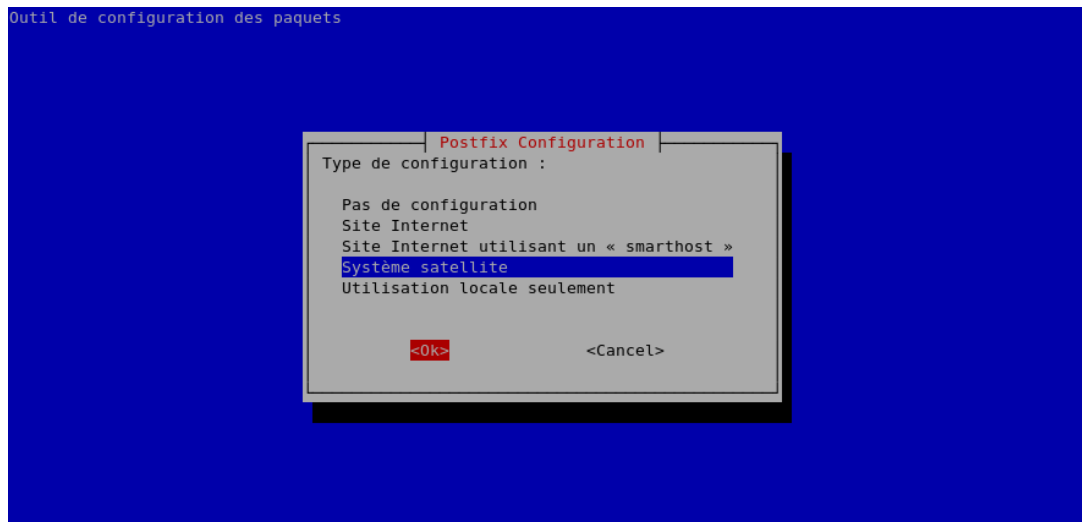
```
root@ubuntu:~# apt-get install postfix
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les paquets supplémentaires suivants seront installés :
  ssl-cert
Paquets suggérés :
  procmail postfix-mysql postfix-pgsql postfix-ldap postfix-pcre sasl2-bin
  resolvconf postfix-cdb
Paquets recommandés :
  mail-reader
Les NOUVEAUX paquets suivants seront installés :
  postfix ssl-cert
0 mis à jour, 2 nouvellement installés, 0 à enlever et 2 non mis à jour.
Il est nécessaire de prendre 0o/1101ko dans les archives.
Après dépaquetage, 2642ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ?
Préconfiguration des paquets...
Sélection du paquet ssl-cert précédemment désélectionné.
(Lecture de la base de données... 18694 fichiers et répertoires déjà installés.)
Dépaquetage de ssl-cert (à partir de ../ssl-cert_1.0.13_all.deb) ...
Sélection du paquet postfix précédemment désélectionné.
Dépaquetage de postfix (à partir de ../postfix_2.3.8-2_i386.deb) ...
Paramétrage de ssl-cert (1.0.13) ...

Paramétrage de postfix (2.3.8-2) ...
```

¹Le premier « e-mail » envoyé n'utilisait pas SMTP qui n'a été spécifié qu'en 1982 dans [RFC821]

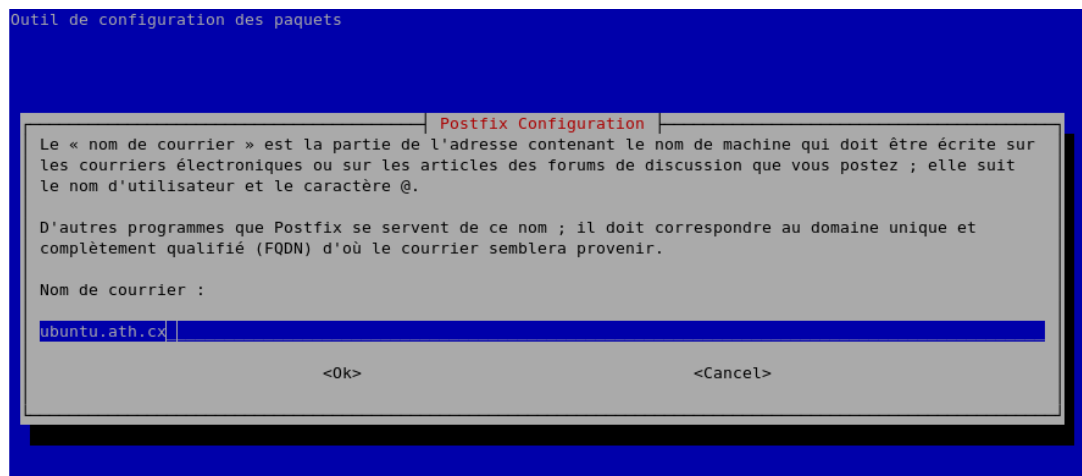
²Source : Postini [<http://www.postini.com/stats/index.php>]

Figure 8.1. Postfix: Choix du type d'installation



Nous choisirons pour l'instant « Système satellite ». L'installateur demandera ensuite le *nom* de notre serveur. En dépit des promesses faites par cette boîte de dialogue, nous ne mettrons pas le suffixe que nous voulons voir sur nos messages, mais le nom complet de notre machine.

Figure 8.2. Postfix: Choix du nom du serveur



Dans l'écran suivant, nous indiquerons le nom de notre relais de messagerie. Si par exemple notre fournisseur d'accès est *Free*, nous utiliserons `smtp.free.fr`.

Figure 8.3. Postfix: Choix du relais SMTP



L'installation se poursuit ensuite et le service est démarré :

```
Adding group `postfix' (GID 112) ...
Done.
Adding system user `postfix' (UID 111) ...
Adding new user `postfix' (UID 111) with group `postfix' ...
Not creating home directory `/var/spool/postfix'.
Creating /etc/postfix/dynamicmaps.cf
Adding tcp map entry to /etc/postfix/dynamicmaps.cf
Adding group `postdrop' (GID 113) ...
Done.
setting myhostname: ubuntu.ath.cx
setting alias maps
setting alias database
setting myorigin
setting destinations: ubuntu.ath.cx, localhost.ath.cx, localhost
setting relayhost: smtp.free.fr
setting mynetworks: 127.0.0.0/8
setting mailbox_size_limit: 0
setting recipient_delimiter: +
setting inet_interfaces: loopback-only

Postfix is now set up with a default configuration. If you need to make
changes, edit
/etc/postfix/main.cf (and others) as needed. To view Postfix configuration
values, see postconf(1).

After modifying main.cf, be sure to run '/etc/init.d/postfix reload'.

Running newaliases
* Stopping Postfix Mail Transport Agent postfix
  [ OK ]
* Starting Postfix Mail Transport Agent postfix
  [ OK ]

root@ubuntu:~#
```

Pour faciliter les différents tests de messagerie, il faudra aussi installer le paquetage mailx qui permet de disposer de la commande **mail**. Cette commande permet d'envoyer (et de lire, mais nous n'utiliserons pas cette possibilité) des messages en ligne de commande.

```
root@ubuntu:~# apt-get install mailx
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les paquets supplémentaires suivants seront installés :
  liblockfile1
Les NOUVEAUX paquets suivants seront installés :
  liblockfile1 mailx
```

```
0 mis à jour, 2 nouvellement installés, 0 à enlever et 2 non mis à jour.
Il est nécessaire de prendre 0o/171ko dans les archives.
Après dépaquetage, 385ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ? O
Sélection du paquet liblockfile1 précédemment désélectionné.
(Lecture de la base de données... 18863 fichiers et répertoires déjà installés.)
Dépaquetage de liblockfile1 (à partir de ../liblockfile1_1.06.1ubuntu1_i386.deb)
...
Sélection du paquet mailx précédemment désélectionné.
Dépaquetage de mailx (à partir de ../mailx_8.1.2-0.20050715cvs-1ubuntu2_i386.deb)
...
Paramétrage de liblockfile1 (1.06.1ubuntu1) ...
Paramétrage de mailx (8.1.2-0.20050715cvs-1ubuntu2) ...

root@ubuntu:~#
```

8.2. Configuration

Le défaut de Postfix est probablement le nombre incalculable de directives de configuration qu'il accepte. Et même si le logiciel fait des efforts particuliers pour être tolérant dans la syntaxe de son fichier de configuration, il n'en reste pas moins que le foisonnement des mots clefs utilisables et parfois leur ambiguïté rendent les configurations souvent complexes et délicates.



Note

Postfix, compilé avec le support TLS, comporte 525 directives de configuration ! (voir `root@ubuntu:~# postconf | wc -l`). On pourra prendre connaissance de leur signification dans la pages de man de `postconf(5)`.

Le package Ubuntu embarque des scripts qui facilitent la configuration initiale. On en a eu un aperçu lors de l'installation, mais ils permettent aussi d'aller un peu plus loin.

8.2.1. Reconfiguration de base

`dpkg` peut être invoqué pour reconfigurer Postfix avec quelques paramètres de base. Il faudra appeler `dpkg-reconfigure` avec l'option `--priority=low`.

L'appel de cette commande fera apparaître des écrans de configuration, identiques à ceux affichés lors de l'installation. Mais maintenant, grâce à l'option `--priority=low`, quelques paramètres supplémentaires pourront être renseignés :

1. *Type de configuration* : même chose que dans la configuration initiale. On conservera le choix *Système satellite*
2. *Destinataire du courrier destiné au superutilisateur* : on donnera une adresse email *complète* qui recevra tous les emails destinés à *root*
3. *Nom de courrier* : même chose que pour l'installation. On veillera à bien rentrer le nom d'hôte complet (*FQDN*) de la machine.
4. *Serveur relais SMTP* : paramètre déjà vu à l'installation; c'est le serveur auquel nous allons envoyer tous les messages reçus, afin qu'ils soient acheminés à leur destination finale.
5. *Autres destinations pour lesquelles le courrier sera accepté* : liste des domaines (en dehors de notre propre *FQDN* pour lesquels nous accepterons des messages. On peut laisser ce champ vide.
6. *Faut-il forcer des mises à jour synchronisées de la file d'attente des courriels* : on peut répondre *no* si le répertoire `data` de Postfix est en `ext3`. Ce n'est pas un paramètre propre à Postfix, mais plutôt lié au système de fichiers. Lorsque l'on choisit *yes*, certains répertoires verront leur attribut *synchrone* modifié (avec `chattr +S`), indiquant à l'OS que toutes les opérations d'entrée sortie impliquant ce

répertoire (ou un objet contenu dans ce répertoire) ne doivent pas être bufferisées mais écrits sur le disque de manière synchrone.

7. *Réseaux internes* : liste des réseaux pour lesquels on accepte de relayer le courrier. On saisira le réseau de *loopback* (127.0.0.0/8), notre subnet (192.168.17.0/24) et on y ajoutera celui prévu pour le déploiement du serveur OpenVPN (192.168.18.0/24).
8. *Taille maximale des boîtes aux lettres* : on peut laisser la valeur par défaut puisque nous n'utilisons pas de « livraison locale » de message.
9. *Caractère d'extension des adresses locales* : même chose ici. On peut laisser ce champ vide puisque nous n'avons pas de « livraison locale ».
10. *Protocoles internet à utiliser* : choisir le protocole IP à utiliser. Nous n'utilisons qu'IPv4.

Lorsque tous les écrans de configuration sont passés, `dpkg-reconfigure` termine la configuration et redémarre le service.

```
root@ubuntu:~# dpkg-reconfigure --priority=low postfix
* Stopping Postfix Mail Transport Agent postfix          [ OK ]
setting synchronous mail queue updates: false
setting myorigin
setting destinations: ubuntu.ath.cx, localhost.ath.cx, localhost
setting relayhost: smtp.free.fr
setting mynetworks: 127.0.0.0/8, 192.168.17.0/24, 192.168.18.0/24
setting mailbox_size_limit:
setting recipient_delimiter:
setting inet_interfaces: loopback-only
setting inet_protocols: ipv4
/etc/aliases does not exist, creating it.
adding root: alice@gmail.com alias
adding postmaster: alice@gmail.com alias

Postfix is now set up with the changes above.  If you need to make changes, edit
/etc/postfix/main.cf (and others) as needed.  To view Postfix configuration
values, see postconf(1).

After modifying main.cf, be sure to run '/etc/init.d/postfix reload'.

Running newaliases
postalias: warning: /etc/aliases.db: duplicate entry: "postmaster"
* Stopping Postfix Mail Transport Agent postfix          [ OK ]
* Starting Postfix Mail Transport Agent postfix          [ OK ]
root@ubuntu:/etc/postfix#
```

8.2.2. Fichiers

Les fichiers de configuration de Postfix sont situés sous `/etc/postfix`. Deux fichiers sont nécessaires à la configuration : `/etc/postfix/main.cf` et `/etc/postfix/master.cf`.

Le premier contient la configuration de Postfix, tandis que le second définit le comportement des divers daemons qui constituent la suite Postfix. En effet, Postfix n'est pas un simple serveur, mais une suite de daemons et d'utilitaires qui se partagent la gestion des messages. C'est la volonté d'avoir une suite logicielle faite de composants bien cloisonnés³ qui a conduit Wietse Venema à développer Postfix.

`/etc/postfix/master.cf` n'est modifié que dans le cas où l'on utilise la livraison de message locaux ou quand on veut mettre en place un nouveau composant s'insérant dans la chaîne de messagerie (antivirus ou antispam par exemple), ce qui n'est pas notre cas ici. On s'intéressera donc surtout à `/etc/postfix/main.cf`.

D'autres fichiers de configuration interviennent dans le fonctionnement de Postfix sans être directement des fichiers de configuration de Postfix. Le fichier `/etc/mailname`, par exemple, contient le nom de messagerie visible de notre système. Si l'on désire, par exemple, que tous les

³Par opposition à l'historique Sendmail d'Eric Allman

utilisateurs Unix de la machine puissent envoyer des emails en « @exemple.com », on mettra cette valeur dans le fichier.

```
root@ubuntu:~# cat /etc/mailname
exemple.com
root@ubuntu:~#
```

Le fichier `/etc/aliases`, lui aussi un peu à part, sera vu un peu plus loin (Section 8.2.4, « Alias »).

8.2.3. Principales directives

Les directives de `main.cf` peuvent être changées en éditant directement le fichier, mais on pourra préférer la méthode utilisant la commande **postconf** qui permet à la fois de lire et d'écrire les directives de `mail.cf` :

postconf en lecture :

```
postconf [clef]
```

postconf en écriture :

```
postconf -e clef = valeur
```

Appelé sans option, **postconf** affiche la totalité des valeurs de configuration (celles données dans `/etc/postfix/main.cf` et celles par défaut), tandis qu'en utilisant l'option `-n`, on ne verra que les valeurs modifiées explicitement dans le fichier de configuration.

```
root@ubuntu:~# postconf
2bounce_notice_recipient = postmaster
access_map_reject_code = 554
address_verify_default_transport = $default_transport
...
519 lignes supprimées
...
virtual_transport = virtual
virtual_uid_maps =
root@ubuntu:~# postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
append_dot_mydomain = no
biff = no
config_directory = /etc/postfix
inet_interfaces = loopback-only
inet_protocols = ipv4
mailbox_size_limit =
mydestination = localhost
myhostname = exemple.com
mynetworks = 127.0.0.0/8, 192.168.17.0/24, 192.168.18.0/24
myorigin = /etc/mailname
recipient_delimiter =
relayhost = smtp.free.fr
smtp_generic_maps = hash:/etc/postfix/generic
smtp_tls_session_cache_database =
  btree:${queue_directory}/smtp_scache
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
smtpd_tls_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_session_cache_database =
  btree:${queue_directory}/smtpd_scache
```

```
smtpd_use_tls = yes
root@ubuntu:~#
```

Chaque directive se présente sous la forme `directive = valeur`. La valeur peut utiliser d'autres directives définies ailleurs en précédant leur nom par « \$ » :

```
mydestination = $myhostname
```

Quelques un de ces paramètres méritent d'être passés en revue dans le cadre de notre configuration. Pour connaître le détail de toutes les directives, on se référera aux différents pages de man de Postfix (en particulier `postconf(5)`) ou au site officiel [<http://www.postfix.org/>].

8.2.3.1. Pour qui sommes nous relais : `mydestination` et `mynetworks`

Lorsque l'on déploie un serveur de messagerie, on doit lui donner au minimum deux informations :

- pour quels domaines acceptons nous des messages ?
- depuis quels subnets acceptons nous des messages qui ne vont pas vers nos domaines ?

Cet aspect est fondamental : on n'accepte un message que s'il nous est destiné ou s'il provient d'une adresse IP pour laquelle nous relayons des messages. Si ces deux paramètres ne sont pas renseignés correctement, notre serveur de messagerie peut se transformer en relais ouvert⁴. Postfix permet de configurer ces valeurs respectivement dans `mydestination` et `mynetworks`.

