Cahiers de l'Admin

Collection dirigée par Nat Makarévitch

Linux
Sécuriser un réseau

Bernard Boutherin
Benoit Delaunay

3e édition

EYROLLES

Cahiers de l'Admin

Linux

Sécuriser un réseau

3e édition

Chez le même éditeur

Admin'sys. Gérer son temps. – T. Limoncelli, adapté par S. Blondeel – N°11957, 2006, 274 pages.
Sécurité informatique. *Principes pour l'administrateur système* – L. Bloch, C. Wolfhugel - N°12021, 2007, 350 pages.
Mémento UNIX/Linux – I. Hurbain, avec la contribution d'E. Dreyfus - N°11954, 2006, 14 pages.
Debian. *Administration et configuration avancées* **–** M. Krafft, adapté par R. Hertzog et R. Mas, dir. N. Makarevitch – N°11904, 2006, 674 pages.
SSL VPN. – J. Steinberg, T. Speed, adapté par B. Sonntag. – N°11933, 2006, 220 pages.
Programmation Python. T. Ziade. – N°11677, 2006, 530 pages.

Collection « Cahiers de l'Admin »

Debian 2e édition
R. Hertzog, C. Le Bars, R. Mas.
N°11639, 2005, 310 pages.

BSD 2e édition
E. Dreyfus - N°11463, 2004, 302 pages.

Collection « Connectez-moi ! »

Partage et publication… Quel mode d'emploi pour ces nouveaux usages de l'Internet ?

Wikipédia. *Comprendre et participer.*
S. Blondeel. – *N°11941, 2006, 168 p.*

Les podcasts. *Écouter, s'abonner et créer.*
F. Dumesnil. – *N°11724, 2006, 168 p.*

Peer-to-peer. *Comprendre et utiliser.*
F. Le Fessant. – *N°11731, 2006, 168 p.*

Créer son blog en 5 minutes.
C. Bechet. – *N°11730, 2006, 132 p.*

Collection « Accès Libre »

Pour que l'informatique soit un outil, pas un ennemi !

La 3D libre avec Blender.
O. Saraja. – *N°11959, 2006, 370 pages. avec CD-Rom et cahier couleur.*

Débuter sous Linux avec Mandriva.
S. Blondeel, D. Cartron, J. Risi.
N°11689, 2006, 530 p. avec CD-Rom.

Ubuntu efficace.
L. Dricot et al.
N°12003, 2e édition 2007, 360 p. avec CD-Rom.

Gimp 2 efficace.
C. Gemy.
N°11666, 2005, 360 p. avec CD-Rom.

OpenOffice.org 2 efficace.
S. Gautier, C. Hardy, F. Labbe, M. Pinquier.
N°11638, 2006, 420 p. avec CD-Rom.

Réussir un site web d'association avec des outils libres !
A.-L. Quatravaux et D. Quatravaux.
N°12000, 2006, 348 p., à paraître.

Réussir un projet de site Web, *4e édition***.**
N. Chu.
N°11974, 2006, 230 pages.

Home cinéma et musique sur un PC Linux.
V. Fabre.
N°11402, 2004, 200 p.

Collection « Poches Accès Libre »

Mozilla Thunderbird.
Le mail sûr et sans spam.
D. Garance, A.-L. et D. Quatravaux.
N°11609, 2005, 320 p., avec CD-Rom.

Firefox. *Un navigateur web sûr et rapide.*
T. Trubacz, préface de T. Nitot.
N°11604, 2005, 250 p.

SPIP 1.8.
M.-M. Maudet, A.-L. et D. Quatravaux.
N°11605, 2005, 376 p.

Gimp 2.2.
Débuter en retouche photo et graphisme libre.
D. Robert.
N°11670, 2006, 296 p.

OpenOffice.org 2 Calc.
S. Gautier, avec la contribution de J.-M. Thomas.
N°11667, 2006, 220 p.

OpenOffice.org 2 Writer.
S. Gautier, avec la contribution de G. Veyssiere.
N°11668, 2005, 248 p.

Bernard **Boutherin**
Benoit **Delaunay**

Cahiers de l'Admin

Linux

Sécuriser un réseau

3e édition

Collection dirigée par Nat **Makarévitch**

EYROLLES

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

DANGER
LE PHOTOCOPILLAGE TUE LE LIVRE

Le code de la propriété intellectuelle du 1er juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée notamment dans les établissements d'enseignement, provoquant une baisse brutale des achats de livres, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.
En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans autorisation de l'éditeur ou du Centre Français d'Exploitation du Droit de Copie, 20, rue des Grands-Augustins, 75006 Paris.
© Groupe Eyrolles, 2003, 2004, 2007, ISBN : 2-212-11960-7, ISBN 13 : 978-2-212-11960-2

Dépôt légal : novembre 2006
N° d'éditeur : 7538
Imprimé en France

Avant-propos

Aujourd'hui, tout système d'information (ou presque) est connecté à Internet, ne serait-ce qu'indirectement, et de plus en plus souvent via un accès haut débit.

En entreprise comme chez le particulier, il abrite des données vitales et confidentielles. Il fait ainsi partie intégrante du système de production et sa compromission peut avoir des conséquences dramatiques (arrêt des traitements, paralysie des communications, perte voire détournement des informations…).

Comment se prémunir des destructions, espionnages, dénis de service et autres intrusions, possibles usurpations d'identité, tentatives visant à rendre le système non opérationnel ? Dans ce contexte, le système Linux peut jouer un rôle majeur pour la sécurité des réseaux et des systèmes connectés. La sûreté de son noyau, les nombreux outils réputés pour leur fiabilité, et pour la plupart directement intégrés dans ses distributions, conduisent de plus en plus d'entreprises à choisir Linux comme système d'exploitation pour les serveurs applicatifs.

À qui s'adresse ce livre ?

Cet ouvrage s'adresse aux administrateurs système et réseau qui veulent avoir une vision d'ensemble des problèmes de sécurité informatique et des solutions existantes, dans l'environnement Linux.

Il offre une marche à suivre aux adeptes de Linux ayant la charge d'un petit réseau informatique connecté à Internet, au sein d'une PME ou chez un particulier.

Plus largement, toute personne ayant des bases en informatique et souhaitant en apprendre davantage sur les pirates des réseaux et la façon de s'en protéger grâce à Linux tirera profit de cette lecture.

Nouveautés de la troisième édition

Cette troisième édition a été enrichie par de nombreux ajouts. Vous y découvrirez en particulier un nouveau chapitre et une annexe entièrement consacrés aux problèmes liés à l'authentification des utilisateurs. Sont traités dans cette partie les systèmes d'authentification centralisés, depuis les plus traditionnels comme la base NIS, jusqu'aux plus évolués qui font appel au protocole LDAP ou au système Kerberos. Le chapitre 10, « Gestion des comptes utilisateur et authentification », décrit les grands principes de fonctionnement et les caractéristiques de ces systèmes d'authentification, tandis que l'annexe B en donne un exemple concret de mise en œuvre.

Dans le chapitre 3, « Attaques et compromission de machines », un exemple de mise en œuvre du *Coroner toolkit* est présenté dans le but de compléter l'analyse forensique d'une machine compromise.