8.2.3.2. Protocole IP : `inet_interfaces` et `inet_protocols`

Postfix permet bien sûr de contrôler les interfaces sur lesquelles le démon `smtpd` sera à l'écoute, grâce à la directive `inet_interfaces`. On pourra utiliser la valeur `all` pour écouter sur toutes les interfaces (valeur par défaut), ou spécifier une liste d'adresses IP correspondant aux interfaces auxquelles on désire *binder* le démon.

```
root@ubuntu:~# postconf -e "inet_interfaces = all"
```

Lorsque nous avons demandé à ne faire que de l'IPv4 dans les écrans de configuration, la variable affectée était `inet_protocols`. Les valeurs permises sont `all`, `ipv6` et `ipv4`.

8.2.3.3. Noms : `myhostname` et `myorigin`

La directive `myhostname` contient le nom *FQDN* de la machine. Cette valeur est renseignée par les écrans de configuration initiaux. Par défaut, Postfix utilise le nom obtenu avec l'appel système `gethostname()`.

`myorigin` contient le nom qui doit être ajouté derrière un nom d'utilisateur lorsqu'un mail est envoyé localement. Si Alice (utilisateur : `alice`) envoie un courriel avec la commande `mail` sur le système, le contenu de `myorigin` sera ajouté après « @ » pour constituer son adresse.

Lors de l'installation, nous avons donné un *nom de courrier*. Ce nom a été utilisé pour remplir le fichier `/etc/mailname`. Et c'est ce même fichier qui est utilisé par Postfix dans la configuration pour affecter une valeur à `myorigin` :

```
root@ubuntu:~# postconf myorigin
myorigin = /etc/mailname
root@ubuntu:~# cat /etc/mailname
exemple.com
```

⁴Même si dans notre cas cette probabilité est faible puisque la configuration utilisée concerne un relais interne.

```
root@ubuntu:~#
```

8.2.3.4. Notre relais : relayhost

Le paramètre `relayhost` est particulièrement important dans notre configuration. En effet, c'est lui qui contrôle à *quel serveur* nous allons transmettre tout le courrier reçu pour qu'il soit livré à destination. Si ce paramètre n'est pas présent, Postfix essaiera lui-même de livrer les messages à destination. C'est donc un choix que l'on doit faire au déploiement, mais ce choix est très souvent dicté par au moins deux contraintes :

- *filtrage du port 25* : la plupart des fournisseurs d'accès bloquent l'accès au port 25 en sortie (en dehors de l'accès à leur propre serveur SMTP). Il ne sera donc pas possible de se connecter à un autre serveur SMTP que celui de son prestataire. *Free Télécom* fait exception à la règle car même si le port 25 est filtré par défaut, on peut le désactiver sur l'interface web d'administration de la FreeBox
- *blacklisting des pools d'IP client* : beaucoup de serveurs de messagerie bloquent le trafic SMTP provenant d'adresses IP considérées comme des adresses de pools client dynamiques. On ne pourra donc pas envoyer de courriel aux domaines utilisant ce système de liste noire si notre adresse IP est dans cette liste.



Attention

Si l'on désire tout de même se passer de relais, on devra surveiller l'activité du serveur de messagerie et en particulier les messages rejetés. On pourra ensuite déployer un mécanisme de transport alternatif pour les sites concernés. Mais on rentre alors dans un schéma d'exploitation très actif du serveur de messagerie.

8.2.4. Alias

Les *alias* permettent de réécrire l'adresse d'un courriel adressé à la machine locale. Même si nous n'acceptons pas de mail de l'extérieur à destination de la machine, beaucoup d'applications unix envoient des courriels à des utilisateurs locaux. `crond`, par exemple, essaie d'envoyer le résultat des commandes exécutées par email.

En utilisant les *alias*, nous pourrons envoyer les courriels de `root` vers une adresse externe (`alice@gmail.com` par exemple), nous évitant ainsi de lire les mails en local sur la machine.

Le fichier contenant les alias est en général `/etc/aliases` et son format est très simple :

```
utilisateur: nouveau_destinataire
```

Par exemple, si l'on veut envoyer les mails de `root` à `alice@gmail.com`, on ajoutera une ligne :

```
root: alice@gmail.com
```

On pourra utiliser des redirections multiples, elles seront correctement résolues :

```
postmaster: root  
root: alice@gmail.com
```

Cette configuration enverra les mails de `root` et de `postmaster` à `alice@gmail.com`.



Attention

La RFC 2142 (voir [RFC2142]) rend obligatoire la mise en place d'un certain nombre d'adresses (`postmaster`, `abuse`, ...), dépendant des services mis en place. Pas de problème dans le cas d'un système isolé qui ne prend pas d'email entrant, mais il faudra considérer ce point lors du déploiement d'un serveur public.

Le chemin fichier d'alias est défini par la directive de configuration `alias_maps`. Après avoir modifié ce fichier, il faudra le mettre à jour le fichier de base de données utilisé par postfix (`/etc/aliases.db`) avec la commande **newaliases**. Cette commande essaiera de mettre à jour la base avec le fichier pointé par la variable `alias_database`. On pourra donc utiliser la configuration suivante :

```
alias_database = $alias_maps
alias_maps = hash:/etc/aliases
```

8.2.5. Réécriture d'adresse

Notre système servira essentiellement de relais de messagerie sortant pour des utilisateurs du réseau local ou de la machine locale. Pour les clients locaux qui veulent réécrire complètement leur adresse (et pas seulement ajouter un nom de domaine à la fin comme avec `myorigin`, Section 8.2.3.3, « Noms : `myhostname` et `myorigin` »), on devra utiliser une table générique de réécriture. Cette table de réécriture change uniquement les adresses de destination. Les adresses locales sont réécrites par la table `/etc/aliases`

Cette table peut se trouver n'importe où, mais il est souhaitable de la mettre dans `/etc/postfix`. Elle contiendra des entrées sous la forme :

```
adresse_initiale    adresse_finale
```

Les adresses initiales utilisées s'entendent :

- après l'utilisation des alias (pour les destinataires locaux uniquement)
- après l'ajout de `myorigin`

Par exemple, avec la configuration ci-dessous :

```
root@ubuntu:~# cat /etc/aliases
# See man 5 aliases for format
root:    eve
postmaster:    bob@exemple.com
root@ubuntu:~# cat /etc/postfix/generic
root@exemple.com    mallaury@gmail.com
oper@exemple.com    bob@gmail.com
root@ubuntu:~#
```

un email envoyé par l'utilisateur `oper` à destination de `root` terminera sa course dans la boîte de `eve@gmail.com` et semblera provenir de `bob@gmail.com`.

Transformations appliqués à l'adresse de l'expéditeur :

```
oper                --[ myorigin ]--> oper@exemple.com
oper@exemple.com    --[ generic   ]--> bob@gmail.com
```

Transformations appliqués à l'adresse du destinataire local :

```
root                --[ aliases   ]--> eve
eve                 --[ myorigin  ]--> eve@exemple.com
```

Si l'adresse cible n'a pas de domaine, postfix ajoute \$myorigin

Si le message avait été envoyé à `postmaster`, c'est `bob@exemple.com` qui l'aurait reçu.

Afin de pouvoir utiliser le fichier `/etc/postfix/generic`, nous devons ici aussi le transformer en base de données, mais cette fois avec la commande **postmap** :

```
root@ubuntu:~# postmap /etc/postfix/generic
```

Enfin, on indiquera à Postfix que l'on souhaite mettre en place la réécriture d'adresse avec les éléments de ce fichier avec la directive `smtp_generic_maps` :

```
postconf -e 'smtp_generic_maps = hash:/etc/postfix/generic'
```

8.3. Gérer le service

8.3.1. Démarrage et arrêt

Par défaut Postfix démarre au boot. Il n'y a rien dans `/etc/default` qui permette de désactiver globalement Postfix. Mais dans la mesure où un service SMTP est quasiment nécessaire sur chaque machine (au moins pour distribuer à l'extérieur les messages générés localement), désactiver Postfix n'a pas vraiment de sens. On pourrait à la limite le remplacer par un client SMTP plus léger (ssmtp par exemple) ou le conserver tout en le restreignant à son adresse de loopback.

On pourra utiliser indifféremment la commande `invoke-rc.d` ou la commande `postfix` pour démarrer (`start`) ou arrêter (`stop`) Postfix.

Pour redémarrer, on utilisera `invoke-rc.d` avec le paramètre `restart` tandis que pour relire la configuration, nous devons utiliser cette fois la commande `postfix` avec le paramètre `reload`. Tout cela n'est malheureusement pas très consistant...

8.3.2. Gestion de la queue

Lorsque l'on envoie un courriel à Postfix afin qu'il l'expédie sur un autre serveur SMTP, il le stocke d'abord dans une file (appelée *queue* ou *spool*). Le message y restera tant qu'il ne sera pas transmis correctement au prochain serveur SMTP. En théorie (et souvent en pratique), les emails ne se perdent pas : ils sont soit dans la boîte du destinataire, soit dans la queue d'un serveur intermédiaire.

Postfix permet de gérer cette queue à l'aide de trois commandes :

- `mailq` : permet d'afficher l'état de la file des messages. Les messages dont le « Queue ID » est suivi de « * » sont en cours d'envoi,
- `postfix flush` : permet de demander le traitement immédiat de la file des messages,
- `postsuper -d [id]` : permet de supprimer un message.

```
root@ubuntu:~# mailq
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-----
954B637CC0      284 Fri Jul  6 22:07:04  root@example.com
                                alice@example.com

1748437CC7      282 Fri Jul  6 22:07:08  root@example.com
                                bob@example.com

8DE2037CC9      283 Fri Jul  6 22:07:10  root@example.com
(Host or domain name not found. Name service error for
name=smtp.orange.fr type=MX: Host not found, try again)
                                nono@orange.fr

-- 1 Kbytes in 3 Requests.
root@ubuntu:~# postfix flush
root@ubuntu:~# mailq
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-----
954B637CC0*     284 Fri Jul  6 22:07:04  root@example.com
                                alice@example.com

1748437CC7*     282 Fri Jul  6 22:07:08  root@example.com
```

bob@exemple.com

```
8DE2037CC9*      283 Fri Jul  6 22:07:10 root@exemple.com
(Host or domain name not found. Name service error for
name=smtp.orange.fr type=MX: Host not found, try again)
nono@orange.fr
```

-- 1 Kbytes in 3 Requests.

```
root@ubuntu:~# mailq
```

```
8DE2037CC9*      283 Fri Jul  6 22:07:10 root@exemple.com
(Host or domain name not found. Name service error for
name=smtp.orange.fr type=MX: Host not found, try again)
nono@orange.fr
```

-- 1 Kbytes in 1 Requests.

```
root@ubuntu:~# postsuper -d 8DE2037CC9
```

```
postsuper: 8DE2037CC9: removed
```

```
postsuper: Deleted: 1 message
```

```
root@ubuntu:~# mailq
```

```
Mail queue is empty
```

```
root@ubuntu:~#
```

8.4. Filtrage

Ce serveur ayant pour vocation *relay* les messages, il devra pouvoir recevoir des messages sur le port 25, et pouvoir en envoyer à destination du même port. On devra donc explicitement ouvrir le port 25 sur la chaîne d'entrée (TCP_IN) ainsi que comme port destination sur la chaîne de sortie (TCP_OUT).

Exemple 8.1. Postfix : configuration du filtrage TCP en entrée

```
#
# #####
# TCP entrant
# Il faudra ouvrir des ports au fil de l'eau
# lors de la mise en place de
# services TCP (ssh, apache, ...).
# #####
#
-A TCP_IN -j TCP_INLIMITS
-A TCP_IN -j STATEFUL
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_IN:" --log-level 6
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
#
# Ajouter les règles ici lors de l'installation de services TCP si
# ces services
# doivent être ouverts
#
-A TCP_IN -s adresse_ip_autorisée -p tcp -m tcp --dport 25 -j
  ACCEPT ❶
# on peut aussi débloquer le port 25 pour tout le monde
-A TCP_IN -p tcp -m tcp --dport 25 -j ACCEPT ❷
#
```

- ❶ Règle autorisant l'accès au port 25/tcp (smtp) pour l'adresse *adresse_ip_autorisée* (qui peut aussi être un subnet). Cette règle peut être répétée autant de fois que nécessaire.
- ❷ Règle autorisant l'accès au port 25/tcp (smtp) pour tout le monde.

Exemple 8.2. Postfix : configuration du filtrage TCP en sortie

```
#
#
# #####
# TCP sortant
# Cette machine initie des connexions HTTP vers
# fr.archive.ubuntu.com
# et security.ubuntu.com pour les mises à jour
# #####
#
-A TCP_OUT -j STATEFUL
...
-A TCP_OUT -p tcp -d 91.189.88.31 --dport 80 -j ACCEPT
-A TCP_OUT -p tcp --dport 25 -j ACCEPT ❶
#
```

- ❶ Régle autorisant l'envoi de paquets vers le port 25/tcp.

Chapitre 9. Déploiement et guide des opérations Samba

\$Revision: 1.9 \$

\$Date: 2007/07/07 14:26:00 \$

Samba est probablement l'un des projets OpenSource les plus impressionnants, et à plus d'un titre. C'est en premier lieu l'un des projets les plus anciens : la première version date de 1992. Ensuite, Samba offre une interopérabilité entre les mondes Windows et Linux et sur la partie la plus importante : le partage de fichiers. Cela ne surprendra personne d'apprendre que dans les premières années du projet, Microsoft n'a au mieux rien fait pour rendre le projet viable. Malgré tout, les développeurs ont réussi à implémenter ces fonctionnalités uniquement grâce à *reverse engineering*. Aujourd'hui, l'implémentation du partage CIFS (ex- NetBIOS) par Samba est probablement plus souple et plus performante que l'original, sans en emprunter la moindre ligne de code, sans avoir pu en lire la moindre spécification. Ce chapitre détaille la configuration de Samba comme serveur de fichier pour un petit groupe de travail.

9.1. Installation

L'installation avec **apt-get** importe automatiquement les utilisateurs présents sur le système. Si `xinetd` est installé, `apt-get` affichera un avertissement. Les messages d'erreur (« failed for field ») sont normaux : le script d'installation (`/var/lib/dpkg/info/samba-common.postinst`) tente d'importer les restrictions liées à la gestion des mots de passe (taille minimum, âge maximum, etc...). Or par défaut aucune de ces politiques n'est utilisée sous Ubuntu.

```
root@ubuntu:~# apt-get install samba
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les paquets supplémentaires suivants seront installés :
  libcupsys2 samba-common
Paquets suggérés :
  cupsys-common
Paquets recommandés :
  smbldap-tools
Les NOUVEAUX paquets suivants seront installés :
  libcupsys2 samba samba-common
0 mis à jour, 3 nouvellement installés, 0 à enlever et 8 non mis à
jour.
Il est nécessaire de prendre 5778ko/5956ko dans les archives.
Après dépaquetage, 14,4Mo d'espace disque supplémentaires seront
utilisés.
Souhaitez-vous continuer [O/n] ?
Réception de : 1 http://security.ubuntu.com feisty-security/main
samba-common...
Réception de : 2 http://security.ubuntu.com feisty-security/main
samba 3.0.24...
5778ko réceptionnés en 37s (155ko/s)

Préconfiguration des paquets...
...
Paramétrage de libcupsys2 (1.2.8-0ubuntu8) ...

Paramétrage de samba-common (3.0.24-2ubuntu1.2) ...

Paramétrage de samba (3.0.24-2ubuntu1.2) ...
Generating /etc/default/samba...
```



```
tdbsam_open: Converting version 0 database to version 3.
account_policy_get: tdb_fetch_uint32 failed for field 1 (min
password length)...
account_policy_get: tdb_fetch_uint32 failed for field 2 (password
history), r...
account_policy_get: tdb_fetch_uint32 failed for field 3 (user must
logon to c...
account_policy_get: tdb_fetch_uint32 failed for field 4 (maximum
password age...
account_policy_get: tdb_fetch_uint32 failed for field 5 (minimum
password age...
account_policy_get: tdb_fetch_uint32 failed for field 6 (lockout
duration), r...
account_policy_get: tdb_fetch_uint32 failed for field 7 (reset
count minutes)...
account_policy_get: tdb_fetch_uint32 failed for field 8 (bad
lockout attempt)...
account_policy_get: tdb_fetch_uint32 failed for field 9 (disconnect
time), re...
account_policy_get: tdb_fetch_uint32 failed for field 10 (refuse
machine pass...
Importing accout for root...ok
Importing accout for daemon...ok
Importing accout for bin...ok
Importing accout for sys...ok
Importing accout for sync...ok
Importing accout for games...ok
Importing accout for man...ok
Importing accout for lp...ok
Importing accout for mail...ok
Importing accout for news...ok
Importing accout for uucp...ok
Importing accout for proxy...ok
Importing accout for www-data...ok
Importing accout for backup...ok
Importing accout for list...ok
Importing accout for irc...ok
Importing accout for gnats...ok
Importing accout for nobody...ok
Importing accout for dhcp...ok
Importing accout for syslog...ok
Importing accout for klog...ok
Importing accout for oper...ok
Importing accout for sshd...ok
Importing accout for alice...ok
Importing accout for bob...ok
Importing accout for eve...ok
Importing accout for mysql...ok
Importing accout for ftp...ok
Importing accout for proftpd...ok
----- IMPORTANT INFORMATION FOR XINETD USERS -----
The following line will be added to your /etc/inetd.conf file:

#<off># netbios-ssn stream tcp nowait root /usr/sbin/tcpd
/usr/sbin/smbd
```

If you are indeed using xinetd, you will have to convert the above into /etc/xinetd.conf format, and add it manually. See

```
/usr/share/doc/xinetd/README.Debian for more information.
```

```
-----  
* Starting Samba daemons...  
  [ OK ]
```

```
root@ubuntu:~#
```

9.2. Configuration

9.2.1. Les démons Samba

Samba utilise deux démons distincts, tous deux configurés dans `/etc/samba/smb.conf`.

`nmbd` (ports udp 137 & 138) gère le service de noms et les annonces (permettant de constituer le fameux « voisinage réseau »). Lorsqu'un hôte « parle » le NetBIOS, il en informera ses voisins. Par ailleurs, chaque hôte aura un nom NetBIOS, distinct de son nom d'hôte au sens TCP/IP. `nmbd` permettra de gérer les requêtes liées à la résolution de ces noms. Par extension, `nmbd` pourra aussi être utilisé comme un *serveur de noms* NetBIOS (serveur WINS).

`smbd` (port 139/tcp) fournit le service de partage de fichiers aux clients. C'est lui qui permettra l'envoi/la réception de fichiers, le partage d'imprimantes, etc...

On pourra exécuter `smbd` depuis `xinetd` : inutile de faire fonctionner en permanence le démon prenant en charge le service de partage. En revanche, il vaut mieux laisser fonctionner sans interruption `nmbd`, car les sollicitations liés au service de noms windows sont nombreuses¹. D'ailleurs Ubuntu, avec les scripts SysV livrés, ne permet pas de déléguer le démarrage de `nmbd` à `xinetd`.

9.2.2. Démon permanent ou super-serveur xinetd

Pour demander à `xinetd` de lancer `smbd` à la demande, on devra créer un fichier de configuration `/etc/xinetd.d/samba` contenant au moins les éléments suivants :

```
service netbios-ssn  
{  
    disable      = no  
    socket_type  = stream  
    protocol     = tcp  
    wait         = no  
    user         = root  
    server       = /usr/sbin/smbd  
}
```

Il faudra ensuite configurer le script SysV de Samba afin de l'empêcher de lancer `smbd` au démarrage. Pour cela on doit modifier la variable `RUN_MODE` dans le fichier `/etc/default/samba` et lui affecter la valeur `inetd` :

```
root@ubuntu:/etc/samba# more /etc/default/samba  
# Defaults for samba initscript  
# sourced by /etc/init.d/samba  
# installed at /etc/default/samba by the maintainer scripts  
#  
...  
  
# How should Samba (smbd) run? Possible values are "daemons"  
# or "inetd".
```

¹On pourra faire un `tcpdump "ether broadcast and port (137 or 138)"` pour s'en convaincre...

```
RUN_MODE="inetd"  
root@ubuntu:/etc/samba#
```

9.2.3. Choix d'architecture

Samba permet d'utiliser 5 modes de sécurité différents. Il faudra déterminer, lors de la configuration initiale, le mode que l'on désire utiliser en fonction du type de serveur que l'on souhaite mettre en œuvre.

- i. *security = user* : le client ouvre une session avec le serveur et demandera par la suite accès à des ressources. toutes les informations d'authentification sont conservées sur le serveur. C'est le mode de fonctionnement de choix pour un serveur isolé, et celui qui sera développé ici.
- ii. *security = share* : le client envoie un mot de passe pour avoir accès à une ressource particulière. Ce mode permet de protéger ces ressources par des mots de passe, sans notion d'utilisateur.
- iii. *security = domain* : le serveur fait partie d'un domaine windows, soit en tant que contrôleur primaire de domaine (PDC) qui maintient la liste des utilisateurs du domaine Windows, soit en tant que contrôleur secondaire de domaine (BDC) qui permet d'authentifier les utilisateurs, soit comme serveur membre de domaine (DMS) qui fournit des ressources à des usagers authentifiés. On utilisera ce mode lorsque l'on souhaite gérer une authentification commune sur plusieurs serveurs ou fournir un service d'ouverture de sessions ou de politique centralisée.
- iv. *security = ads* : le serveur Samba prendra part à un domaine Active Directory.
- v. *security = server* : le serveur Samba agira comme un *intermédiaire* et ira valider les login/mot_de_passe des utilisateurs sur une autre serveur. Attention à l'ambiguïté : le mode *security = server* signifie que notre machine délèguera l'authentification à un *autre* serveur et non qu'elle est serveur.

9.2.4. /etc/samba/smb.conf

Le fichier de configuration de Samba, à l'image de celui de MySQL, est divisé en sections. A l'exception de « [global] », les autres sections représentent des *shares* (partages), c'est à dire des répertoires ou des imprimantes qui seront disponibles sur le réseau.

Les paramètres suivent le schéma *clef = valeur*. lorsque *valeur* est de type booléen, on pourra utiliser sans distinction *true*, *yes* ou *1* et *false*, *no* ou *0*.

Après chaque modification de la configuration, on pourra utiliser l'outil **testparm** afin de la valider, avant de redémarrer le service avec **invoke-rc.d**.

```
root@ubuntu:~# testparm  
Load smb config files from /etc/samba/smb.conf  
Processing section "[homes]"  
Processing section "[public]"  
Global parameter guest account found in service section!  
Loaded services file OK.  
Server role: ROLE_STANDALONE  
Press enter to see a dump of your service definitions  
  
[global]  
  workgroup = FEISTYGROUP  
  server string = %h server (Samba sous Feisty)  
  obey pam restrictions = Yes  
  passwd backend = tdbsam  
  pam password change = Yes  
  passwd program = /usr/bin/passwd %u  
  passwd chat = *Enter\snew\sUNIX\spassword:* %n\  
*Retype\snew\sUNIX\s...
```

```
unix password sync = Yes
syslog = 0
log file = /var/log/samba/log.%m
max log size = 1000
load printers = No
dns proxy = No
panic action = /usr/share/samba/panic-action %d
invalid users = root
```

```
[homes]
comment = Home Directories
valid users = %S
read only = No
create mask = 0600
directory mask = 0700
browseable = No
```

```
[public]
comment = Espace public
path = /home/public
read only = No
guest only = Yes
guest ok = Yes
```

```
root@ubuntu:~#
```

Samba possède toutefois une particularité concernant la gestion de son fichier de configuration. A l'instar de **crond**, Samba vérifie toutes les minutes si son fichier de configuration a été modifié. Si c'est le cas, il le relit et applique immédiatement les modifications. Samba est intelligent et ne « s'auto-détruit » pas en appliquant un fichier de configuration invalide. Mais afin d'éviter les mauvaises surprises, on pourra effectuer les modifications de configuration en plusieurs temps :

1. copie du fichier de configuration original
2. édition de la copie
3. vérification de la copie avec **testparm**
4. correction de erreurs éventuelles et retour au point 3s
5. déplacement de la copie vers l'original

Cela peut sembler fastidieux. Mais c'est un automatisme à acquiescer au même titre que la sauvegarde d'un fichier avant modification (Section 2.1, « Modifier un fichier de configuration »).

```
root@ubuntu:~# cd /etc/samba/
root@ubuntu:/etc/samba# cp smb.conf smb.conf.new
root@ubuntu:/etc/samba# vi smb.conf.new
...édition du fichier...
root@ubuntu:/etc/samba# testparm smb.conf.new
Load smb config files from smb.conf.new
params.c:Parameter() - Ignoring badly formed line in configuration file: blabla
Processing section "[homes]"
Processing section "[administration]"
Processing section "[public]"
Global parameter guest account found in service section!
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
^C
root@ubuntu:/etc/samba# vi smb.conf.new
...correction du fichier...
```

```
root@ubuntu:/etc/samba# vi smb.conf.new
root@ubuntu:/etc/samba# testparm smb.conf.new
Load smb config files from smb.conf.new
Processing section "[homes]"
Processing section "[administration]"
Processing section "[public]"
Global parameter guest account found in service section!
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
^C
root@ubuntu:/etc/samba# mv smb.conf.new smb.conf
root@ubuntu:/etc/samba#
```

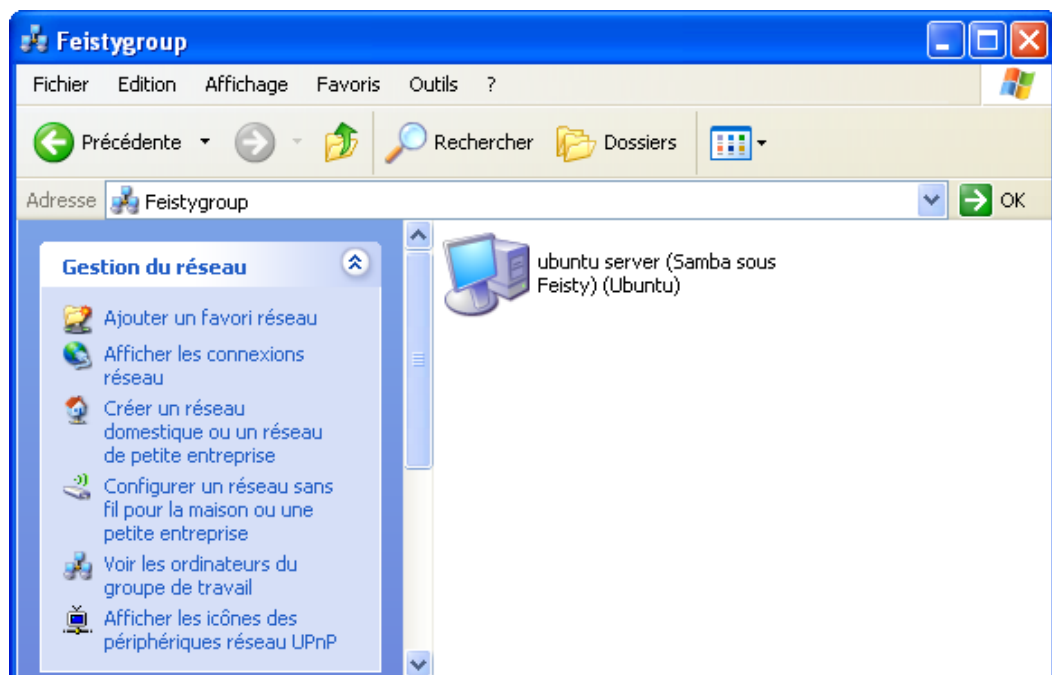
9.2.4.1. Configuration générale « [global] »

Cette section contient les paramètres généraux de configuration pour Samba (nmbd et smbд). On ne détaillera ici que les plus importants. On pourra toujours se référer à (l'excellente et volumineuse) documentation sur le site de Samba (<http://www.samba.org>). Certains paramètres globaux peuvent être surchargés dans une section spécifique à un partage. Ils seront accompagnés du symbole « † ».

Paramètres généraux

- `security = user` : c'est le mode de fonctionnement choisi par défaut. Le serveur configuré ici sera autonome pour la gestion des comptes.
- `workgroup` permet de spécifier le groupe de travail du serveur. Par défaut, Samba utilisera la valeur WORKGROUP.
- `netbios name` quand il est spécifié permet de faire apparaître la machine sous un nom NetBIOS spécifique. Par défaut, Samba annoncera « %h »: le nom d'hôte de la machine..
- `server string` : affecte une description à notre serveur qui sera vue sur le réseau. Par exemple, la valeur %h server (Samba sous Feisty) sera vue sur le réseau comme sur la figure ci-dessous. %h sera remplacé par le nom d'hôte TCP/IP (la substitution de variables est décrite dans la Section 9.2.4.6, « Substitution de variables »).

Figure 9.1. Vue du serveur dans le « Voisinage réseau »



- `log file` permet d'indiquer à Samba le chemin du fichier de log. Habituellement, Samba crée des logs individuels pour chaque machine cliente en mettant la valeur `/var/log/samba/log.%m` dans ce paramètre.
- `encrypt passwords` : demande à Samba d'utiliser des mots de passe chiffrés dans ses négociations avec les clients. Par défaut la valeur est à `true` et sauf si l'on a des clients très anciens sur le réseau (NT4, Windows 95, ...) il est recommandé de la laisser telle qu'elle.
- `passdb backend` permet de choisir la méthode de conservation des données utilisateur. On privilégiera l'utilisation de `tdbsam`, plus efficace que `smbpasswd`. L'autre possibilité, `ldap`, nécessite le déploiement d'un serveur LDAP et n'est pas couverte ici.
- `load printers` indique à Samba de charger ou non les imprimantes système afin de les partager (voir Section 9.2.4.3, « Partage d'imprimante « [printers] » » pour le partage des imprimantes).

Contrôles d'accès

D'autres paramètres permettent de cadrer l'utilisation du serveur : restriction du service à des hôtes ou utilisateurs, masquage de fichiers, gestion des droits...

- `interfaces` : liste les interfaces sur lesquelles les démons accepteront les paquets. Il faut noter que `nmbd` recevra toujours les paquets de toutes les interfaces mais fera le tri ensuite en fonction de l'IP source. Il est donc assez facile d'envoyer un paquet avec une IP source *spoofée* pour le faire accepter à `nmbd`. On pourra spécifier plusieurs paramètres séparés par des espaces. Ces paramètres pourront être des noms d'interface (`eth1`), éventuellement génériques (`eth*`). On peut aussi spécifier une adresse IP. Dans ce cas le service sera *bindé* (associé) à l'interface ayant cette adresse. Pour que ce paramètre soit pris en compte par Samba, on devra ajouter `bind interfaces only = true` dans le fichier de configuration. Cette possibilité de configuration est particulièrement utile lorsque les clients arrivent sur le serveur via OpenVPN (voir Chapitre 11, *Déploiement et guide des opérations OpenVPN*) : on peut alors demander à Samba de n'accepter que les connexions arrivant par ce tunnel :

```
# interfaces autorisées : tun* (tunnels) et eth0 (LAN) mais pas
eth1 (WAN)
interfaces = tun* eth0
bind interfaces only = yes
```

- `hosts allow` † (ou `allow hosts`) donne la liste des machines autorisées à utiliser le service. On pourra utiliser des adresses IP, des adresses partielles, des adresses avec masque et des noms d'hôte. On peut aussi exclure des machines de l'ensemble avec le mot clef `EXCEPT`.
- `hosts deny` † (ou `deny hosts`) donne la liste des machines qui n'ont pas le droit d'utiliser le service. Pour représenter tout l'espace adressable, on pourra utiliser `ALL` ou `0.0.0.0/0`.

```
Cette configuration autorise l'accès à toutes les machines de
192.168.17.0 à 192.168.18.127 sauf 192.168.18.1
hosts allow = 192.168.17. 192.168.18.0/255.255.255.128 EXCEPT
192.168.18.1
hosts deny = ALL
```

- `admin users` † permet de spécifier la liste des utilisateurs qui pourront agir en tant que *root*, c'est à dire pouvant écrire (et donc effacer) dans n'importe quel fichier via un partage Samba. On évitera d'utiliser cette option si possible, et on privilégiera une gestion plus fine des droits (voir Section 9.2.4.4, « Partages spécifiques « [...] » »). On peut spécifier un ou plusieurs utilisateurs séparés par des espaces (ou des virgules) ainsi que des groupes en les préfixant par « + ». Attention, cette option définit les droits de l'utilisateur *une fois loggué*. Il faudra donc aussi ajouter ces utilisateurs dans `valid users` le cas échéant afin qu'ils puissent d'abord se connecter à la ressource.
- `invalid users` † en revanche permet de faire la liste des utilisateurs qui n'auront aucun accès aux services Samba (ou au partage en question). On spécifie ici aussi une liste d'utilisateurs et de groupes

séparés par des espaces ou des virgules. On pourra y mettre la liste des utilisateurs « système », mais on lui préférera généralement `valid users`.