Le chapitre 6, « Sécurisation des services réseaux DNS, Web et mail », comprend quelques ajouts d'importance : moyens de détection des virus dans les courriers électroniques, méthodes de lutte contre les courriers non sollicités, ou *spam*, avec la mise en œuvre des listes grises (*greylists* en anglais), et la sécurisation d'un ensemble de services avec `stunnel`.

Enfin, les possibilités de marquage de paquets d'IPtables sont développées au chapitre 8, « Topologie, segmentation et DMZ », et un exemple de mise en place d'un écran captif utilisant cette technique est présenté. Ce même chapitre est enrichi par la description des principes et de la configuration d'un pare-feu transparent.

Structure de l'ouvrage

Les notes situées en marge, en éclairant certains points de détail, pourront constituer un deuxième fil conducteur pour la lecture.

La sécurisation et la protection d'un réseau d'entreprise demandent une excellente vue d'ensemble de l'architecture étudiée. Cette troisième édition du Cahier de l'Admin consacré à la sécurisation de systèmes et réseaux sous Linux, reprend la démarche méthodique que nous avions eue lors de la première édition. À travers une étude de cas générique mettant en scène un réseau d'entreprise, nous effectuerons un audit de sécurité pour aboutir à l'amélioration de l'architecture du réseau : filtrage des flux en entrée, sécurisation par chiffrement avec SSL et (Open)SSH, détection des intrusions, surveillance quotidienne…

L'étude de cas met en scène l'entreprise Tamalo.com, d'où sont issus les nombreux exemples pratiques qui illustrent notre propos.

Tout commence avec l'attaque d'une machine connectée au réseau, après laquelle la décision est prise de remodeler la structure informatique de la société. Un dispositif de protection adapté aux objectifs de sécurité de l'entreprise sera alors mis en place.

- Les **chapitres 1 à 3** présentent le contexte de l'étude de cas qui a favorisé ce piratage. On y décrit le développement formidable d'Internet, les problèmes de sécurité qui en découlent, et l'émergence de Linux comme système d'exploitation.

  Celui-ci, bien configuré, pourra servir de parade efficace à ces problèmes. La jeune société Tamalo.com a misé sur Linux pour son système informatique, mais un déploiement trop rapide, sans prise en compte des impératifs de sécurité, aboutit au piratage du réseau.

  L'analyse des machines compromises dévoile le scénario de l'intrusion et met en évidence l'exploitation de la faille (*exploit*) utilisée pour pénétrer les systèmes. Le *rootkit* utilisé par les pirates pour masquer leur présence est découvert.

Chapitre 1, « La sécurité et le système Linux »
Chapitre 2, « L'étude de cas : un réseau à sécuriser »
Chapitre 3, « Attaques et compromissions des machines »

- À partir du **chapitre 4**, la réplique se met en place. Les communications entre les machines sont sécurisées grâce aux techniques de chiffrement. Une section introduit le concept de réseau privé virtuel. Ces techniques qui protégent en particulier contre le *sniff*, ou écoute frauduleuse du réseau.

Chapitre 4, « Chiffrement des communications avec SSH et SSL »

- Les **chapitres 5 et 6** abordent la mise en sécurité des systèmes et des services (une section est notamment consacrée à la sécurité du serveur d'affichage X11). Celle-ci s'appuie sur deux principes simples : préférer des installations automatiques pour garantir l'homogénéité du parc, et opter pour une configuration minimale, sans services inutiles.

Chapitre 5, « Sécurisation des systèmes »
Chapitre 6, « Sécurisation des services réseau : DNS, Web et mail »

- Les services réseau qui subsistent, nécessairement ouverts à l'extérieur, sont alors configurés pour être le moins vulnérables possible.
- Grâce à l'utilisation de pare-feu reposant sur le couple IPtables/Netfilter, on déploie une protection réseau qui constituera le premier rempart contre les attaques extérieures (**chapitres 7 et 8**). La nouvelle topologie du réseau de Tamalo.com fait alors apparaître une zone démilitarisée, DMZ, ouverte à l'extérieur. Cette discussion sur la protection réseau inclut une réflexion sur la sécurité de la technologie Wi-Fi utilisée pour la réalisation d'un réseau sans fil ; elle présente notamment les risques qu'encourent leurs usagers et les solutions de sécurité existantes pour rendre cette technologie plus sûre.

Chapitre 7, « Filtrage en entrée de site »
Chapitre 8, « Topologie, segmentation et DMZ »

- Pour prévoir les cas où une machine de Tamalo.com, restée vulnérable, serait attaquée, voire compromise, on se dote de l'indispensable panoplie d'outils d'audit système et de surveillance : métrologie, prise d'empreintes, détection d'intrusions. Des techniques de leurre, les pots

de miel, permettront d'observer et d'analyser le comportement des pirates lors d'une compromission, et de les détourner des serveurs de production.

Chapitre 9, « Surveillance et audit »

- Tous ces outils, décrits au **chapitre 9**, permettent de réagir au plus vite lors d'une attaque. Les données qu'ils produiront seront ensuite analysées pour servir à la réalisation des tableaux de bord, véritables baromètres du réseau informatique, destinés en général aux instances dirigeantes de l'entreprise.

Chapitre 10, « Gestion des comptes utilisateur et authentification »

- Enfin, le **chapitre 10** expliquera comment fonctionnent trois grands systèmes centralisés d'identification et d'authentification des utilisateurs : la base NIS, le protocole LDAP et le système Kerberos.
- L'**annexe A** concernant les infrastructures à gestion de clés (IGC ou PKI en anglais) vient compléter la partie du chapitre 4 concernant les certificats X.509.
- Enfin, l'**annexe B** met en œuvre les trois grands systèmes centralisés d'identification et d'authentification des utilisateurs présentés au chapitre 10.

Remerciements

Nous adressons nos vifs remerciements à tous ceux qui ont permis que cet ouvrage voie le jour, et en particulier à notre éditrice Muriel Shan Sei Fan des éditions Eyrolles, qui nous a soutenus tout au long de notre travail de rédaction, ainsi qu'à Nat Makarévitch qui a bien voulu relire ce livre et y apporter sa pertinente contribution.

Table des matières

1. La sécurité et le système Linux 1
Enjeux et objectifs de sécurité 2
La menace 2
Principaux facteurs de motivation des pirates 3
Risques liés au type de connexion 3
Risques liés aux failles des systèmes 4
Émergence des systèmes Linux 4
Linux et la sécurité 5
Des distributions Linux sécurisées 5
En résumé... 6

2. L'étude de cas : un réseau à sécuriser 9
Une jeune entreprise 10
Les besoins de la société en termes de services 10
Les choix techniques initiaux de Tamalo.com 11
Web et services associés 12
Transfert de fichiers 12
Base de données 12
Résolution de noms 12
Messagerie électronique 13
Partage de fichiers 13
Impression réseau 13
L'infrastructure informatique vieillissante et vulnérable 13
La compromission du site 14
Mise en évidence des vulnérabilités 15
La refonte du système informatique 15
Le projet d'une nouvelle infrastructure réseau 16
Études des flux réseau 18
Vers des outils de communication sécurisés 18
Un suivi et une gestion quotidienne du système d'information 20
En résumé... 20