- `valid users` † permet de lister les utilisateurs pouvant accéder au service. Les utilisateurs non-listés (ou n'appartenant pas à un groupe listé) ne pourront accéder au service. Si un utilisateur est listé dans `valid users` et `invalid users`, il n'aura pas accès au service.
- `hide dot files` † permet de masquer les fichiers commençant par un « . ». Cela n'empêche pas d'y accéder, mais les supprime juste de l'affichage. Cette option est très utile dans les répertoires personnels, souvent envahis de fichiers et répertoires de préférences applicatives, traditionnellement préfixés par un point.
- `hide special files` † contrôle l'affichage des fichiers spéciaux (sockets, devices, fifos). Ils sont affichés par défaut, on pourra donc supprimer cet affichage en affectant `no` à cette valeur.
- `hide unreadable` † permet de masquer les fichiers que l'utilisateur ne peut pas lire.
- `create mask` † permet de changer les droits unix qui seront affectés à la création d'un fichier. La valeur par défaut est `744`. On pourra utiliser une valeur plus raisonnable de `0600`.
- `directory mask` † permet de changer les droits unix qui seront affectés à la création d'un répertoire. La valeur par défaut est `755`. On pourra là aussi utiliser une valeur de `0700` un peu plus restrictive.
- `hide files` † permet de masquer les fichiers correspondant au patrons passés en paramètre. Ces patrons (façon *globs* du shell) devront être séparés par des « / ».

```
# on affiche pas les fichiers "point" (.*), le bureau (Desktop),  
# la poubelle (.Trash, déjà couvert par .* !) et les backup emacs  
(*~)  
hide files = /*/Desktop/.Trash/*~
```

Ressources

Samba propose quelques options pour limiter la consommation de ressources, voire désactiver certains services.

- `max connections` limite le nombre maximum de connexions simultanées sur une ressource (partage, imprimante) Samba. Par défaut il n'y a aucune limite.
- `max smbd processes` limite le nombre maximum de process `smbd` simultanés. Par défaut, il n'y a aucune limite. Ce paramètre permet, dans une certaine mesure, de se prémunir contre des clients bogués ou des dénis de service. On notera que normalement Samba démarre un process `smbd` par client (via `xinetd` ou directement en `dæmon`). Chaque processus pourra gérer plusieurs connexion simultanées avec ce client.
- `available` † permet de couper l'accès à une ressource. L'intérêt est de pouvoir empêcher les utilisateurs d'accéder à une ressource sans avoir à commenter une grande partie du fichier de configuration.

9.2.4.2. Répertoires personnels « [homes] »

La section `[homes]` permet de configurer l'accès de tous les utilisateurs à leur répertoire personnel sans avoir à créer un partage pour chacun d'entre eux. Les paramètres par défaut de Samba sous la Ubuntu On pourra mettre la directive `browsable` à `no` afin de masquer le partage `[homes]`.

9.2.4.3. Partage d'imprimante « [printers] »

La section `[printers]` dans la configuration par défaut permet de mettre à disposition les imprimantes locales à travers le réseau. Le client Windows devra installer dans ce cas son propre driver correspondant à l'imprimante utilisée via le partage réseau.

Samba offre aussi la possibilité de mettre le driver à disposition des clients et afin de leur permettre de l'installer automatiquement. On se reportera à Printing and Name Resolution [http://www.oreilly.com/catalog/samba/chapter/book/ch07_01.html#ch07-30008] pour plus de détails sur les opérations d'impression.

9.2.4.4. Partages spécifiques « [. . .] »

Samba permet bien sûr de partager n'importe quel répertoire du filesystem serveur. On devra créer un *share* pour chacune des arborescences que l'on souhaite mettre à disposition. La création d'un tel partage est très simple, il suffit d'ajouter une section [nomdupartage] dans le fichier de configuration. On spécifiera pour chacun des partages le paramètre *path* indiquant le répertoire physique partagé.

```
[isos]
  comment = Images ISOs
  path = /var/spare/isos/
  writeable = no
```

On pourra utiliser des paramètres dans ces définitions qui permettront de restreindre l'utilisation de la ressource.

- *writeable* permet de définir si la ressource est accessible en lecture seule (*no*) ou en lecture écriture (*yes*). Dans ce dernier cas, les utilisateurs autorisés à utiliser le partage pourront modifier son contenu. On peut, selon ses goûts, utiliser *read only* au lieu de *writeable*.
- *public* permet d'indiquer à Samba que la ressource est publique, et qu'il n'est pas nécessaire d'être un utilisateur authentifié pour y avoir accès.

Ces paramètres étant définis, on pourra créer des exceptions. Par exemple, il est possible de spécifier la liste des utilisateurs ayant l'autorisation d'écrire sur une ressource *writeable no* (la valeur par défaut) en les ajoutant à *write list*.

De la même manière, on pourra spécifier la liste des utilisateurs ne pouvant accéder qu'en lecture à une ressource *writeable yes* en les ajoutant dans *read list*

Il faut donc bien comprendre que nous maîtrisons l'accès à notre ressource en spécifiant le mode de mise à disposition (*writeable*) et la liste des utilisateurs autorisés (*valid users*) ou refusés (*invalid users*). Ce n'est qu'ensuite que nous pourrons créer des exceptions dans notre schéma avec *read list* et *write list*

Imaginons par exemple que nous désirions créer un partage « documentation » accessible en lecture à tous les membres d'un club (ces membres font tous partie du groupe unix *club*) mais que les rédacteurs (groupe unix *redac*) et le secrétaire du club puissent y avoir accès en écriture. On pourra réaliser cela avec la petite section de configuration suivante :

```
[documentation]
  comment = Documentation pour le club
  path = /home/doc/
  writeable = no
  le groupe « redac » et l'utilisateur « secretaire » font partie du
  groupe « club »
  valid users = +club
  write list = +redac secretaire
```

Lorsque l'on met à disposition une ressource partagée entre plusieurs utilisateurs, il va falloir gérer la problématique des droits. En effet, un si utilisateur ayant les droits d'écriture crée un fichier sur une ressource partagée, il faut que les autres utilisateurs ayant droit d'écriture puissent le faire. Or le fichier est créé sur le serveur Linux sera assorti de droits unix standards, et, bien évidemment, Samba n'outrepasse jamais les droits Unix ².

²De même, des droits Unix laxistes ne peuvent permettre d'outrepasser des contrôles d'accès Samba

Ce problème a deux solutions. La première consiste à forcer un masque avec les directives `create mask` et `directory mask` (voir la section intitulée « Contrôles d'accès »). L'inconvénient dans ce cas est la création des fichiers avec des droits forcément laxistes : en effet, un fichier créé appartiendra à son créateur (et au groupe principal du créateur généralement homonyme). Donc pour permettre à un autre utilisateur d'y accéder, il faudra avoir au moins un masque en `666`. Toute considération satanique mise à part, ce mode pose un réel problème : n'importe qui ayant accès au filesystem de la machine pourra accéder en lecture+écriture au fichier.

On préférera donc la deuxième solution qui met en œuvre deux nouvelles directives : `force user` et `force group`. Elles permettent d'effectuer toutes les opérations liées au filesystem sous l'identité d'un utilisateur et d'un groupe particuliers.

L'idée est donc de créer un utilisateur « virtuel » pour chaque ressource partagée à plusieurs de forcer les opérations sur ce username, comme dans l'exemple précédent, repris et amélioré ci-dessous.

Pour notre exemple, on commencera par créer l'utilisateur `doc`, et on bloquera immédiatement son compte (inutile de se connecter sous cet identifiant).

```
root@ubuntu:~# adduser doc
Adding user `doc' ...
Adding new group `doc' (1005) ...
Adding new user `doc' (1003) with group `doc' ...
Creating home directory `/home/doc' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for doc
Enter the new value, or press ENTER for the default
    Full Name []: Documentaliste Virtuel
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [y/N] y
root@ubuntu:~# passwd -l doc
Password changed.
root@ubuntu:~# rm ~doc/.*
rm: cannot remove `.' or `..'
rm: cannot remove `.' or `..'
root@ubuntu:~#
```

On reprendra ensuite la section `[documentation]` afin de lui apporter les modifications nécessaires.

```
[documentation]
    comment = Documentation pour le club
    path = /home/doc/
    writeable = no
le groupe « redac » et l'utilisateur « secretaire » font partie du
groupe « club »
    valid users = +club
    write list = +redac secretaire
    force user = doc
    force group = doc
il est conseillé de faire apparaître les directives mask dans la
section « [global] »
    create mask = 0600
    directory mask = 0700
```

On pourra ensuite créer un fichier sur la ressource depuis un poste de travail et vérifier qu'il possède bien les droits prévus.

```
root@ubuntu:~# ls -la /home/doc/
total 24
drwxr-xr-x 2 doc  doc  4096 2007-06-29 18:54 .
drwxr-xr-x 8 root root  4096 2007-06-29 18:37 ..
-rw----- 1 doc  doc    6 2007-06-29 18:54 fichier_test.txt
root@ubuntu:~#
```

9.2.4.5. Partages public « [. . .] »

Samba permet d'accéder à des partages sans avoir à fournir une authentification valide grâce à `guest ok` ou son synonyme `public`. On pourra ici aussi mettre ce partage en lecture seule ou en lecture-écriture, voire spécifier des groupes autorisés à écrire. Si on permet l'écriture, on utilisera si possible un utilisateur virtuel (voir la section précédente) afin de simplifier la gestion des droits.

```
[public]
comment = Secteur public
path = /home/public
public = yes
read only = yes
write list = +redac
force user = public
```

9.2.4.6. Substitution de variables

Dans tout le fichier de configuration, on pourra utiliser des variables de substitution. Ces variables sont interprétées à la volée par Samba. Dans la configuration par défaut, cette fonctionnalité est utilisée pour générer un fichier de log pour chaque machine, ou encore pour générer `server string`.

On pourra aussi, par exemple, changer certains paramètres en fonction de ces variables. Imaginons qu'un kiosque public de consultation (nom NetBIOS : `KIOSK`) soit dans nos locaux, et que nous ne voulons pas que les membres du groupe `redac` puissent écrire sur le share `[public]` défini plus haut depuis ce poste, on pourra modifier la section précédente pour qu'elle devienne

```
[public]
comment = Secteur public
path = /home/public
public = yes
read only = yes
write list = +redac
force user = public

include = /etc/samba/smb.%m.conf
```

On demande ici à Samba de lire un fichier de configuration dépendant de la machine qui utilise le service. On pourra créer un fichier `/etc/samba/smb.KIOSK.conf` avec une nouvelle directive de configuration `write list` vide qui écrasera l'ancienne :

```
write list =
```

Tableau 9.1. Samba : principales variables de substitution

Variable	Correspondance
%U	Nom d'utilisateur demandé pour la session.
%u	Nom d'utilisateur unix réel pour la session.
%G	Groupe primaire de %U.
%g	Groupe primaire de %u.
%H	Répertoire personnel de %u.
%G	Groupe principal de %U.
%h	Nom d'hôte TCP/IP du serveur Samba.
%m	Nom NetBIOS de la machine cliente.
%M	Nom DNS de la machine cliente.
%S	Nom de la ressource courante
%P	Racine sur le disque de la ressource courante
%T	Date et heure courantes
;%\$VAR	Variable d'environnement VAR

9.3. Filtrage

NetBIOS utilise les ports 137 et 138 sur udp pour le service de noms windows et le « voisinage réseau ». On devra ouvrir ces ports afin de participer la discussion.

Exemple 9.1. Samba : configuration du filtrage UDP en entrée et en sortie

```
#
# #####
# UDP entrant
# L'appel à STATEFUL suffit pour accepter les réponses DNS
# Il faudra cependant ouvrir des ports au fil de l'eau lors de la
# mise en place
# de services UDP (DNS, NTP par exemple).
# #####
#
-A UDP_IN -j STATEFUL
#
# Ajouter les règles ici lors de l'installation de services UDP si
# ces services
# doivent être ouverts
#
-A UDP_IN -p udp -m udp -s adresse_ip_autorisée --dport 137:138 -j
ACCEPT ❶
# on peut aussi débloquer les ports 137 à 138 pour tout le monde
-A UDP_IN -p udp -m udp --dport 137:138 -j ACCEPT ❷
#
# #####
# UDP sortant
# -remplacer SERVEUR_DNS par le serveur DNS et répéter la ligne
# pour chacun des
# serveurs (primaire, secondaire, etc...)
# -remplacer SERVEUR_NTP par l'adresse IP du serveur NTP si ce
# protocole est
# utilisé
# #####
#
-A UDP_OUT -p udp -m udp -j STATEFUL
...
## -A UDP_OUT -p udp -m udp --sport 68 --dport 67 -j ACCEPT
#
-A UDP_OUT -p udp -m udp --dport 137:138 -j ACCEPT ❸
COMMIT
#
```

- ❶ Règle autorisant l'accès aux ports udp 137 (netbios-ns) à 138 (netbios-dgm) pour l'adresse *adresse_ip_autorisée* (qui peut aussi être un subnet). Cette règle peut être répétée autant de fois que nécessaire.
- ❷ Règle autorisant l'accès aux ports udp 137 (netbios-ns) à 138 (netbios-dgm) pour tout le monde.
- ❸ Règle autorisant l'envoi de paquets vers les ports udp 137 et 138.

Le port 139/tcp doit aussi être ouvert dans les deux sens afin de faire fonctionner le transfert de données (fichiers, spooling, ...).

Exemple 9.2. Samba : configuration du filtrage TCP en entrée

```
#
# #####
# TCP entrant
# Il faudra ouvrir des ports au fil de l'eau
# lors de la mise en place de
# services TCP (ssh, apache, ...).
# #####
#
-A TCP_IN -j TCP_INLIMITS
-A TCP_IN -j STATEFUL
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_IN:" --log-level 6
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
#
# Ajouter les règles ici lors de l'installation de services TCP si
# ces services
# doivent être ouverts
#
-A TCP_IN -s adresse_ip_autorisée -p tcp -m tcp --dport 139 -j
  ACCEPT ❶
# on peut aussi débloquent le port 139 pour tout le monde
-A TCP_IN -p tcp -m tcp --dport 139 -j ACCEPT ❷
#
```

- ❶ Règle autorisant l'accès au port 139/tcp (netbios-ss) pour l'adresse *adresse_ip_autorisée* (qui peut aussi être un subnet). Cette règle peut être répétée autant de fois que nécessaire.
- ❷ Règle autorisant l'accès au port 139/tcp (netbios-ss) pour tout le monde.

Exemple 9.3. Samba : configuration du filtrage TCP en sortie

```
#
#
# #####
# TCP sortant
# Cette machine initie des connexions HTTP vers
# fr.archive.ubuntu.com
# et security.ubuntu.com pour les mises à jour
# #####
#
-A TCP_OUT -j STATEFUL
...
-A TCP_OUT -p tcp -d 91.189.88.31 --dport 80 -j ACCEPT
-A TCP_OUT -p tcp -d adresse_ip_autorisée --dport 139 -j ACCEPT ❶
#
```

- ❶ Règle autorisant l'envoi de paquets vers le port 139/tcp.

9.4. Gérer le service

Gérer Samba se borne en général à la création/modification des utilisateurs. Il est même inutile de redémarrer le service puisqu'il vérifie lui-même la « fraîcheur » de son fichier de configuration et redémarre seul s'il a été modifié depuis son dernier démarrage (voir Section 9.2.4, « /etc/samba/smb.conf »).

9.4.1. Gestion des utilisateurs

Les utilisateurs Samba sont en premier lieu des comptes Unix. Pour créer un utilisateur, on commencera donc par créer un utilisateur unix avec `adduser`. L'utilisateur devra ensuite être ajouté dans la base spécifique Samba avec `pdbedit` :

```
pdbedit {-a} {-u} {utilisateur}
```

Par exemple :

```
root@ubuntu:~# pdbedit -a -u alice
```

ajoute l'utilisateur *alice* dans la base d'authentification Samba.