3. Attaques et compromissions des machines 23
Kiddies, warez et rebonds 24
Scénario de l'attaque du réseau de Tamalo.com 26
Une faille dans le système 26
L'exploitation de la faille (« exploit ») 26
Utilité des scans réseau 26
La compromission 27
Analyse de la machine compromise 28
Traces visibles sur le système avant réinitialisation 28
Sauvegarde du système compromis 29
Analyse fine de l'image du disque piraté 29
Montage pour l'analyse 29
Étude des fichiers de démarrage et configuration 30
Étude des fichiers créés lors du piratage 30
Analyse avec The Coroner toolkit 30
Trousse à outils du pirate : le rootkit t0rn 33
Sniffer réseau d'un rootkit 33
Le mode promiscuous 35
Rootkit : effacer les traces et masquer la présence du pirate 37
Rootkit : la porte dérobée (backdoor) 38
Rootkit t0rn : conclusion 38
Détecter la compromission à partir des logs 39
Origine de l'attaque 40
En résumé… 42

4. Chiffrement des communications avec SSH et SSL 45
Les quatre objectifs du chiffrement 46
Authentification 46
Intégrité 46
Confidentialité 47
Signature électronique 47
Facteurs de fiabilité des techniques de chiffrement 47
Algorithmes de chiffrement symétrique et asymétrique 48
Chiffrement symétrique 48
Chiffrement asymétrique 49
Le protocole SSL (Secure Socket Layer) 51
Qu'est ce que SSL ? 51
SSL, comment ça marche ? 51
Les certificats X.509 52

Authentification et établissement de la connexion SSL 53
Utilisation de SSL par les applications client/serveur 54
Le protocole SSH (Secure Shell) 54
Qu'est-ce que SSH ? 54
À quels besoins répond SSH ? 54
Caractéristiques d'OpenSSH 56
Installation d'OpenSSH 57
Fichiers de configuration d'OpenSSH 58
Activation et lancement du serveur SSH 58
Désactivation et arrêt du serveur SSH 59
Utilisation de SSH 59
Connexion interactive 59
Exécution de commandes à distance 59
Copie distante de fichiers ou de répertoires 60
Transfert interactif de fichiers 60
Options des commandes SSH 60
Authentification avec SSH 60
Configuration du service SSH 60
Authentification par mot de passe 61
Authentification à clé publique 61
Relais d'affichage X11 64
Gestion des accès au service SSH 65
Dépannage 65
L'alternative VPN 66
En résumé… 67

5. SÉCURISATION DES SYSTÈMES 69
Installation automatisée 70
Mise à jour régulière des systèmes 73
Mise à jour et installation optimale avec APT 74
Mise à jour avec Red Hat Network 74
L'indispensable protection par mot de passe au démarrage 74
Mise en configuration minimale, limitation des services actifs 75
Identification des processus 76
Identification des ports réseau utilisés 76
Identification des services actifs 77
Désactivation des services inutiles 78
Sécurisation du système de fichiers 79
Permissions des fichiers 79
Détection des fichiers dotés de droits trop permissifs 80
Droits suid et sgid 80
Alternative à la protection suid : sudo 81
Options de montage des systèmes de fichiers 82
Gestion des accès et stratégie locale de sécurité 82
Compte privilégié root 82
Blocage des comptes inutiles 83
Filtrage réseau avec TCP Wrapper 83
Configuration des services système cron et syslog 84
cron 84
syslog 84
Configuration sécurisée de la pile TCP/IP 85
Ignorer certains messages ICMP 85
ICMP Redirect 85
ICMP Echo request 87
ICMP Ignore Bogus Response 87
Interdiction du source routing 87
Surveillance des martiens ! 88
Protection contre les attaques IP spoofing et SYN flooding 88
Configuration en pare-feu avec IPtables 89
Extension du noyau 89
Serveur d'affichage X11 et postes de travail 89
En résumé… 90

6. SÉCURISATION DES SERVICES RÉSEAU : DNS, WEB ET MAIL 93
Bases de la sécurisation des services réseau 94
Service de résolution de noms DNS 95
Comment ça marche ? 96
Serveurs de noms et sécurité 97
Installation du logiciel BIND 97
Configuration des serveurs DNS 98
Compte non privilégié 98
Changement de la racine du système de fichiers avec « chroot » 98
Activation et lancement du serveur 103
Configuration des clients DNS 104
Messagerie électronique 104
Comment ça marche ? 104
Les logiciels de transfert de courrier 105
Messagerie électronique et sécurité 106
Spam et relais ouvert 106
L'architecture du système de messagerie 107
Installation de sendmail 109
Activation de sendmail 109
Configuration de sendmail 110
Sendmail et Milter 115
Configuration antivirus et antispam à Tamalo.com 116
Lutte antivirus : Sendmail, Milter et ClamAV 117
Lutte antispam : Sendmail, milter et milter-greylist. 121
Installation d'IMAP 124
Configuration et activation du serveur IMAPS 124
Serveur Web 125
Serveur Web et sécurité 125
Installation de HTTPD 125
Configuration et activation de HTTPD 126
Sécurisation des accès nomades à la messagerie avec stunnel 127
Configuration du serveur stunnel accessible depuis l'extérieur 127
Authentification du serveur 127
Authentification des utilisateurs 128
Configuration de stunnel sur le serveur 129
Configuration d'un client nomade supportant SSL et l'authentification par certificat 132
Configuration d'un client nomade ne supportant pas SSL ou l'authentification par certificat 134
En résumé… 135

7. Filtrage en entrée de site 137
But poursuivi 138
Principes de base du filtrage en entrée de site 138
Filtrage sans état 139
Adresses IP source et destination 139
Protocole, ports source et destination 139
Drapeaux TCP et filtrage en entrée 140
Les limites du filtrage sans état 142
Filtrage avec états 143
Politique de filtrage : avant la compromission, « tout ouvert sauf » 144
Politique de filtrage : du « tout ouvert sauf » au « tout fermé sauf » 145
Déploiement de service FTP avec (et malgré) les filtres 146
Filtrage d'un client FTP actif 147
Filtrage d'un serveur FTP destiné à fonctionner en mode actif 150
Filtrage d'un client FTP passif 150
Filtrage du serveur FTP passif, limitation du serveur à une plage de ports 150
En résumé… 151

8. Topologie, segmentation et DMZ 153
Pourquoi cloisonner ? 154
Définition des zones du réseau de Tamalo.com 155
Définition des flux à l'extérieur et à l'intérieur du réseau de Tamalo.com 155
Postes de travail 155
Serveurs applicatifs internes 155
Serveurs accessibles depuis l'extérieur et l'intérieur : DMZ 155
Topologie du réseau 156
Topologie à un seul pare-feu 156
Topologie à double pare-feu adoptée pour le réseau de Tamalo.com 157
Détails de la configuration réseau de Tamalo.com 158
DMZ 158
Services internes 160
Postes de travail 160
Comment segmenter ? Les VLAN et leurs limites 160
VLAN par port physique 160
VLAN par adresse MAC 161
Configuration VLAN retenue pour Tamalo.com 162
Proxy et NAT 163
Proxy 163
Traduction d'adresses NAT 165
Source NAT – un pour un – ou NAT statique 166
Source NAT -N pour M – ou NAT dynamique 168
Proxy versus NAT 171
Netfilter/IPtables 171
Fonctionnalités d'IPtables 171
Tables et chaînes 171
Écriture des règles 173
Suivi de connexion 173
Journalisation 173
Traduction d'adresses – NAT 174
Filtrage 174
Configuration IPtables des deux pare-feu Linux 175
Configuration IPtables de chaque poste de travail 177
Configuration IPtables du serveur SMTP 178
Marquage de paquets avec IPtables 178
Modification des champs TOS, TTL 178
Marquage simple du paquet 179
Pare-feu transparent, mode bridge 180
Positionnement du pare-feu transparent 180
Adressage IP 180
Proxy ARP 181
Configuration pratique du pare-feu transparent 182
Configuration en proxy ARP coté DMZ 182
Configuration en proxy ARP coté interne 182
Configuration des interfaces et mise en place des routes 182
Configuration IPtables 183
Sécurité du réseau sans fil 183
Risque d'accès frauduleux au réseau 183
Le protocole 802.1X 184
Risque d'écoute du réseau 185
En résumé… 186

9. Surveillance et audit 189
Des traces partout 190
Linux et le syslog 190
Empreinte des machines : Tripwire 192
Métrologie réseau avec MRTG 193
Installation et configuration de MRTG chez Tamalo.com 195
Configuration SNMP du firewall A pour accepter les requêtes MRTG 195
Installation et configuration de MRTG sur la machine d'analyse 196
NMAP 197
Audit réseau avec Nessus 197
Configuration de Nessus 198
Rapport d'audit 200
Détection d'intrusion : Snort 201
Mise en place de la sonde Snort 201
Configuration et validation de Snort, détection des scans 201
Le pot de miel 203
Tableau de bord de la sécurité 204
Les indicateurs de sécurité 204
Synthèses des indicateurs dans un tableau de bord 206
En résumé… 206

Scénario de l'attaque du réseau de Tamalo.com

Une faille dans le système

À l'heure actuelle, les compromissions de machines exploitent pour la plupart une faille du système ou une erreur dans la configuration de ce dernier. La faille la plus classique utilisée pour prendre le contrôle d'un service est le débordement de mémoire-tampon ou *buffer overflow*, en anglais.

B.A.-BA Débordement de mémoire et exécution de code arbitraire

En voici le principe : un service, par exemple wu-ftpd sous Linux, reçoit des arguments depuis l'application cliente. Ces arguments transitent par une zone de mémoire allouée par le service.
Si le service ne vérifie pas correctement les arguments qu'il reçoit (taille…), un débordement de mémoire est possible.
Ainsi, l'application cliente, parfois modifiée par un cracker, envoie au serveur un argument beaucoup trop long. Une partie de cet argument écrase donc le contenu de la mémoire qui suit immédiatement la zone allouée par le service. À cet endroit étaient stockées des instructions qui vont être prochainement exécutées par le processus serveur.
En écrasant ces instructions, le pirate va donc pouvoir faire exécuter le code de son choix à la machine, sous l'identité du service !
Si ce dernier tourne en mode privilégié, il est probable que le client aura bientôt un accès sur le compte administrateur `root` de la machine…

L'exploitation de la faille (« exploit »)

La publication d'une nouvelle faille entraîne une course contre la montre entre les crackers et les développeurs des systèmes. Les premiers travaillent à l'écriture d'un « exploit » : il s'agit d'un programme permettant d'exploiter la faille, c'est-à-dire de prendre le contrôle de toute machine employant la version vulnérable. Les autres travaillent à l'écriture d'une nouvelle version ou d'un correctif, nommé *patch* en anglais.

À ce jeu-là, les pirates ont malheureusement toujours l'avantage, car même si dans la plupart des cas le patch arrive avant l'exploit, celui-ci n'est jamais déployé à temps sur l'ensemble du parc immense que constitue Internet.

Le cracker met son exploit à la disposition de la communauté des pirates par l'intermédiaire d'un certain nombre de sites Internet connus dans ce milieu. De leur côté, les développeurs diffusent le patch de sécurité.

Utilité des scans réseau

Si vous avez un jour la curiosité d'analyser le trafic à la frontière de votre réseau, vous aurez la surprise de voir que les scans y sont permanents. Cela signifie qu'il y a toujours quelqu'un, quelque part, en train d'analyser votre parc informatique pour voir si une machine de votre réseau n'offre pas, par hasard, une faille connue.

Si vous êtes en charge de ce réseau, ne négligez pas toute cette surveillance. En effet, une compromission est le plus souvent précédée d'un scan. C'est l'élément qui permet au pirate d'avoir la liste des machines et des services de votre réseau qui présentent une vulnérabilité.

Par exemple, un attaquant pourra scanner le port 80 de l'ensemble de votre réseau afin d'établir la liste des serveurs web accessibles depuis l'extérieur.
Le scanner n'est pas un outil réservé aux pirates. Un administrateur réseau se doit d'en posséder un pour établir la liste des services offerts sur son réseau. Le scanner le plus réputé disponible sous Linux s'appelle Nmap ; il est téléchargeable à partir de l'URL suivante :

- http://www.insecure.org/nmap/nmap_download.html

OUTILS Scanner réseau

Un scanner est un programme qui balaye une plage de ports sur un ensemble de machines, afin d'établir la liste des couples machine/service ouverts.

- Le scan horizontal consiste à scanner un port sur un ensemble de machines.
- Le scan vertical consiste à scanner une plage de ports sur une même machine.

La compromission

Les pirates ont le plus souvent une connaissance minimale du site choisi pour cible. Ils organisent une attaque pendant une période où la vigilance de l'administrateur est relâchée. Ainsi, les attaques ont souvent lieu la nuit, le week-end, les jours fériés ou pendant les périodes de vacances.

C'est souvent une accumulation de détails plus ou moins anodins qui laisse suspecter la compromission d'une machine.

C'est ainsi qu'au retour des vacances de Noël, un développeur de Tamalo.com se plaint du comportement anormal de la commande `ps` : elle rend un `segmentation fault` en lieu et place de la liste des processus attendue.

Quelques commandes permettent rapidement de cerner le problème :

1 Déterminer où se trouve le programme `ps`.

```
which ps
/bin/ps
```

2 Examiner la date de dernière modification puis de création du fichier.

```
ls -l /bin/ps
ls -lc /bin/ps
```

1 Examiner la nature du fichier `ps`...

```
file /bin/ps
ELF 32-bits LSB executable, blabla.., not stripped ?????
```

L'attribut `not stripped` est étrange pour une commande système. En général, les commandes système sont déployées pour économiser de l'espace disque et la table des symboles n'y figure pas. Cet attribut ne devrait donc pas apparaître, ce qui laisse redouter que le programme ne provient pas de la distribution.

De plus, si le résultat de `ls -l /bin/ps` paraît acceptable, `ls -lc /bin/ps` révèle que le fichier a été créé le 26 décembre dernier ; un administrateur aurait-il installé un nouveau `ps` pendant les vacances ?

Par ailleurs, les utilisateurs de la machine concernée se plaignent de ralentissements et d'accès disque permanents, ce que ne confirme pas la commande `ps`… sauf si cette commande a été modifiée par un pirate afin de dissimuler son activité.

Une seule conclusion s'impose : il est urgent de débrancher la machine du réseau afin de faire des vérifications plus approfondies. Ce n'est pas sans conséquence pour Tamalo.com car ce serveur contient le référentiel CVS qui gère les versions de logiciels.