Si cet utilisateur ne se connectera que par Samba, il est possible de bloquer son compte unix avec l'option `-l` de `passwd`

```
root@ubuntu:~# passwd -l alice
```

```
Password changed.
```

```
root@ubuntu:~#
```

La commande `pdbedit` possède aussi une autre option à connaître : `-L`, qui permet de lister tous les utilisateurs de la base *tdbsam*. On pourra utiliser `-v` pour avoir des informations plus détaillées, et spécifier un nom d'utilisateur sur la ligne de commande afin de ne voir que les informations qui s'y rapportent.

```
root@ubuntu:~# pdbedit -L alice
```

```
alice:1001:Alice,,,
```

```
root@ubuntu:~# pdbedit -Lv alice
```

```
Unix username:      alice
```

```
NT username:
```

```
Account Flags:      [U          ]
```

```
User SID:           S-1-5-21-223759179-143051563-2494116590-3002
```

```
Primary Group SID:  S-1-5-21-223759179-143051563-2494116590-513
```

```
Full Name:          Alice,,,
```

```
Home Directory:     \\ubuntu\alice
```

```
HomeDir Drive:
```

```
Logon Script:
```

```
Profile Path:       \\ubuntu\alice\profile
```

```
Domain:             UBUNTU
```

```
Account desc:
```

```
Workstations:
```

```
Munged dial:
```

```
Logon time:         0
```

```
Logoff time:        mar, 19 jan 2038 04:14:07 CET
```

```
Kickoff time:       mar, 19 jan 2038 04:14:07 CET
```

```
Password last set:  ven, 29 jun 2007 22:03:58 CEST
```

```
Password can change: ven, 29 jun 2007 22:03:58 CEST
```

```
Password must change: mar, 19 jan 2038 04:14:07 CET
```

```
Last bad password  : 0
```

```
Bad password count  : 0
```

```
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

```
root@ubuntu:~#
```

Pour un simple changement de mot de passe Samba, on utilisera `smbpasswd` de la même manière que `passwd` :

```
root@ubuntu:~# smbpasswd alice
```

```
New SMB password:
```

```
Retype new SMB password:
```

```
root@ubuntu:~#
```

Chapitre 10. Chiffrement SSL/TLS

\$Revision: 1.14 \$

\$Date: 2007/07/10 22:01:08 \$

Faire circuler des protocoles sans chiffrement sur Internet peut paraître une hérésie. Mais cela s'explique facilement par le contexte qui a entouré la création de ces protocoles taxés aujourd'hui de non sécurisés. Aujourd'hui, grâce en particulier au projet *OpenSSL*, la plupart des protocoles peuvent utiliser une couche transport sécurisée par SSL ou TLS (le successeur de SSL). Ce chapitre détaille la sécurisation des différents protocoles vus jusqu'ici par le déploiement d'une petite PKI et l'utilisation de SSL/TLS.

10.1. Généralités

10.1.1. Problématique

La quasi-totalité des services et des protocoles utilisés aujourd'hui ont été spécifiés il y a un quart de siècle. A l'époque, Internet ne concernait que quelques milliers de chercheurs. On était alors assez peu concerné par des problèmes de confidentialité sur le réseau. Le résultat est que les protocoles conçus dans ce contexte sont en texte clair : tout ce qu'ils transportent est visible pour peu que l'on puisse capturer des trames réseau contenant ce protocole.

Parmi les remèdes à cette situation, on a vu des protocoles être remisés au rang d'antiquités et remplacés par de nouveaux plus sûrs. C'est le cas de telnet par exemple. Mais la solution la plus courante consiste à chiffrer le canal de communication sous-jacent au protocole. HTTPS, par exemple, n'est rien d'autre que du HTTP circulant sur une couche chiffrée *SSL* (Secure Socket Layer) et sur un port différent ¹.

Grâce à cette technique, il suffit de modifier « légèrement » les clients et les serveurs pour qu'ils utilisent une couche SSL.

10.1.2. Architecture

La couche SSL utilise souvent des certificats pour s'assurer de l'identité de chacun et pour chiffrer les communications. Nous devons donc créer ces certificats afin de sécuriser nos différents services en déployant une *PKI* : Public Key Infrastructure.

Le principe est le suivant. Le client qui veut se connecter à un serveur de manière sécurisée cherche essentiellement à :

- s'assurer de l'identité du serveur, afin d'éviter les détournements de connexions,
- chiffrer ses communications avec le serveur.

De son côté, le serveur peut aussi avoir besoin de vérifier l'identité du client (bien que ce soit rare en pratique).

Pour parvenir à ce résultat, le serveur devra prouver son identité au client avec un *certificat*. Ce certificat sera signé par une autorité de certification qui se porte garante de l'identité du serveur. Par exemple, lorsque l'on se connecte au site <https://www.nsa.gov>, le certificat renvoyé est signé par la société Verisign, qui est une autorité de certification (CA) ou *tiers de confiance*. Comme nous savons que Verisign est « sérieux » dans son processus de vérification d'identité, nous pouvons être sûr que le site est bien celui de la NSA. Cela n'implique pas que le site *est* de confiance; cela indique seulement que le site est bien celui qu'il prétend être.

Comme le certificat contient une clef publique, toutes nos communications avec ce site pourront être chiffrées avec un algorithme négocié à l'établissement de la connexion SSL.

¹D'autres protocoles, FTP/TLS par exemple, passent en mode chiffré dans la connexion ouverte. On appelle cette technique « upward negotiation ».

Pour déployer un serveur, il n'est cependant pas nécessaire de demander un certificat (plus exactement de demander à un tiers de confiance de signer notre certificat) : nous pouvons créer notre propre CA (Certificate Authority, autorité de certification). Bien sûr, dans ce cas, nous n'apportons plus la preuve au client que nous sommes bien *qui* nous prétendons être. En revanche, si seul le chiffrement nous intéresse, c'est tout à fait acceptable.

Une autre possibilité consiste à créer un certificat auto-signé (*self-signed certificate*) : aucune autorité de certification ne confirme l'authenticité du certificat. Nous n'utiliserons cependant pas cette méthode car la mise en place de SSL pour MySQL *requiert* l'utilisation du certificat de l'autorité ayant signé le certificat serveur

On aura remarqué que l'accès au site web de la NSA en HTTPS ne provoque l'apparition d'aucun message dans le navigateur. La raison est que par défaut, la plupart des navigateurs connaissent le certificat de Verisign. Voyant que le certificat de la NSA est signé par Verisign, et faisant par défaut confiance à Verisign, le navigateur fait confiance au site distant et n'affiche aucun message. En revanche, si nous signons nous-même nos certificats, les navigateurs afficheront un avertissement car le signataire ne sera pas connu, donc pas de confiance. Il faudra « faire avec », ou importer le certificat de notre propre autorité de certification dans le navigateur.

10.2. Préparatifs

10.2.1. Le paquetage OpenSSL

Afin de générer des certificats et de créer notre propre autorité de certification, nous devons installer le paquetage openssl.

```
root@ubuntu:~# apt-get install openssl
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Paquets suggérés :
  ca-certificates
Les NOUVEAUX paquets suivants seront installés :
  openssl
0 mis à jour, 1 nouvellement installés, 0 à enlever et 8 non mis à
jour.
Il est nécessaire de prendre 0o/1001ko dans les archives.
Après dépaquetage, 2359ko d'espace disque supplémentaires seront
utilisés.
Sélection du paquet openssl précédemment désélectionné.
(Lecture de la base de données... 18124 fichiers et répertoires
déjà installés.)
Dépaquetage de openssl (à partir de
.../openssl_0.9.8c-4build1_i386.deb) ...
Paramétrage de openssl (0.9.8c-4build1) ...

root@ubuntu:~#
```

OpenSSL permet de fixer un certain nombre de paramètres par défaut. Cela évitera de saisir des informations répétitives lors de la création des certificats (au moins un certificat de CA et un serveur). Pour modifier les valeurs par défaut de création de certificats, il faut éditer le fichier `/etc/ssl/openssl.cnf` et modifier les paramètres suivants :

- `countryName_default` : mettre le CC (country code) du pays à la place de « AU ». A priori, « FR ».
- `stateOrProvinceName_default` : remplacer « Some-State » par une valeur plus adaptée.

- `localityName_default` : mettre la ville où se trouve la société recevant le certificat.
- `0.organizationName_default` : mettre le nom de la société recevant le certificat.
- `organizationalUnitName_default` : nom du service utilisant le certificat. Il n'est pas nécessaire de prendre ce champ au pied de la lettre et l'on peut mettre *Serveur* par exemple.
- `commonName_default` : le nom d'hôte du serveur. Il ne faut pas se tromper ici sous peine de recevoir un pop-up supplémentaire du navigateur !
- `emailAddress_default` : une adresse email de contact dans la société recevant le certificat.

10.2.2. Création d'une autorité de certification

Le paquetage OpenSSL installe des scripts permettant de simplifier la génération de certificats. La manipulation d'**openssl** en ligne de commande étant particulièrement inhumaine, il est vivement recommandé de les utiliser, au moins pour la création de notre CA.

Avant de créer notre *PKI*, il est important de comprendre que créer une autorité de certification n'est pas anodin. Si la clef privée de notre CA est compromise, l'attaquante (*Mallory*) signer n'importe quel certificat (client ou serveur) à notre place, et, si elle est en mesure d'intercepter des communications qui nous sont destinées, pourra même faire passer pour notre serveur. La RFC 3647 ([RFC3647]) définit un certain nombre de bonnes pratiques pour la gestion d'infrastructures à clef publiques.

La première tâche dans la création de notre mini-PKI consiste donc à créer notre propre autorité de certification (afin de signer à terme nos propres certificats). On utilisera le script `/usr/lib/ssl/misc/CA.pl` pour la totalité de nos besoins.

```
root@ubuntu:~# /usr/lib/ssl/misc/CA.pl -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be
  incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
  or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Rhone]:
Locality Name (eg, city) [Souris City]:
Organization Name (eg, company) [Souris SARL]:
Organizational Unit Name (eg, section) [Servers]:
Common Name (eg, YOUR name) [ubuntu.example.com]:
Email Address [souris@example.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

```

Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number:
        93:02:12:ea:93:09:59:f3
    Validity
        Not Before: Jun 24 13:47:39 2007 GMT
        Not After  : Jun 23 13:47:39 2010 GMT
    Subject:
        countryName           = FR
        stateOrProvinceName   = Rhone
        organizationName      = Souris SARL
        organizationalUnitName = Servers
        commonName            = ubuntu.example.com
        emailAddress          = souris@example.com
    X509v3 extensions:
        X509v3 Subject Key Identifier:

        FD:12:E1:CA:84:81:87:A3:E4:A1:DD:43:5B:88:E3:0D:E0:5B:D4:25
        X509v3 Authority Key Identifier:

        keyid:FD:12:E1:CA:84:81:87:A3:E4:A1:DD:43:5B:88:E3:0D:E0:5B:D...
        DirName:/C=FR/ST=Rhone/O=Souris
        SARL/OU=Servers/CN=ubuntu.exa...
        serial:93:02:12:EA:93:09:59:F3

        X509v3 Basic Constraints:
            CA:TRUE
Certificate is to be certified until Jun 23 13:47:39 2010 GMT (1095
days)

Write out database with 1 new entries
Data Base Updated
root@ubuntu:~#

```

Comme nous avons préalablement renseigné `/etc/ssl/openssl.conf`, il nous suffit d'accepter toutes les valeurs par défaut. Il faudra en revanche utiliser un mot de passe raisonnable (*Enter PEM pass phrase*) pour chiffrer la clef privée.

10.2.3. Création d'un certificat serveur

Pour générer un certificat serveur, on invoquera `CA.pl` avec l'argument `-newreq-nodes`. On veillera à utiliser `-newreq-nodes` et non `-newreq`. Ce dernier chiffre la clef privée liée au certificat (et nécessite donc un mot de passe pour l'utiliser). Bien que dans l'absolu ce soit une bonne pratique (quelqu'un qui déroberait notre certificat et notre clef non protégée pourrait se faire passer pour nous), ça n'est pas très pratique en production : à chaque démarrage, les serveurs demanderont le mot de passe pour débloquer la clef, et il faudra donc être présent physiquement sur la console. Comme souvent, augmenter la sécurité induit une complication d'usage. Chacun pourra choisir en fonction du ratio sécurité / « praticité » désiré.

```

root@ubuntu:~# /usr/lib/ssl/misc/CA.pl -newreq-nodes
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'newkey.pem'
-----

```

You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank. For some fields there will be a default value, If you enter '.', the field will be left blank.

```
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Rhone]:
Locality Name (eg, city) [Souris City]:
Organization Name (eg, company) [Souris SARL]:
Organizational Unit Name (eg, section) [Servers]:
Common Name (eg, YOUR name) [ubuntu.example.com]:
Email Address [souris@example.com]:
```

Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []:
An optional company name []:
Request is in newreq.pem, private key is in newkey.pem
root@ubuntu:~#

A cet instant, nous avons un certificat (`newreq.pem`) contenant une demande de signature (CSR, Certificate Signing Request) ainsi qu'une clef privée associée (`newreq.pem`). Là aussi les opérations sont très simples puisque la majorité des informations a déjà été saisie dans `openssl.conf`.

Nous pouvons maintenant *signer* la demande de certificat avec la clef de notre CA. Cela permettra à ceux qui font confiance à notre autorité de certification de faire confiance à ce certificat, établissant ainsi une « chaîne de confiance » (*web of trust*).

```
root@ubuntu:~# /usr/lib/ssl/misc/CA.pl -sign
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:xxxxxx
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number:
        93:02:12:ea:93:09:59:f4
    Validity
        Not Before: Jun 24 13:47:54 2007 GMT
        Not After  : Jun 23 13:47:54 2008 GMT
    Subject:
        countryName           = FR
        stateOrProvinceName   = Rhone
        localityName          = Souris City
        organizationName      = Souris SARL
        organizationalUnitName = Servers
        commonName            = ubuntu.example.com
        emailAddress          = souris@example.com
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        Netscape Comment:
            OpenSSL Generated Certificate
        X509v3 Subject Key Identifier:
```

```
F8:06:7D:14:27:C2:76:07:24:12:16:D9:80:71:C2:23:36:2F:15:D2
```

```
X509v3 Authority Key Identifier:
```

```
keyid:FD:12:E1:CA:84:81:87:A3:E4:A1:DD:43:5B:88:E3:0D:E0:5B:D...
```

```
Certificate is to be certified until Jun 23 13:47:54 2008 GMT (365
days)
```

```
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
```

```
Write out database with 1 new entries
```

```
Data Base Updated
```

```
Signed certificate is in newcert.pem
```

```
root@ubuntu:~#
```

On déplacera ensuite le certificat généré et sa clef dans des emplacements plus appropriés, c'est à dire respectivement sous `/etc/ssl/cert/` et `/etc/ssl/private/`. On pourra aussi copier le certificat de notre CA.

```
root@ubuntu:~# mv newcert.pem /etc/ssl/certs/serveur.pem
```

```
root@ubuntu:~# mv newkey.pem /etc/ssl/private/serveur.key
```

```
root@ubuntu:~# cp demoCA/cacert.pem /etc/ssl/certs/cacert.pem
```

10.2.4. Liste de révocation

Emettre un certificat n'est heureusement pas irrévoquable. Il peut être compromis, le serveur démantelé, le porteur du certificat ne bénéficie plus du service, ... Dans ce cas, on pourra *révoquer* ce certificat, en le plaçant dans une CRL (Certificate Revocation List).

Cette CRL pourra ensuite être utilisée par tous ceux qui potentiellement peuvent rencontrer des certificats révoqués. En général, les CRL sont utilisées par les serveurs afin de ne plus fournir de service aux usagers qui n'y ont plus droit, mais on peut aussi utiliser des CRLs sur des clients afin de ne plus accepter les certificats serveurs révoqués (suite à un piratage par exemple).

```
root@ubuntu:~# openssl ca -revoke alice.pem -crl_reason
```

```
keyCompromise ①
```

```
Using configuration from /usr/lib/ssl/openssl.cnf
```

```
Enter pass phrase for ./demoCA//private/cakey.pem:
```

```
Revoking Certificate 930212EA930959F6.
```

```
Data Base Updated
```

```
root@ubuntu:~# openssl ca -gencrl ②
```

```
Using configuration from /usr/lib/ssl/openssl.cnf
```

```
Enter pass phrase for ./demoCA//private/cakey.pem:
```

```
-----BEGIN X509 CRL-----
```

```
MIIBi jCB9AIBATANBgkqhkiG9w0BAQUFADCBhTELMakGAlUEBhMCRlIx D jAMBgNV
BAgTBVJJob25lMRQwEgYDVQQKEwtTb3VyaXMGU0FSTDEQMA4GAlUEC xMHU2VydmVy
czE bMBkGAlUEAxMSdWJlbnR1LmV4YW1wbGUuY29tMSEwHwYJKoZIhvcNAQkBFhJz
b3VyaXNAZXhhbXBsZS5jb20XDTA3MDcwMzE1NTYwNl oXDTA3MDgwMjE1NTYwNl o
KjAoAgkAkWIS6pMJWfYXDTA3MDcwMzE1NTU1NVowDDAKBgNVHRUEAwBAAaOMAww
CgYDVVR0UBAMCAQEwDQYJKoZIhvcNAQEFBQADgYEAcq+U2r8S3TdQnT50Hm0XzHcy
inw04P43r5WJ2Xw5DHieJfDwpoASpBjrkXQeQdF3zc6e0GxSkJpvc051bdbQ3dx4
P Semqff+trxCapHA4pXBt2M3PqPAhze8bx7af6EQw27+k5HWGFbEew2ByAuqHa7S
jJteWU6hYkCymbqrDX4=
```

```
-----END X509 CRL-----
```

```
root@ubuntu:~#
```

- ① L'option `-revoke` demande la révocation du certificat passé en paramètre, tandis que `-crl_reason` donne la raison de la révocation. La liste des raisons de révocation est donnée dans la mague de man d'openssl/ca (ca(1ssl)).

- ② On peut ensuite générer une liste de révocation que l'on peut distribuer aux serveurs et/ou clients qui en ont besoin, et qui contient la liste signée de tous les certificats révoqués.