Le temps de ressortir les CD-Rom de la distribution Red Hat, de recompiler de manière statique quelques commandes originales (`ps`, `ls`, `netstat`) et il va être possible de mesurer l'ampleur des dégâts.

ATTENTION

Dans certains cas, le programme au comportement anormal qu'on soupçonne d'avoir été installé par les pirates (ici `ps`) peut contenir une bombe qui se déclenchera après un certain nombre d'invocations. L'existence même d'un doute nous contraint à douter de l'ensemble des programmes installés. On comprendra que la première chose à faire est de réamorcer sur un système sûr et de sauvegarder afin de préserver les données des utilisateurs mais aussi les programmes du système pour une analyse ultérieure. Il faut donc invoquer le moins possible de programmes sur la machine potentiellement compromise.

B.A.-BA En cas de compromission…

Il est impossible d'affirmer que l'analyse d'une machine compromise permettra de détecter toutes les modifications qu'elle a pu subir. C'est pourquoi la compromission d'un système doit conduire à sa réinstallation complète. Le système de fichiers doit être entièrement reformaté. Cette règle est essentielle pour garantir le retour à un système d'exploitation sain après piratage.

C'était le premier contact pratique de Tamalo.com avec un piratage informatique. Dans ce qui suit, nous allons analyser plus en détail les traces laissées par les pirates au cours de cette compromission afin de mieux connaître les outils utilisés et les failles de nos systèmes qui ont été exploitées.

Analyse de la machine compromise

Traces visibles sur le système avant réinitialisation

En cas de compromission d'une machine, il est utile de récupérer quelques indicateurs de l'état du système d'exploitation avant tout redémarrage. En effet, les pirates disposent parfois d'outils permettant d'effacer leurs traces après un redémarrage.

Il faut en premier lieu s'assurer que les commandes utilisées sont saines. Certaines commandes peuvent avoir été modifiées par les outils dont dispose le pirate, par exemple afin de masquer sa présence. Ces modifications ont pu être faites à différents niveaux :

- Modification des commandes : `ps`, `ls`, `netstat`, `find`, `du`, `passwd`, etc. Dans ce cas, la parade consiste simplement à recopier la commande d'origine sur le système.
- Modification des bibliothèques dynamiques : les fonctions incluses dans les bibliothèques dynamiques sont utilisées lors de l'exécution d'une commande (la commande `ldd /bin/ps` fournira la liste des bibliothèques utilisées par la commande `/bin/ps`). Pour y remédier, il suffit de compiler (bien entendu pas sur la machine compromise) les commandes en mode statique afin qu'elles n'en dépendent plus. Pour cela, utilisez l'option de compilation `-static` de gcc. L'exécutable résultant contiendra le code de toutes les fonctions utilisées, il ne chargera aucune bibliothèque au moment de son exécution.
- Modification des modules du noyau : si les modules du noyau sont modifiés, on considèrera que l'analyse à chaud ne peut pas apporter de résultat fiable et on passera directement, après sauvegarde, à la réinstallation complète.

Dans notre exemple, l'analyse à chaud dévoile quelques anomalies qui seront confirmées par la suite.

La commande `netstat -tupan` sur le système fait apparaître un service en écoute sur le port 15 000, invisible auparavant.

La commande `ps` modifiée nous cachait quelques processus, dont le programme : `/usr/sbin/nscd`, qui est ici une *backdoor* SSH en écoute sur le port 15 000. Grâce à cette porte dérobée, le pirate pouvait revenir se connecter

Investigation Forensique

Forensique, en anglais *forensic*, est synonyme de criminalistique. Une investigation forensique a pour objectif de prouver l'existence d'un crime et de déterminer l'identité de l'auteur ainsi que son mode opératoire.

L'analyse forensique d'une machine compromise est donc du ressort des autorités de police judiciaire plutôt que le travail courant d'un administrateur système ! Pour autant, il n'est pas inutile de connaître et d'avoir mis en œuvre les outils et méthodes d'une telle analyse, et ce pour deux raisons. D'une part, en cas de compromission grave, cela permettra d'éviter les fausses manipulations qui risqueraient d'avoir pour conséquence l'effacement de certaines traces du pirate… des traces qui seront indispensables si un dépôt de plainte est envisagé ! D'autre part, la connaissance du mode opératoire des pirates permet d'être mieux armés pour se protéger contre de futures attaques.

OUTILS « rescue » et « live CD »

Il est nécessaire de réamorcer le système afin de bénéficier, lors de l'exploration du contexte, d'un environnement logiciel de provenance sûre, autonome donc non dépendant des éléments installés et non exposé aux modifications intempestives de programmes déployés par le pirate. La disquette « rescue » produite durant l'installation exploite d'ordinaire les programmes placés sur le disque dur. On lui préférera donc les distributions de Linux disponibles sous forme « live CD », c'est-à-dire utilisables grâce à leur seul CD-Rom et sans installation. Certaines furent conçues et réalisées en fonction de cet objectif, d'autres sont si riches qu'elles intègrent cela.

sur notre machine de façon discrète. Les connexions du pirate sont chiffrées. Elles ne sont pas journalisées par le système ; le pirate n'est pas détectable par les commandes `who` ou `w`.

Enfin, la commande `ifconfig` indique que l'interface réseau est en mode `PROMISCUOUS`, ce qui laisse penser qu'un *sniffer* réseau a été installé sur la machine.

ATTENTION nscd

`nscd` est aussi le nom d'un démon tout à fait honorable (le Name Server Cache Daemon). La présence de `nscd` n'implique pas que la machine est piratée ! Notons que le risque de confusion est délibéré de la part du pirate.

Sauvegarde du système compromis

Chaque partition est sauvée sur un autre système à l'aide des commandes `dd` pour le dump et `nc` (netcat) pour le transfert réseau :

```
machine-saine> nc -l -p 10101 > fich-hda1
machine-compromise> dd if=/dev/hda1 | nc machine-saine 10101
```

`nc` écoute sur le port 10101

BON SENS Choix des noms de fichier

Dans le cas où plusieurs machines sont compromises, faites apparaître le nom de la machine compromise dans le nom du fichier : par exemple `tamalo1-hda1` plutôt que `fich-hda1`.

Analyse fine de l'image du disque piraté

L'analyse à froid du système sera faite en poursuivant différents objectifs :

1. Déterminer la date précise de la compromission initiale. La connaissance de celle-ci permettra un certain nombre de corrélations avec les fichiers de journalisation du routeur d'entrée et des machines du réseau.
2. Déterminer la faille exploitée pour prendre le contrôle du système. Il sera alors possible de mettre à jour le service correspondant.
3. Déterminer la nature des outils installés par le pirate et en identifier les fichiers de traces et les programmes afin de rechercher sur d'autres machines du site des signes éventuels de compromission.
4. Connaître la source de l'attaque afin de la contacter pour avoir des explications (attention : il est très probable que la machine attaquante soit elle-même sous le contrôle du pirate).
5. Déterminer jusqu'à quel point l'intrusion a réussi, savoir si les mots de passe du réseau ont pu être compromis.

OUTILS Application client/serveur avec netcat

Le logiciel netcat, fourni en standard avec Linux, permet de réaliser très simplement une application client/serveur.

Le serveur est lancé sur la machine `host1.tamalo.com` pour écouter sur le port 99999 avec la commande :

```
nc -l -p 99999
```

Le client est lancé sur la machine `host2.tamalo.com` pour se connecter sur `host1.tamalo.com:99999` avec la commande :

```
nc host1.tamalo.com 99999
```

Tout message envoyé sur l'entrée standard `<sdtin>` du client `nc` qui s'exécute sur `host1` – c'est-à-dire saisi au clavier de `host1` – apparaîtra sur la sortie standard `<stdout>` du serveur `nc` qui tourne sur `host2` – c'est-à-dire à l'écran de `host2`.

Cette commande est très utile pour analyser le fonctionnement de certaines applications serveurs, comme on le verra au chapitre 7 pour l'analyse du fonctionnement de FTP actif.

▸ http://netcat.sourceforge.net

Montage pour l'analyse

Pour l'analyse, il reste à monter (en loopback) le système de fichiers concerné.

```
mount -o loop,ro,noexec,nodev fich-hda1 /root/
host_compromis_hda1
```

Il peut être utile d'utiliser l'option `ro` (read only) pour ne pas altérer les traces sur le système compromis. De plus, pour éviter d'exécuter par erreur des commandes sur la machine compromise, on utilisera l'option `noexec`. Enfin, on pourra ajouter l'option `nodev` pour ignorer les fichiers de type `device`, qui sont des points d'entrée vers les périphériques.

Étude des fichiers de démarrage et configuration

Pour déceler les traces sur l'image du disque d'une machine piratée, le plus simple est de commencer par l'étude des fichiers de démarrage, très souvent modifiés par les pirates, afin de :

- masquer certaines traces en cas de redémarrage de la machine ;
- relancer un certain nombre de processus : backdoor, scanner, sniffer, à chaque redémarrage du système.

Sous Linux, il faut s'intéresser aux fichiers et aux répertoires suivants :

- `/etc/inittab`
- `/etc/init.d/`
- `/etc/rc.sysinit`
- `/etc/sysconfig/`
- `/etc/rc.d/`
- `/etc/inetd.conf` ou `/etc/xinetd.conf` et `/etc/xinetd.d/`
- `/etc/crontab`
- `/etc/cron.daily`, `/etc/cron.hourly...`

Étude des fichiers créés lors du piratage

Bon sens

Aucune démarche de recherche n'est éternellement valide car les attaquants disposent d'outils de plus en plus évolués.

Il convient aussi de rechercher les fichiers créés le jour du piratage, à l'aide d'une simple commande `find`. Il faut préférer une recherche basée sur la date de création du fichier `CTIME`, qui n'est modifiée que par le noyau, plutôt que sur la date de modification `MTIME`. En effet, la date de modification peut-être altérée facilement par le pirate à l'aide de la commande `touch`.

Analyse avec The Coroner toolkit

The Coroner Toolkit, ou TCT est une panoplie d'outils forensiques destinés à l'analyse d'une machine compromise. TCT fournit des outils très performants pour analyser une machine compromise. Dans ce qui suit, nous allons l'utiliser pour affiner notre analyse et retrouver certaines traces moins évidentes laissées par le pirate.

TCT appuie sa démarche de recherche sur le recoupement des événements temporels. Pour cela, il introduit la notion de *MAC time*, MAC étant l'acronyme de Modification Access Creation. En effet, la connaissance de ces trois attributs d'un fichier peut fournir des informations décisives sur l'acti-

vité du pirate. Par rapport à une simple commande `find`, TCT ajoute la détermination de l'*access time* qui est impossible en passant par les appels système standards. Pour que cette détermination soit possible, le système de fichiers doit avoir été sauvegardé par une copie des partitions, comme le permet `dd`, et non par une commande d'archivage ou de copie de fichiers qui altérerait l'*access time*.

La commande `grave-robber` capture les informations utiles dans l'image du système de fichiers et renseigne la base de données de TCT. Notez que cette même commande capture également les informations concernant les processus et les connexions réseau actives sur un système vivant.

```
grave-robber -c /host_compromis_hda1 -o LINUX2 -m -i
Le fichier /root/tct-1.15/data/tamalo1_01_23_17\:36\:37+0100/
body contient la base de données de TCT.
```

Il est possible de compléter les informations fournies par `grave-robber` avec des informations sur les fichiers effacés, grâce à la commande `fls` issue de la boîte à outils *sleuthkit*. Les formats étant compatibles, il suffit de rediriger la sortie de `fls` pour compléter la base de données de TCT comme indiqué ci-dessous :

```
fls -f linux-ext2 -r -m /host_compromis_hda1 fich-hda1 >> /
root/tct-1.15/data/tamalo1_01_23_17\:36\:37+0100/body
```

La commande `mactime` ci-dessous génère, à partir de la base de données de TCT, la liste chronologique des modifications du système de fichiers.

```
mactime -p /host_compromis_hda1/etc/passwd -g /
host_compromis_hda1/etc/group 1/1/1971 > mactime-tamalo1.out
-p indique le chemin du fichier passwd utilisé pour résoudre les
noms d'utilisateurs
-g indique le chemin du fichier de groupes
Sont considérés tous les événements postérieurs au 1/1/1971, on
n'en rejette donc aucun.
```

La figure 3-2 montre les répercussions sur le système de fichiers de l'activité du pirate pendant la configuration du rootkit t0rn (voir section suivante). On voit assez clairement le déroulement des opérations, qui commence par une lecture attentive de la documentation de SSHD (nul n'est parfait) ! Notez la présence des attributs `m`, `a`, `c`, ainsi que des références à des fichiers effacés marqués `(deleted)`. Dans ce cas précis, les références à ces fichiers ne sont pas d'une grande utilité car le pirate ne s'est pas donné la peine d'effacer ses programmes sources.

Dans certains cas au contraire, on pourra être très motivé pour récupérer un fichier effacé afin d'en analyser le fonctionnement, s'il s'agit par exemple du code source d'un exploit.