10.3. Déploiement

10.3.1. HTTPS

10.3.1.1. Configuration

Pour mettre en place le chiffrement pour Apache, on devra tout d'abord lui indiquer d'écouter sur le port 443 en complétant le fichier `/etc/apache2/ports.conf` :

```
Listen 443
```

On devra ensuite modifier les fichiers de configuration de *virtualhosts* dans `/etc/apache2/sites-available/` afin de préciser à quel port s'appliquent ces *virtualhost* :

```
NameVirtualHost *:80
NameVirtualHost *:443
```

```
<VirtualHost *:80>
...
```

- ① on remplace la « * » par « *:80 » et on ajoute un entrée pour le port 443. Cela permet d'indiquer à Apache qu'il y aura des *virtualhosts* sur ces deux ports.
 - ② chaque *virtualhost* devra dorénavant être accompagné du port auquel il est rattaché.
- Si un site doit être accessible en HTTP et HTTPS, il faudra créer deux entrées `<VirtualHost>`.

```
NameVirtualHost *:443
```

```
<VirtualHost *:80>
    ServerAdmin alice@exemple.org
    ...
    ServerSignature Off
</VirtualHost>
```

```
<VirtualHost *:443>
    ServerAdmin alice@exemple.org
    ServerName alice.exemple.org
    DocumentRoot /home/alice/monsieuebe/

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/serveur.pem
    SSLCertificateKeyFile /etc/ssl/private/serveur.key
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
    </Directory>

    ErrorLog /home/alice/meslogs/error_ssl.log

    # Possible values include: debug, info, notice, warn,
    error, crit,
```

```
# alert, emerg.
LogLevel info

CustomLog /home/alice/meslogs/access_ssl.log combined
ServerSignature Off
</VirtualHost>
```

On devra ensuite activer le module SSL avec **a2enmod** (voir Section 4.2.2, « Modules »), avant de redémarrer Apache.

```
root@ubuntu:~# a2enmod ssl
Module ssl installed; run /etc/init.d/apache2 force-reload to
enable.
root@ubuntu:~# /etc/init.d/apache2 force-reload
* Forcing reload of web server (apache2)...
  [ OK ]
root@ubuntu:~#
```

10.3.1.2. Redirection SSL

Si l'on désire force l'utilisation de SSL, on pourra utiliser les règles suivantes dans la configuration Apache. On devra auparavant activer `mod_rewrite` avec **a2enmod** (Section 4.2.2, « Modules »).

```
#
# Force SSL
RewriteEngine on
RewriteCond %{HTTPS} !=on [NC]
RewriteRule ^.*$ https://%{SERVER_NAME}%{REQUEST_URI} [R,L]
```

10.3.1.3. Filtrage

Coté serveur, on devra, pour finir, autoriser l'accès extérieur au port 443.

Exemple 10.1. Apache HTTPS: configuration du filtrage TCP en entrée

```

#
# #####
# TCP entrant
# Il faudra ouvrir des ports au fil de l'eau
# lors de la mise en place de
# services TCP (ssh, apache, ...).
# #####
#
-A TCP_IN -j TCP_INLIMITS
-A TCP_IN -j STATEFUL
-A TCP_IN -j TCP_SYNLIMITS
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_IN:" --log-level 6
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
# Ajouter les règles ici lors de l'installation de services TCP si
# ces services
# doivent être ouverts
#
-A TCP_IN -p tcp -m tcp --dport 80 -j ACCEPT
# Ouverture du port HTTPS
-A TCP_IN -s adresse_ip_autorisée -p tcp -m tcp --dport 443 -j
  ACCEPT ❶
# on peut aussi débloquer le port 80 pour tout le monde
-A TCP_IN -p tcp -m tcp --dport 443 -j ACCEPT ❷
#

```

- ❶ Règle autorisant l'accès au port 443/tcp (https) pour l'adresse *adresse_ip_autorisée* (qui peut aussi être un subnet). Cette règle peut être répétée autant de fois que nécessaire.
- ❷ Règle autorisant l'accès au port 443/tcp (https) pour tout le monde.

Un test avec un navigateur permet de tester le fonctionnement du serveur en mode HTTPS. Comme prévu, Firefox se plaint du certificat et nous explique que l'autorité de certification est inconnue.

Figure 10.1. Test HTTPS



10.3.2. FTP/TLS

ProFTPD permet aux clients de négocier une session SSL/TLS, chiffrant ainsi les connexion de contrôle et de transfert. Ce serveur, à l'instar d'Apache, utilise un système de modules permettant d'ajouter des fonctionnalités. On pourra faire varier les paramètres en fonction des besoins. Ceux utilisés ci-dessous suffisent pour établir un chiffrement entre le client et le serveur.

```
<IfModule mod_tls.c>
  TLSEngine on
  TLSLog /var/log/proftpd/tls.log

  # TLS est-il obligatoire ?
  TLSRequired off

  # Certificats serveur
  TLSRSACertificateFile /etc/ssl/certs/serveur.pem
  TLSRSACertificateKeyFile /etc/ssl/private/serveur.key

  # Authentifier les clients utilisant TLS ?
  TLSVerifyClient off

  # Autorise les renégotiations SSL/TLS sans les forcer.
  # Certains clients ne supportent pas les renégotiations et
  # ferment la connexion data.
  TLSRenegotiate required off
</IfModule>
```

Il y a malheureusement un petit accroc au tableau. Comme expliqué dans le chapitre consacré à ProFTPD (Section 7.3, « Filtrage »), le protocole FTP fonctionne selon deux modes, actif et passif. Lorsque les communications circulent en clair, aucun problème point de vue filtrage : le module

`nf_conntrack_ftp` scrute la connexion de contrôle et autorise les connexions nécessaires pour le fonctionnement du protocole dans les deux modes.

En revanche, lorsque les communications sont chiffrées ce module n'a aucun moyen de connaître le numéro de port négocié dans l'application puisqu'elle est chiffrée. Pour le mode actif, on peut résoudre le problème : le port source de la connexion *data* établie par le serveur vers le client sera 20, et on pourra donc explicitement laisser sortir les connexions ayant ce port source dans notre chaîne `TCP_OUT`. Mais en mode passif, le serveur va ouvrir un port > 1023 (négocié dynamiquement dans le protocole applicatif FTP) auquel le client tentera de se connecter. Netfilter n'ayant pas la possibilité d'accepter cette connexion entrante comme étant « RELATED », les paquets en provenance du client seront rejetés.

En résumé, on ne pourra faire, en SSL/TLS que des connexions FTP en mode actif. Pour cela, on devra donc autoriser le serveur FTP à se connecter au client dans la chaîne contenant les règles régissant le trafic sortant. On pourra bien sûr là aussi restreindre la liste des adresses avec lesquelles on veut faire du FTP.

```
#
# #####
# TCP sortant
# Cette machine initie des connexions HTTP vers
# fr.archive.ubuntu.com
# et security.ubuntu.com pour les mises à jour
# #####
#
-A TCP_OUT -j STATEFUL
-A TCP_OUT -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
  --limit 10/min -j LOG --log-prefix "TCP_OUT:" --log-level 6
-A TCP_OUT -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
#
# Règle pour le ftp actif
#
-A TCP_OUT -p -d adresse_ip_autorisée --sport 20 --dport 1024: -j
  ACCEPT ❶
# on peut aussi débloquent le ftp actif pour tout le monde
-A TCP_OUT -p --sport 20 --dport 1024: -j ACCEPT ❷
#
```

- ❶ Règle autorisant l'établissement de connexion du port 20/tcp (ftp-data) depuis le serveur vers l'adresse *adresse_ip_autorisée* (qui peut aussi être un subnet). Cette règle peut être répétée autant de fois que nécessaire.
- ❷ Règle autorisant l'établissement de connexion du port 20/tcp (ftp-data) depuis le serveur vers tout le monde.

10.3.3. MySQL/SSL

Contrairement aux services précédents, MySQL veut systématiquement vérifier le certificat du serveur et s'assurer qu'il a bien été signé par une autorité de certification connue. Il faudra donc que le client possède le certificat de l'autorité de certification ayant signé le certificat serveur. La configuration coté serveur consiste en deux lignes ajoutées dans la section `[mysqld]` du fichier `/etc/mysql/my.cnf`.

```
ssl-cert=/etc/ssl/certs/cert_serveur.pem
ssl-key=/etc/ssl/private/key_serveur.pem
```

Coté client, on emploiera l'option `--ssl-ca` pour demande l'utilisation de SSL tout en indiquant le fichier certificat du CA ayant signé le certificat du serveur. On peut aussi employer `--ssl-capath`

pour indiquer l'emplacement d'un répertoire contenant plusieurs certificats CA. Sous mysql, la requête `SHOW VARIABLES LIKE 'have_openssl'` ; permet de savoir si l'on utilise une connexion SSL et `SHOW STATUS LIKE 'Ssl_cipher'` ; donne la chaîne de chiffrement en cours d'utilisation.

```
alice@linus:~$ mysql -u dbadmin -h serveur --ssl-ca=/tmp/cacert.pem
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> SHOW VARIABLES LIKE 'have_openssl';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_openssl  | YES   |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql>
```

L'intérêt du chiffrement est évidemment lié à l'accès que l'on désire offrir au serveur : s'il n'est *bindé* que sur 127.0.0.1 ou que `skip-networking` est activé, les connexions ne seront que locales et le chiffrement n'a que peu d'intérêt. Comme pour FTP/TLS, le port TCP ne change pas et le chiffrement est négocié au sein de la connexion habituelle. Il n'y a donc pas de modification à effectuer en ce qui concerne le filtrage.

10.3.4. SMTP/TLS

Afin de permettre à Postfix d'utiliser TLS pour sécuriser ses échanges SMTP, nous devons lui fournir au minimum un certificat et une clef serveur. Le chemin de ces éléments est à mettre respectivement dans `smtpd_tls_cert_file` et `smtpd_tls_key_file`.

```
root@ubuntu:~# postconf -e
  smtpd_tls_cert_file=/etc/ssl/certs/cert_serveur.pem
root@ubuntu:~# postconf -e
  smtpd_tls_key_file=/etc/ssl/private/key_serveur.pem
root@ubuntu:~# invoke-rc.d postfix restart
* Stopping Postfix Mail Transport Agent postfix
[ OK ]
* Starting Postfix Mail Transport Agent postfix
[ OK ]
root@ubuntu:~#
```

Postfix permet d'ajuster beaucoup d'autres paramètres pour TLS, mais la plupart des valeurs qu'on trouve dans `postconf(5)`.

Chapitre 11. Déploiement et guide des opérations OpenVPN

\$Revision: 1.9 \$

\$Date: 2007/07/06 20:49:09 \$

Pour créer un VPN quelques années en arrière, il y avait deux possibilités : IPsec et le reste du monde. Le reste du monde était assez hétéroclite : vendeurs interlopes de boîtiers SSL et autres produits fermés, produits OpenSource aux fondements cryptographiques douteux, empilements protocolaires aussi inefficaces que prétentieux (A over B over C over... head). IPsec de son côté cumulait presque toutes les tares possibles : interopérabilité hasardeuse, complexité des protocoles mis en oeuvre (notamment pour l'échange de clefs), imbrication importante dans le kernel, fonctionnement hors des principes habituels (bypass des tables de routage, pas d'interface associée au trafic encapsulé selon les implémentations, ...).

Avec ce panorama, on comprend mieux l'intérêt porté à OpenVPN lors de son arrivée en 2001-2002. Parmi ses qualités on retiendra notamment :

- *simplicité* : OpenVPN est un simple binaire, identique côté client et serveur. Un fichier de configuration de moins de 10 lignes suffit en général pour déployer un VPN.
- *sécurité* : OpenVPN n'invente pas sa cryptographie, et s'appuie sur OpenSSL.
- *portabilité* : OpenVPN tourne sur la quasi totalité des OS du marché (Linux, Windows 2000/XP/Vista 32 et 64bits, {Open,FreeBSD,Net,DragonFly}BSD, MacOS X, Solaris, PocketPC...).
- *intégration* : OpenVPN s'intègre naturellement dans la gestion du réseau sur les machines. Une interface est créée pour chaque tunnel chiffré et les règles de la table routage *s'appliquent*. OpenVPN ne crée pas « d'exception » au fonctionnement habituel du réseau.
- *userspace* : OpenVPN fonctionne complètement en espace utilisateur.

Ce chapitre détaillera les étapes à suivre pour déployer OpenVPN entre des clients nomades et une passerelle de sécurité.

11.1. Installation

Le packaging OpenVPN est très léger. Décompacté il pèse à peine 1 Mo dûs presque exclusivement à la documentation et aux fichiers exemple.

```
root@ubuntu:~# apt-get install openvpn
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Reading state information... Fait
Les NOUVEAUX paquets suivants seront installés :
  openvpn
0 mis à jour, 1 nouvellement installés, 0 à enlever et 2 non mis à
jour.
Il est nécessaire de prendre 336ko dans les archives.
Après dépaquetage, 1012ko d'espace disque supplémentaires seront
utilisés.
Réception de : 1 http://fr.archive.ubuntu.com feisty/universe
openvpn 2.0.9-5...
336ko réceptionnés en 14s (23,4ko/s)

Préconfiguration des paquets...
```

```
Sélection du paquet openvpn précédemment désélectionné.  
(Lecture de la base de données... 18598 fichiers et répertoires  
déjà installés.)  
Dépaquetage de openvpn (à partir de ../openvpn_2.0.9-5_i386.deb)  
...  
Paramétrage de openvpn (2.0.9-5) ...  
Starting virtual private network daemon:.  
  
root@ubuntu:~#
```

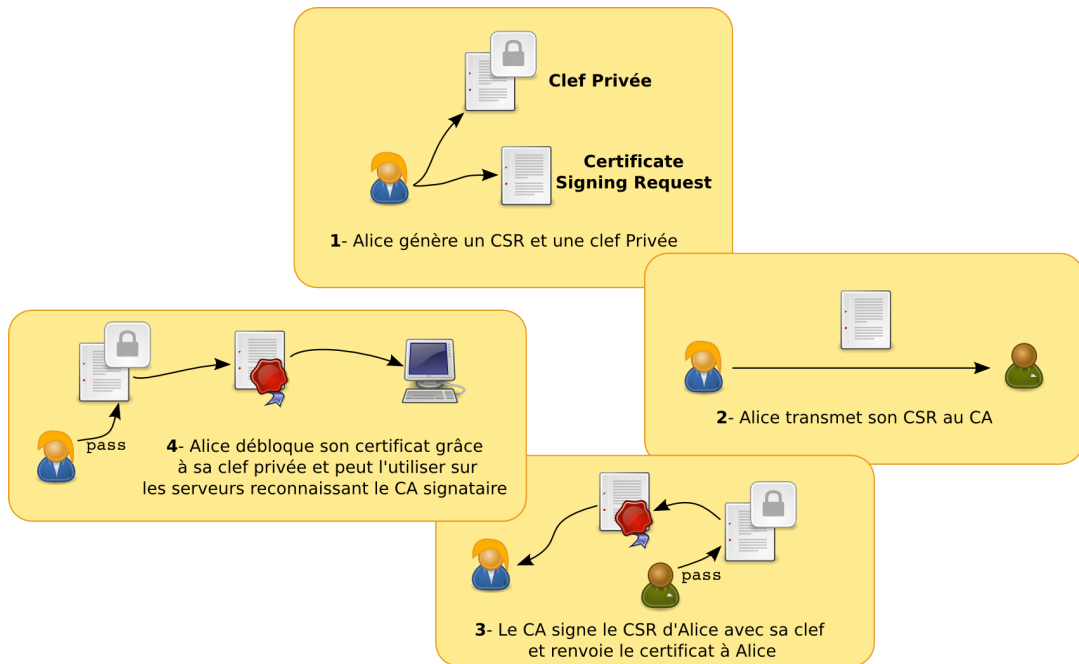
11.2. Configuration

Avant de procéder à la configuration d'OpenVPN, il faudra procéder à un choix d'architecture. L'authentification d'OpenVPN peut utiliser soit des clefs, soit des certificats. On peut évidemment utiliser les deux (certificats pour un client, clef pour un autre, ...), mais pour simplifier l'exploitation, on évitera d'avoir les deux.

Comme nous avons déjà pas mal avancé dans la création d'une petite PKI, nous utiliserons des certificats. L'avantage des certificats est que leur transmission est *potentiellement* beaucoup plus simple : le client peut générer son propre certificat (et sa propre clef) et transmettre à l'autorité de certification une « requête de signature de certificat » que le CA pourra signer. Ainsi, aucune donnée sensible ne transitera entre le client et le serveur et la seule signature du certificat par le CA lui permettra à OpenVPN d'autoriser le client à se connecter. Le client pourra d'ailleurs utiliser *un seul* certificat pour tous ses besoins cryptographiques (VPN, client HTTPS, déclaration d'impôts, MySQL, ...).

Nous aurons aussi un autre intérêt à utiliser les certificats : notre certificat serveur et notre CA sont déjà en place pour Postfix, MySQL, ProFTPD et Apache. Notre certificat serveur servira donc aussi pour OpenVPN.

Figure 11.1. Principe de fonctionnement d'une PKI



Dernier avantage : en fonctionnant en mode certificat, nous n'aurons qu'un fichier de configuration sur le serveur alors que dans le mode « clef partagée », nous avons un fichier de configuration (et un port tcp/udp) par client.