Référence The Coroner Toolkit

Les outils du Coroner toolkit sont disponibles à l'adresse http://www.porcupine.org/forensics/
Pour ceux qui veulent en savoir plus sur l'analyse forensique, le livre (en anglais) de Dan Farmer et Wietse Venema, *Forensic Discovery*, est disponible en libre téléchargement sur ce même site.

```
root@rh71: /root
File  Edit  Settings  Help
Dec 27 01 16:09:14   13609 .a. -rw-r--r-- root 15       /host_compromis_hda1/usr/man/cs/man8/sshd.8.gz
Dec 27 01 16:54:04    7578 .a. -rwxr-xr-x 1133 100  /host_compromis_hda1/usr/src/.puta/t0rnp
                      1345 .a. -rwxr-xr-x 1133 100  /host_compromis_hda1/usr/src/.puta/t0rnsb
Dec 27 01 16:54:18     524 ..c -rwxr-xr-x root root /host_compromis_hda1/usr/info/.t0rn/shhk
                        31 m.c -rw-r--r-- root root /host_compromis_hda1/usr/src/.puta/.1addr
                     53364 ..c -rwxr-xr-x 1133 100  /host_compromis_hda1/bin/netstat
                        28 m.c -rw-r--r-- root root /host_compromis_hda1/etc/ttyhash
                      1024 m.c drwxr-xr-x root root /host_compromis_hda1/usr/src
                     22460 ..c -rwxr-xr-x 1133 100  /host_compromis_hda1/usr/bin/du
                    201552 ..c -rwxr-xr-x root root /host_compromis_hda1/usr/sbin/nscd
                    201552 ..c -/-rwxr-xr-x root root /host_compromis_hda1/usr/info/.t0rn/sharsed (deleted-real
                       328 .ac -rwxr-xr-x root root /host_compromis_hda1/usr/info/.t0rn/shhk.pub
                     13726 m.c -rwxr-xr-x root root /host_compromis_hda1/etc/rc.d/rc.sysinit
                      6408 ..c -rwxr-xr-x 1133 100  /host_compromis_hda1/usr/sbin/in.fingerd
                      3072 m.c drwxr-xr-x root root /host_compromis_hda1/sbin
                        21 mac -rw-r--r-- root root /host_compromis_hda1/usr/src/.puta/.1logz
                     20452 ..c -rwxr-xr-x root root /host_compromis_hda1/sbin/xlogin
                      1024 m.c drwxr-xr-x root root /host_compromis_hda1/usr/info/.t0rn
                     20452 m.c -r-sr-xr-x 1133 100  /host_compromis_hda1/bin/login
                      9216 m.c drwxr-xr-x root root /host_compromis_hda1/usr/info
                        62 m.c -rw-r--r-- root root /host_compromis_hda1/usr/src/.puta/.1proc
                       499 m.c -rw-r--r-- root root /host_compromis_hda1/usr/info/.t0rn/shdcf
                     32728 ..c -rwxr-xr-x 1133 100  /host_compromis_hda1/sbin/ifconfig
                      3072 m.c drwxr-xr-x root root /host_compromis_hda1/usr/sbin
                     43024 .a. -rwxr-xr-x root root /host_compromis_hda1/lib/security/.config/bin/ls
                        95 m.c -rw-r--r-- root root /host_compromis_hda1/usr/src/.puta/.1file
Dec 27 01 16:54:19   39484 ..c -rwxr-xr-x 1133 100  /host_compromis_hda1/bin/ls
                         9 .a. -/lrwxrwxrwx root root /host_compromis_hda1/usr/bin/awk -> /bin/gawk
                         5 m.c -rw-r--r-- root root /host_compromis_hda1/tmp/info.tmp
                       347 .a. -rw-r--r-- root root /host_compromis_hda1/etc/hosts.deny
--More--(88%)
```

Figure 3–2
Fichiers lus et modifiés par le pirate au moment de la compromission

Sachez que TCT fournit la méthode et les outils nécessaires à une telle récupération. Pour cela, on s'appuiera sur le fait que le système d'exploitation incrémente linéairement les *inodes* des fichiers créés dans un même répertoire comme le montre la figure 3-3.

```
root@rh71: /root/TOOLS-PX1150
File  Edit  Settings  Help
151678 host_compromis_hda1/dev/ptyi/bscan/scan
151679 host_compromis_hda1/dev/ptyi/bscan/r00t
151680 host_compromis_hda1/dev/ptyi/bscan/scan.c
151681 host_compromis_hda1/dev/ptyi/bscan/try
151682 host_compromis_hda1/dev/ptyi/bscan/xlist
151683 host_compromis_hda1/dev/ptyi/bscan/core
151685 host_compromis_hda1/dev/ptyi/genoXyZ.TgZ
151686 host_compromis_hda1/dev/ptyi/koglione.tar.gz
151687 host_compromis_hda1/dev/ptyi/wget
151688 host_compromis_hda1/dev/ptyi/whp.tgz
151689 host_compromis_hda1/dev/ptyi/sc.tgz
151690 host_compromis_hda1/dev/ptyi/sm/juno
--More--(37%)

root@rh71: /root
File  Edit  Settings  Help
[root@rh71 /root]# icat host compromis hda1  151680 | more
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>

#define MAX_SOCKETS 1000
#define TIMEOUT 20

#define S_NONE       0
#define S_CONNECTING 1

struct conn_t {
  int s;
  char status;
  time_t a;
  struct sockaddr_in addr;
};
struct conn_t connlist[MAX_SOCKETS];

void init_sockets(void);
void check_sockets(void);
```

Figure 3–3
Recherche d'un fichier et visualisation à partir de son inode

Ainsi pour récupérer le code source d'un exploit, on recherchera des fichiers, toujours présents sur le disque, créés immédiatement avant et après le fichier perdu. Un fichier effacé sera caractérisé par un trou dans la séquence des inodes du répertoire. Si les blocs occupés par le fichier manquant n'ont pas été réaffectés, il sera possible de le visualiser. La figure 3-3 montre comment voir le contenu du fichier `scan.c` dont l'inode est 151 680, à l'aide de la commande `icat` fournie par TCT.

Trousse à outils du pirate : le rootkit t0rn

Un rootkit est défini par l'Agence nationale de sécurité américaine (National Security Agency) comme une panoplie de logiciels utilisés par des pirates. Cette panoplie fournit des outils pour :

- capturer le trafic réseau et les mots de passe ;
- créer des portes dérobées (backdoors) dans le système ;
- collecter sur le réseau des informations sur d'autres systèmes (scanner) ;
- dissimuler que le système est compromis.

Dans notre exemple, le rootkit t0rn a été installé par le pirate. Il s'agit d'un classique du genre. À l'exception du scanner, il implémente toutes les fonctionnalités prévues par la définition.

Sur notre machine, nous avons trouvé deux répertoires utilisés par le rootkit : `/usr/src/.puta` et `/usr/info/.t0rn`.

Sniffer réseau d'un rootkit

L'exécutable `/usr/src/.puta/t0rnp` est un sniffer réseau, c'est-à-dire un programme qui écoute le réseau dans le but de récolter les éventuels mots de passe qui transitent en clair. Il faut savoir que les applications que nous utilisons couramment, comme TELNET, FTP, IMAP ou HTTP, n'effectuent en général aucun chiffrement du mot de passe au moment où ce dernier est envoyé sur le réseau.

Des logiciels tels que tcpdump, disponibles en standard sous Linux, permettent de constater combien il est facile d'écouter sur le réseau. Notons aussi Ethereal et dsniff qui sont faciles à installer.

La figure 3-6 retrace les échanges de paquets au cours de la phase d'authentification dans une session FTP. Cet exemple est facile à reproduire avec un PC Linux. Il met en évidence le risque lié à l'utilisation des applications non chiffrées.

1. Lancer Ethereal à partir du compte `root`.
2. Sélectionner le menu *capture start* et indiquer `port ftp` dans la rubrique *filter* (voir figure 3-4).

OUTILS dsniff, Ethereal et tcpdump

Ces performants outils d'analyse rendent de grands services aux administrateurs réseau. Leur utilisation est très simple et constitue une aide importante pour comprendre le fonctionnement des applications client/serveur.