OpenVPN propose toute une batterie d'outils pour gérer les différents certificats dans `/usr/share/doc/openvpn/examples/easy-rsa/`. Mais ces scripts ne sont pas adaptés à notre situation, car nous avons déjà notre petite PKI en marche. Nous invoquerons donc **openssl** directement.

11.2.1. Génération des paramètres Diffie-Hellman

Diffie-Hellman est un protocole d'échange de clefs cryptographiques qui est utilisé pour changer périodiquement les clefs de chiffement dans une communication protégée par TLS. Ce changement de clefs apporte une fonctionnalité appelée *Perfect Forward Secrecy* (PFS), qui garanti que si votre clef est compromise, il ne sera pas possible à l'attaquant de décoder des anciens messages chiffrés avec une clef précédente : grâce à Diffie-Hellman elle est indépendante de la clef compromise.

L'échange Diffie-Hellman nécessite quelques paramètres (nombres premiers par exemple) qui sont particulièrement longs à calculer. On devra donc les pré-générer avec **openssl**.

```
root@ubuntu:~# openssl dhparam -out /etc/ssl/private/dh1024.pem
1024
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
.....+.....+.....+.....+
.+.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
...coupé...
.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
++*++*++*
root@ubuntu:~#
```

Le fichier `/etc/ssl/private/dh1024.pem` sera utilisé par la suite dans la configuration OpenVPN. Ce fichier n'est pas secret.

11.2.2. Génération de la requête de signature de certificat client

Nous avons déjà notre autorité de certification et notre certificat serveur. Il nous reste donc à générer des certificats pour les clients. Dans notre exemple, Alice va générer un certificat accompagné d'une demande de signature afin de le transmettre au CA.

```
alice@linus:~$ openssl req -days 3650 -new -keyout alice.key -out
alice.csr
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'alice.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be
  incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Rhône
```

```
Locality Name (eg, city) []:Souris City
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Souris
SARL
Organizational Unit Name (eg, section) []:Servers
Common Name (eg, YOUR name) []:linus.exemple.org
Email Address []:alice@exemple.org
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
alice@linus:~$
```

La clef privée générée sera protégée par un mot de passe. Cela protège la clef en cas de vol. En revanche, on devra taper un mot de passe lors de l'établissement de la connexion. Si l'on désire une clef sans mot de passe, on ajoutera l'option `-nodes` à la ligne de commande d'`openssl`.

Le « CSR » généré, alice peut l'envoyer au CA. Dans notre exemple, elle dépose directement son CSR sur le serveur du CA mais dans la pratique, il est recommandé d'utiliser un CA déconnecté du réseau (voir Section 10.2.2, « Création d'une autorité de certification »).

```
alice@linus:~$ scp alice.csr root@ubuntu:
Enter passphrase for key '/home/alice/.ssh/id_dsa':
alice.csr                                100% 720
0.7KB/s 00:00
alice@linus:~$
```

11.2.3. Signature de la demande de signature

Le CA utilise ensuite sa clef privée afin de signer la demande émise par Alice.

```
root@ubuntu:~# openssl ca -days 3650 -out alice.pem -in alice.csr
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/akey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number:
    93:02:12:ea:93:09:59:f6
  Validity
    Not Before: Jun 30 19:15:54 2007 GMT
    Not After : Jun 27 19:15:54 2017 GMT
  Subject:
    countryName           = FR
    stateOrProvinceName  = Rhone
    organizationName     = Souris SARL
    organizationalUnitName = Servers
    commonName            = linus.exemple.org
    emailAddress         = alice@exemple.org
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    Netscape Comment:
      OpenSSL Generated Certificate
    X509v3 Subject Key Identifier:
      00:00:7F:25:F7:A6:63:83:53:68:99:ED:DF:6E:08:D3:77:21:AC:37
    X509v3 Authority Key Identifier:
```

```
keyid:FD:12:E1:CA:84:81:87:A3:E4:A1:DD:43:5B:88:E3:0D:E0:5B:D4:25

Certificate is to be certified until Jun 27 19:15:54 2017 GMT (3650
days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
root@ubuntu:~#
```

La signature du CSR crée un certificat signé par notre CA que l'on pourra renvoyer à Alice (dans notre exemple, elle se sert elle-même avec **scp**). Elle pourra ensuite l'utiliser pour prouver son identité à tous les serveurs qui nous acceptent en tant qu'autorité de confiance. Alice pourra aussi vérifier l'identité des serveurs auxquels elle se connecte en vérifiant que leur certificats sont bien signés par notre CA.

```
alice@linus:~$ scp root@ubuntu:alice.pem .
Enter passphrase for key '/home/alice/.ssh/id_dsa':
alice.pem                                100% 3297
   3.2KB/s   00:00
alice@linus:~$
```

11.2.4. Fichier de configuration serveur

Le fichier de configuration serveur sera déposé dans `/etc/openvpn` et devra avoir l'extension `.conf`.

```
port 1194 ❶
proto udp

dev tun ❷

ca /etc/ssl/certs/cacert.pem ❸
cert /etc/ssl/certs/cert_serveur.pem
key /etc/ssl/private/key_serveur.pem
dh /etc/ssl/private/dh1024.pem

server 192.168.18.0 255.255.255.0 ❹
ifconfig-pool-persist ipp.txt ❺
keepalive 10 120 ❻

comp-lzo ❼

status /var/log/openvpn-status.log ❽
verb 3 ❾
```

- ❶ Définition du port et du protocole. OpenVPN peut fonctionner sur UDP ou TCP. On préférera UDP si possible, plus léger.
- ❷ Mode tunnel. OpenVPN peut fonctionner en mode tunnel (routé) ou tap (bridgé). Ce dernier mode consituera un bridge entre le LAN du client et le LAN du serveur, qui seront virtuellement connectés sur un même switch.
- ❸ Chemins des différents certificats (CA, serveur, clef privée) et des paramètres Diffie-Hellman
- ❹ Subnet utilisé pour l'attribution automatique d'adresse au client
- ❺ Demande à OpenVPN de conserver une association client/IP (et donne le chemin du fichier d'association).

- ⑥ Ce paramètre permet d'envoyer régulièrement (toutes les 10 secondes ici) un message au client afin qu'il puisse vérifier le fonctionnement du lien. Si au bout de 120 s. le client ne reçoit pas de paquets, il redémarrera son process OpenVPN. Le serveur attendra de son côté $120 \times 2 = 240$ secondes avant de redémarrer.
- ⑦ Active la compression du trafic.
- ⑧ OpenVPN écrira régulièrement son état dans ce fichier : statistiques de trafic, liste des clients, ...
- ⑨ Verbose des logs.

11.2.5. Fichier de configuration client

La configuration client est encore plus simple que celle du serveur. Elle comprend essentiellement les paramètres IP d'établissement du tunnel (dont l'adresse du serveur) et la liste des certificats à mettre en œuvre.

```
remote ubuntu.example.com 1194 ①
proto udp ②
dev tun ③

client ④

ca /etc/ssl/certs/cacert.pem ⑤
cert alice.pem
key alice.key

tls-remote ubuntu.example.com ⑥

comp-lzo ⑦
verb 3 ⑧
```

- ① Adresse IP et port du serveur distant.
- ② Choix du protocole (doit correspondre au choix sur le serveur).
- ③ Mode tunnel (doit correspondre au choix sur le serveur).
- ④ Indique à OpenVPN qu'il pourra recevoir des paramètres par `push` (voir Section 11.2.7, « Autres paramètres »)
- ⑤ Certificat du CA et certificat client accompagné de sa clef.
- ⑥ Permet de vérifier que le *commonName* du certificat serveur est bien celui attendu.
- ⑦ Active la compression sur le lien (doit correspondre au choix sur le serveur).
- ⑧ Verbose des logs.

En utilisant un nom de certificat générique au lieu de `alice`, on pourra même utiliser la même configuration pour tous les clients.

11.2.6. Test de la configuration

Pour vérifier le fonctionnement du tunnel, on devra démarrer le serveur avec `invoke-rc.d` (`invoke-rc.d openvpn start`). Si le démarrage passe sans accroc, on pourra tester la configuration cliente depuis le poste distant. Ce test peut être fait directement « à la main », permettant de voir le résultat de l'opération sans avoir à consulter les logs.

```
root@linus:/etc/openvpn# openvpn alice.cfg
Thu Jun 28 13:59:48 2007 OpenVPN 2.0.9 i486-pc-linux-gnu [SSL]
[LZO] [EPOLL] ...
Thu Jun 28 13:59:48 2007 IMPORTANT: OpenVPN's default port number
is now 1194,
based on an official port number
assignment by IANA.
```

```
OpenVPN 2.0-beta16 and earlier used 5000
as the defa...
Enter Private Key Password:xxxxxxx
Thu Jun 28 13:59:51 2007 LZO compression initialized
Thu Jun 28 13:59:51 2007 Control Channel MTU parms [ L:1542 D:138
EF:38 EB:0 E ]
Thu Jun 28 13:59:51 2007 Data Channel MTU parms [ L:1542 D:1450
EF:42 EB:135 E ]
Thu Jun 28 13:59:51 2007 Local Options hash (VER=V4): '41690919'
Thu Jun 28 13:59:51 2007 Expected Remote Options hash (VER=V4):
'530fddd'
Thu Jun 28 13:59:51 2007 UDPv4 link local: [undef]
Thu Jun 28 13:59:51 2007 UDPv4 link remote: 192.168.17.139:1194
Thu Jun 28 13:59:52 2007 TLS: Initial packet from
192.168.17.139:1194, sid=0b28e
Thu Jun 28 13:59:52 2007 VERIFY OK: depth=1,
/C=FR/ST=Rhone/O=Souris_SARL/OU=...
Thu Jun 28 13:59:52 2007 VERIFY X509NAME OK:
/C=FR/ST=Rhone/L=Souris_City/O=S...
Thu Jun 28 13:59:52 2007 VERIFY OK: depth=0,
/C=FR/ST=Rhone/L=Souris_City/O=S...
Thu Jun 28 13:59:52 2007 Data Channel Encrypt: Cipher 'BF-CBC'
initialized wi...
Thu Jun 28 13:59:52 2007 Data Channel Encrypt: Using 160 bit
message hash 'SHA1'
Thu Jun 28 13:59:52 2007 Data Channel Decrypt: Cipher 'BF-CBC'
initialized wi...
Thu Jun 28 13:59:52 2007 Data Channel Decrypt: Using 160 bit
message hash 'SHA1'
Thu Jun 28 13:59:52 2007 Control Channel: TLSv1, cipher TLSv1/SSLv3
DHE-RSA-AES256
Thu Jun 28 13:59:52 2007 [ubuntu.example.com] Peer Connection
Initiated with ...
Thu Jun 28 13:59:53 2007 SENT CONTROL [ubuntu.example.com]:
'PUSH_REQUEST' (s...
Thu Jun 28 13:59:53 2007 PUSH: Received control message:
'PUSH_REPLY,route 19...
Thu Jun 28 13:59:53 2007 OPTIONS IMPORT: timers and/or timeouts
modified
Thu Jun 28 13:59:53 2007 OPTIONS IMPORT: --ifconfig/up options
modified
Thu Jun 28 13:59:53 2007 OPTIONS IMPORT: route options modified
Thu Jun 28 13:59:53 2007 TUN/TAP device tun1 opened
Thu Jun 28 13:59:53 2007 ifconfig tun1 192.168.18.6 pointopoint
192.168.18.5 ...
Thu Jun 28 13:59:53 2007 route add -net 192.168.18.1 netmask
255.255.255.255 ...
Thu Jun 28 13:59:53 2007 Initialization Sequence Completed
```

11.2.7. Autres paramètres

OpenVPN possède un nombre conséquent de paramètres que l'on pourra découvrir sur le site officiel [<http://www.openvpn.org>] ou dans sa page de man (`openvpn(8)`). Certains sont très utilisés et méritent une attention particulière.

- `push` permet d'envoyer des paramètres de configuration au client. C'est extrêmement pratique puisque l'on peut modifier la configuration du client sans avoir à lui renvoyer un nouveau fichier

de configuration. Pour que le client accepte des paramètres du serveur, il devra avoir `pull` ou `client` dans son propre fichier de configuration.

- `route` suivi d'un subnet et d'un masque permet d'insérer une route dans la table de routage de l'OS lorsque la connexion sera active. Par exemple, si un client utilise la directive `route 192.168.19.0 255.255.255.0`, cette route sera ajoutée dans sa table de routage lorsque le tunnel sera monté et le *next-hop* pour ce subnet sera le serveur auquel ce client sera connecté. Pour plus de souplesse, on préférera envoyer ce paramètre au client avec la commande **push** : `push "route 192.168.19.0 255.255.255.0"`
- `user` et `group` demandent à **openvpn** de déléguer les droits de *root* dès que possible, et d'utiliser l'utilisateur et le groupe donnés en paramètres. En cas de faille dans OpenVPN permettant à un attaquant de prendre le contrôle du démon (débordement de pile par exemple), cet attaquant héritera des droits restreints de l'utilisateur spécifié.
- `chroot` force OpenVPN à s'installer sous une nouvelle racine du système de fichiers (voir l'entrée du glossaire `chroot`). En cas de faille, l'attaquant sera cantonné sous le répertoire pointé par `chroot`.
- `curl-verify` permet d'indiquer à OpenVPN l'emplacement de la liste de révocation de certificats. On l'utilise en général sur le serveur pour indiquer la liste des certificats clients qui ne doivent plus avoir accès au service (voir aussi Section 10.2.4, « Liste de révocation »).

11.2.8. Démarrage au boot

Le comportement au boot d'OpenVPN est défini dans `/etc/default/openvpn`. Le paramètre `AUTOSTART` liste les configurations OpenVPN à charger au démarrage. Ces configurations correspondent à des fichiers `*.conf` situés dans `/etc/openvpn`. Par exemple, si l'on a :

```
AUTOSTART="vpn1 vpn2"
```

OpenVPN essaiera automatiquement de charger les configuration `/etc/openvpn/vpn1.conf` et `/etc/openvpn/vpn2.conf` au boot. Si `AUTOSTART` est à *none*, aucune configuration ne sera chargée (donc OpenVPN ne démarrera pas) tandis que si ce paramètre est à *all*, OpenVPN tentera de charger toutes les configurations se terminant en `.conf`. Coté serveur, on aura dans notre cas une seule configuration (puisque le mode de fonctionnement utilisé ici est « multi-client ». En revanche, un client peut avoir de multiples configurations (i.e. des VPN vers des destinations différentes). Il pourra dans ce fichier spécifier le liste des tunnels qu'ils souhaite monter au démarrage de la machine.

Un autre paramètre, `STATUSREFRESH`, pour lequel nous pouvons spécifier un nombre entier, demande au script de démarra d'OpenVPN d'invoquer le service avec le paramètre `--status`. Cela aura pour effet de créer un fichier d'état dans `/var/run/openvpn.conf.status` ou `conf` est le nom de fichier de configuration. Si cette variable est indéfinie ou est 0, OpenVPN ne sera pas invoqué avec `--status`. Dans le cas contraire, le fichier d'état sera créé et rafraîchi toutes les *n* secondes, ou *n* est la valeur affectée à la variable. Si l'on utilise cette possibilité, on devra supprimer la directive `status` du fichier de configuration.

```
root@ubuntu:~# cat /var/run/openvpn.openvpn.status
OpenVPN CLIENT LIST
Updated,Sun Jul  1 17:51:02 2007
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
alice.example.org,213.245.112.213:1194,9610,9795,Sun Jul  1
 17:31:53 2007
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
192.168.18.6,alice.example.org,213.245.112.213:1194,Sun Jul  1
 17:31:53 2007
GLOBAL STATS
Max bcst/mcast queue length,0
END
```

root@ubuntu:~#

11.3. Filtrage

Le filtrage pour OpenVPN est très simple : il suffit de laisser entrer les paquets à destination du port qui lui est dédié (1194/udp, éventuellement tcp selon configuration).

Exemple 11.1. OpenVPN : configuration du filtrage UDP en entrée

```
#
# #####
# UDP entrant
# L'appel à STATEFUL suffit pour accepter les réponses DNS
# Il faudra cependant ouvrir des ports au fil de l'eau lors de la
# mise en place
# de services UDP (DNS, NTP par exemple).
# #####
#
-A UDP_IN -j STATEFUL
# Ajouter les règles ici lors de l'installation de services UDP si
# ces services
# doivent être ouverts
-A UDP_IN -s adresse_ip_autorisée -p udp --dport 1194 -j ACCEPT ❶
# on peut aussi débloquer le port 1194 pour tout le monde
-A UDP_IN -p udp --dport 1194 -j ACCEPT ❷
#
```

- ❶ Règle autorisant l'accès au port 1194/udp (openvpn) pour l'adresse *adresse_ip_autorisée* (qui peut aussi être un subnet). Cette règle peut être répétée autant de fois que nécessaire.
- ❷ Règle autorisant l'accès au port 1194/udp (openvpn) pour tout le monde.

Le serveur OpenVPN sera probablement aussi *routeur* : il fera transiter les paquets reçus depuis les clients VPN vers le réseau local et vice-versa. On devra donc utiliser la chaîne FORWARD de manière intensive afin d'appliquer les règles de filtrage qui s'imposent. On notera que les interfaces créées par OpenVPN en mode tunnel sont appelées *tun*, et que l'on peut utiliser le *wildcard* *tun+* avec IPtables pour désigner « toutes les interfaces tunnel ».

Annexe A. Firewall de base

Ce fichier est décrit au chapitre 2 (Section 2.4.3, « Filtrage de base »). Il est reproduit ici sans annotations afin de faciliter le copier/coller.



Fichier à modifier

Des valeurs nécessitent d'être modifiées pour que ce fichier soit utilisable. Un devra changer au moins les serveurs DNS. Toutes les lignes précédées de « ## » doivent être étudiées afin d'être modifiées le cas échéant.

```
#
#
*filter
#
# Création et remise à zéro des chaînes
#
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
:DROP_ME - [0:0]
:ICMP_IN - [0:0]
:ICMP_OUT - [0:0]
:STATEFUL - [0:0]
:TCP_IN - [0:0]
:TCP_SYNLIMITS - [0:0]
:TCP_INLIMITS - [0:0]
:TCP_OUT - [0:0]
:UDP_IN - [0:0]
:UDP_OUT - [0:0]
#
# #####
# INPUT Dispatch
# #####
#
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state INVALID -j DROP_ME
-A INPUT -s 127.0.0.0/255.0.0.0 -i ! lo -j DROP_ME
-A INPUT -p tcp -j TCP_IN
-A INPUT -p udp -j UDP_IN
-A INPUT -p icmp -j ICMP_IN
#
# #####
# OUTPUT Dispatch
# #####
#
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -p udp -j UDP_OUT
-A OUTPUT -p tcp -j TCP_OUT
-A OUTPUT -p icmp -j ICMP_OUT
-A OUTPUT -j STATEFUL
-A OUTPUT -j REJECT
#
# #####
# STATEFUL : accepte les paquets liés à une connexion existante
# #####
#
```

```

-A STATEFUL -m state --state RELATED,ESTABLISHED -j ACCEPT
-A STATEFUL -j RETURN
#
# #####
# ICMP entrant
# #####
#
-A ICMP_IN -j STATEFUL
-A ICMP_IN -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A ICMP_IN -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A ICMP_IN -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A ICMP_IN -p icmp -m icmp --icmp-type 8 -m limit --limit 5/sec -j
ACCEPT
#
# #####
# ICMP sortant
# #####
#
-A ICMP_OUT -j STATEFUL
-A ICMP_OUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
#
# #####
# TCP entrant
# Il faudra ouvrir des ports au fil de l'eau
# lors de la mise en place de
# services TCP (ssh, apache, ...).
# #####
#
-A TCP_IN -j TCP_INLIMITS
-A TCP_IN -j STATEFUL
-A TCP_IN -j TCP_SYNLIMITS
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
--limit 10/min -j LOG --log-prefix "TCP_IN:" --log-level 6
-A TCP_IN -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
# Ajouter les règles ici lors de l'installation de services TCP si
ces services
# doivent être ouverts
##-A TCP_IN -p tcp --dport 80 -j ACCEPT
##-A TCP_IN -p tcp --dport 443 -j ACCEPT
##-A TCP_IN -p tcp --dport 3306 -j ACCEPT
##-A TCP_IN -p tcp --dport 21 -j ACCEPT
##-A TCP_IN -p tcp --dport 22 -j ACCEPT
#
# #####
# TCP sortant
# Cette machine initie des connexions HTTP vers
fr.archive.ubuntu.com
# et security.ubuntu.com pour les mises à jour
# #####
#
-A TCP_OUT -j STATEFUL
-A TCP_OUT -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m limit
--limit 10/min -j LOG --log-prefix "TCP_OUT:" --log-level 6
-A TCP_OUT -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -j DROP
-A TCP_OUT -p tcp -d 194.2.0.36 --dport 80 -j ACCEPT
-A TCP_OUT -p tcp -d 82.211.81.138 --dport 80 -j ACCEPT
-A TCP_OUT -p tcp -d 91.189.88.31 --dport 80 -j ACCEPT
#

```

```

# ftp-data to clients
##-A TCP_OUT -p tcp --sport 20 --dport 1024: -j ACCEPT
#
# #####
# Limitation des connexions TCP entrantes
# Les connexions trop nombreuses sont rejetées.
# #####
#
# Si la connexion est dans les limites fixées, on retourne d'ou
l'on vient
-A TCP_SYNLIMITS -p tcp -m tcp --syn -m limit --limit 1/sec
--limit-burst 10 -j RETURN
#
# Sinon, on loggue
#
-A TCP_SYNLIMITS -m limit --limit 1/min -j LOG --log-prefix
"TCP_SYNLIMITS:" --log-level 6
-A TCP_SYNLIMITS -j REJECT
#
# #####
# Limitation TCP entrant
# Limitation du trafic TCP entrant
# Le trafic hors limite est droppé.
# #####
#
-A TCP_INLIMITS -p tcp --dport 80 -m recent --name HTTP_DOS --set
-A TCP_INLIMITS -p tcp --dport 80 -m recent --name HTTP_DOS
--update --hitcount 20 --seconds 1 -j DROP
## On peut aussi utiliser LIMIT
##-A TCP_INLIMITS -p tcp --dport 80 -m limit --limit 10/sec
--limit-burst 100 -j RETURN
##-A TCP_INLIMITS -p tcp --dport 80 -j DROP
-A TCP_INLIMITS -j RETURN
#
# #####
# UDP entrant
# L'appel à STATEFUL suffit pour accepter les réponses DNS
# Il faudra cependant ouvrir des ports au fil de l'eau lors de la
mise en place
# de services UDP (DNS, NTP par exemple).
# #####
#
-A UDP_IN -j STATEFUL
# Ajouter les règles ici lors de l'installation de services UDP si
ces services
# doivent être ouverts
#
# #####
# UDP sortant
# -remplacer SERVEUR_DNS par le serveur DNS et répéter la ligne
pour chacun des
# serveurs (primaire, secondaire, etc...)
# -remplacer SERVEUR_NTP par l'adresse IP du serveur NTP si ce
protocole est
# utilisé
# #####
#
-A UDP_OUT -p udp -m udp -j STATEFUL

```

```

-A UDP_OUT -d 192.168.0.254 -p udp -m udp --dport 53 -j ACCEPT
## Remplacer SERVEUR_DNS1, SERVEUR_DNS2 et SERVEUR_NTP par les
   bonnes valeurs
##-A UDP_OUT -d SERVEUR_DNS1 -p udp -m udp --dport 53 -j ACCEPT
##-A UDP_OUT -d SERVEUR_DNS2 -p udp -m udp --dport 53 -j ACCEPT
##-A UDP_OUT -d SERVEUR_NTP -p udp -m udp --dport 123 -j ACCEPT
## Dans le cas surprenant ou le serveur serait en DHCP
## -A UDP_OUT -p udp -m udp --sport 68 --dport 67 -j ACCEPT
#
# #####
# DROP_ME : la chaîne qui jette en laissant des traces dans les
   logs
# Par défaut, cette chaîne poubellise les paquets en silence
# En changeant la dernière règle par les deux commentées, on
   notifie la source
# du rejet du paquet, c'est plus conforme à la norme. Après, chacun
   décide s'il
# vaut mieux se conformer à la norme avec du trafic qui n'a pas
   lieu d'être...
# --limit permet d'éviter de mettre la machine à genoux en cas de
   déni
# de service
# #####
#
-A DROP_ME -p tcp -m limit --limit 10/min -j LOG --log-prefix
  "DROP:" --log-level 6
-A DROP_ME -p udp -m limit --limit 10/min -j LOG --log-prefix
  "DROP:" --log-level 6
## On pourra utiliser REJECT si l'on souhaite être poli
##-A DROP_ME -p tcp -j REJECT --reject-with tcp-reset
##-A DROP_ME -p udp -j REJECT --reject-with icmp-port-unreachable
-A DROP_ME -j DROP
#
COMMIT
#

```

Annexe B. GNU Free Documentation License

Version 1.2, November 2002
Copyright © 2000,2001,2002 Free Software Foundation, Inc.

Free Software Foundation, Inc.
51 Franklin St, Fifth Floor,
Boston,
MA
02110-1301
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Version 1.2, November 2002

B.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

B.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

B.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

B.4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

B.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

GNU FDL Modification Conditions

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

B.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in

parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

B.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

B.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

B.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

B.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received

copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

B.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

B.12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Sample Invariant Sections list

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

Sample Invariant Sections list

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Glossary

Définitions des termes techniques ou acronymes utilisés dans ce document.

A

AC	Voir Autorité de certification.
Autorité de certification	Tiers de confiance ayant autorité pour certifier de l'identité d'une personne physique ou morale. Les autorités de certification contresignent des certificats afin d'attester de l'identité du propriétaire. En anglais, « Certification Authority » ou « Certificate Authority ».

C

CA	Voir Autorité de certification.
chroot	La commande chroot permet de changer créer un nouvel environnement dans lequel la racine du filesystem sera modifiée (en fonction du paramètre donné) et dans lequel on pourra exécuter un processus. Appelé parfois une « prison chroot » (<i>chroot jail</i>), ce système permet de ne pas exposer le vrai système de fichiers en cas de défaillance du logiciel ainsi <i>chrooté</i> .

D

DBA	DataBase Administrator. L'administrateur de la base de données.
Diffie-Hellman	Protocole cryptographique publié en 1976 par Whitfield Diffie et Martin Hellman et permettant à deux participants de négocier une clef sans qu'un attant en position d'intercepter les échanges puisse en déduire cette clef.
DoS	Deni de Service. Attaque consistant à provoquer une rupture du service fourni.
DDoS	Deni de Service Distribué (Distributed Reflected Denial of Service). Attaque consistant à provoquer un <i>DoS</i> à partir de plusieurs machines.
DRDoS	Deni de Service Distribué Réfléchi (Distributed Reflected Denial of Service). Attaque constitué à partir des réponses (SYN+ACK, RST) renvoyées en réponse à des paquets dont l'adresse IP source spoofée est celle de la victime ([Gibson2002]).

E

easter-egg	Littéralement « Œuf de Pâques ». C'est une surprise cachée par les développeurs dans un logiciel. La procédure pour y accéder n'est généralement pas divulguée, bien que dans le cas de logiciels OpenSource il soit difficile d'en masquer l'existence.
------------	--

F

FQDN	<i>Fully Qualified Domain Name</i> . Nom de domaine complètement qualifié. Désigne un nom d'hôte contenant aussi le nom de domaine la machine. Pour la machine <i>www.exemple.com</i> , le nom d'hôte « court » est <i>www</i> , son domaine est « <i>exemple.com</i> » tandis que son nom d'hôte FQDN est
------	--

« `www.exemple.com` ». On peut obtenir le FQDN configuré pour une machine avec l'option `--fqdn` de la command **hostname** :

```
alice@linus:~$ hostname
linus
alice@linus:~$ hostname --fqdn
alice.exemple.com
alice@linus:~$
```

I

ISN *Initial Sequence Number*. Numéro de séquence initial choisi pour un connexion TCP. Ce numéro doit être le plus difficile à deviner possible, afin d'éviter le vol ou la perturbation de connexion. Il sont partiellement générés par des algorithmes cryptographiques. Les faiblesses liées à une mauvaise génération d'ISN sont connues depuis longtemps, et la qualité cryptographique des ISN à fait l'objet de plusieurs études (dont [Zalewski2001] et [Zalewski2002]).

K

KISS Acronyme pour « Keep It Simple, Stupid » (garder la chose simple, voir stupide).

L

LAMP Acronyme pour Linux/Apache/MySQL/PHP.

M

Man In The Middle (MITM) Attaque qui consiste à se faire passer pour un serveur au yeux d'un client. Cela est généralement possible si l'on se trouve *entre* le client et le serveur (« In The Middle »).

P

Partage Voir Share.

PKI Voir Public Key Infrastructure.

Public Key Infrastructure Infrastructure à Clefs Publiques. Désigne l'ensemble des composants et des processus permettant de mettre en oeuvre une gestion et une authentification par certificats et, éventuellement, une autorité de certification.

S

Secure Sockets Layer Protocole de communication permettant, après négociations des participants, de remplacer la couche transport originale par une couche transport chiffrée.

SGBD Voir Système de Gestion de Bases de Données.

Share Ressource partagée par un serveur NetBIOS en utilisant le protocole SMB ou CIFS. Synonyme de Partage.

SGBD Voir Structured Query Language.

SSL	Voir Secure Sockets Layer.
Structured Query Language	Langage de requête structuré. Langage permettant d'interroger une base de données. Toutes les base de données supportent aujourd'hui ce langage de requête, mais elles ne supportent souvent qu'un sous ensemble du langage.
Système de Gestion de Bases de Données	Logiciel permettant de gérer des bases de données.

Bibliographie

RFCs

- [RFC792] RFC 792 - Internet Control Message Protocol Jon Postel <http://www.ietf.org/rfc/rfc792.html>
- [RFC821] RFC 821 - Simple Mail Transfer Protocol Jon Postel <http://www.ietf.org/rfc/rfc821.html> [<http://www.ietf.org/rfc/rfc821.html>]
- [RFC1812] RFC 1812 - Requirements for IP Version 4 Routers F. Baker <http://www.ietf.org/rfc/rfc792.html>
- [RFC2142] RFC 2142 - Mailbox Names for Common Services, Roles and Functions D. Crocker <http://www.ietf.org/rfc/rfc2142.html>
- [RFC2617] RFC 2617 - HTTP Authentication: Basic and Digest Access Authentication J. FranksP. Hallam-BakerJ. HostetlerS. LawrenceP. LeachA. LuotonenL. Stewart juin 1999 <http://www.ietf.org/rfc/rfc2617.html> [<http://www.ietf.org/rfc/rfc2069.html>]
- [RFC3647] RFC 3647- Internet X.509 Public Key Infrastructure - Certificate Policy and Certification Practices Framework S. ChokhaniW. FordR. SabettC. Merrills. Wu novembre 2003 <http://www.ietf.org/rfc/rfc3647.html>

Livres

- [LARTC] Linux Advanced Routing & Traffic Control Bert HubertThomas Graf (Section Author)<tgraf@suug.ch>Gregory Maxwell (Section Author)Remco van Mook (Section Author)<remco@virtu.nl>Martijn van Oosterhout (Section Author)<kleptog@cupid.suninternet.com>Paul B Schroeder (Section Author)<paulsch@us.ibm.com>Jasper Spaans (Section Author)<jasper@spaans.ds9a.nl>Pedro Larroy (Section Author)<piotr@member.fsf.org> <http://lartc.org>
- [MySQL] MySQL 5.0 Reference Manual MySQL AB <http://dev.mysql.com/doc/refman/5.0/fr/index.html>
- [SecDebian] Manuel de sécurisation de Debian Alexander ReelsenJavier Fernández-Sanguino Peña Debian Project <http://www.debian.org/doc/manuals/securing-debian-howto/>

Articles

- [Gibson2002] Distributed Reflection Denial of Service Steve Gibson 22 février 2002 Gibson Research Corporation <http://www.grc.com/dos/drdo.htm>
- [Hain2007] IPv4 Address Pool Tony Hain Cisco Systems, Inc. <http://www.tndh.net/~tony/ietf/ipv4-pool-combined-view.pdf>
- [Netcraft] Web Server Survey Netcraft LTD. http://news.netcraft.com/archives/web_server_survey.html
- [Potaroo2007] IPv4 Address Report Geoff Huston <http://www.potaroo.net/tools/ipv4/index.html>
- [Schneier1999] Crypto-Gram : Open Source and Security Bruce Schneier September 15, 1999 Counterpane Internet Security, Inc. <http://www.counterpane.com/crypto-gram-9909.html>
- [SynDJB] SYN Cookies Dr. Daniel J Bernstein <http://cr.yo.to/syncookies.html>
- [SynWiki] SYN cookies <http://en.wikipedia.org/wiki/Syncookies/> Wikimedia Foundation, Inc.
- [Tomlinson] The First Network Email <http://openmap.bbn.com/~tomlinso/ray/firstemailframe.html> Ray Tomlinson

- [v6SrcRoute] IPv6 Routing Header Security Philippe Biondi Arnaud Ebalard http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf
- [Zalewski2001] Strange Attractors and TCP/IP Sequence Number Analysis Michal Zalewski Bindview Corporation <http://lcamtuf.coredump.cx/oldtcp/tcpseq.html>
- [Zalewski2002] Strange Attractors and TCP/IP Sequence Number Analysis - One Year Later Michal Zalewski Bindview Corporation <http://lcamtuf.coredump.cx/newtcp/>