- dsniff : http://monkey.org/~dugsong/dsniff/
- ethereal : http://www.ethereal.com
- tcpdump : http://www.tcpdump.org

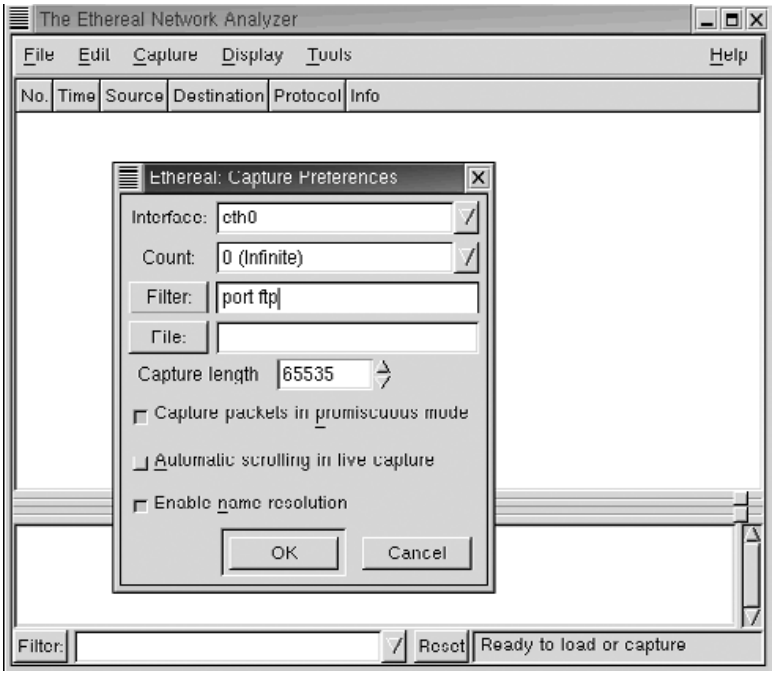


Figure 3–4
Lancement de Ethereal

Ethereal est démarré. Il écoute le réseau en ne conservant que les paquets correspondant au protocole FTP.

3 Ouvrir une connexion FTP vers un serveur quelconque pour analyser le trafic correspondant, comme indiqué à la figure 3-5.

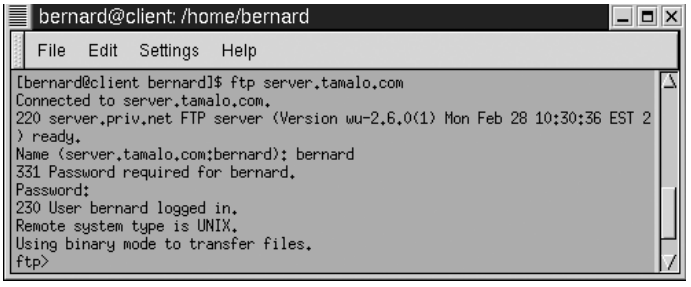


Figure 3–5 Ouverture d'une connexion FTP

La figure 3-6 montre que les informations circulent en clair sur le réseau. Ainsi, le paquet numéro 13 contient le nom de l'utilisateur `bernard`, dont nous avons écouté la connexion, tandis que le paquet numéro 17 nous renseigne sur son mot de passe : `u1arp17` !

OUTILS **Sniffer et analyse réseau**

Une différence essentielle entre un sniffer et un outil d'analyse de réseau est que le premier est écrit dans l'unique but d'extraire des couples : identification/mot de passe, tandis que le second permet d'analyser l'ensemble du trafic qui circule sur la couche de transport sur laquelle la sonde est posée.

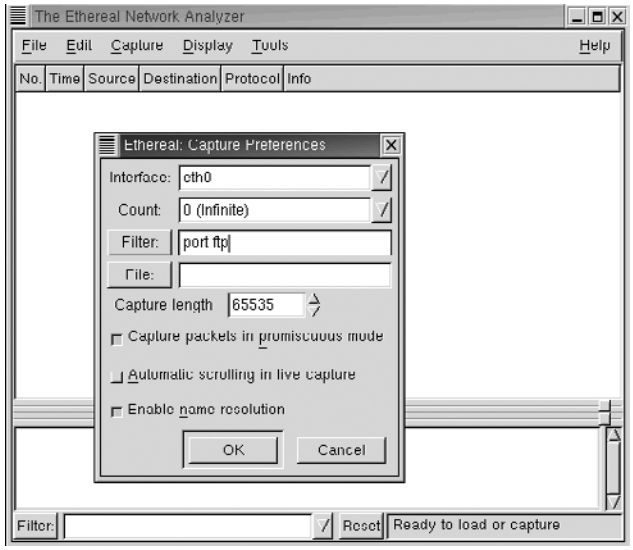


Figure 3–6 Écoute d'une session FTP avec Ethereal

Afin d'écouter sur le réseau, le pirate doit faire passer la carte Ethernet en mode promiscuous. Sur une machine Linux, et sur les systèmes Unix en général, cela nécessite un accès privilégié (compte `root`).

Le mode promiscuous

Pour comprendre ce qu'est le mode promiscuous, il est nécessaire de faire référence au modèle OSI.

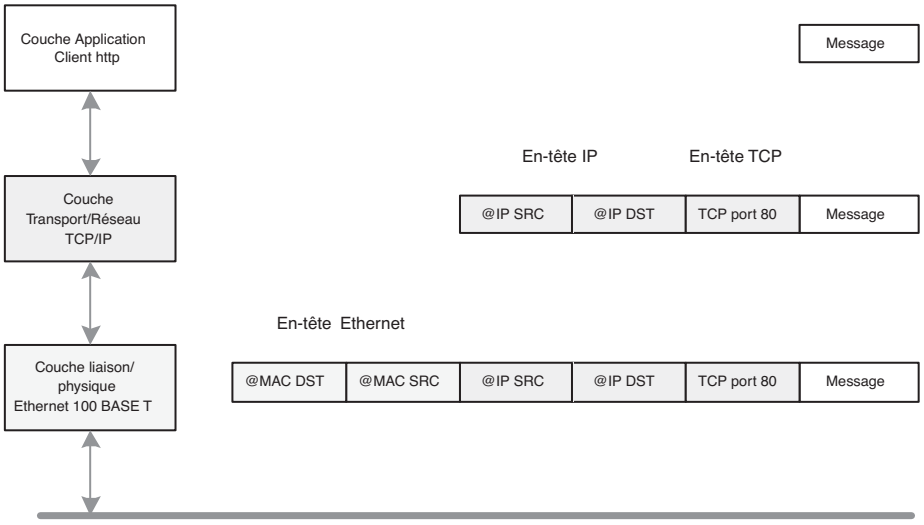


Figure 3–7 Modèle OSI : encapsulation des messages

RAPPEL Adresse MAC

Une adresse MAC (Media Access Control) est constituée de 2 champs de 3 octets chacun :

`06:60:B0:59:DE:B3`

Les trois premiers octets constituent le champ fournisseur tandis que les trois autres constituent un numéro de série. Deux interfaces Ethernet ne peuvent pas avoir la même adresse MAC – une même machine pouvant avoir plusieurs interfaces.

Notez que l'adresse MAC de destination est placée au début de la trame, ce qui permet à la carte Ethernet de déterminer tout de suite si elle doit garder la trame ou non.

La figure 3-7 montre une trame Ethernet arrivant à destination d'une machine. Elle est traitée par les couches basses du pilote réseau : les couches 1 (physique) et 2 (liaison) du modèle OSI (Open Systems Interconnection). Ces couches sont généralement implémentées dans le microcode de la carte Ethernet. Les couches supérieures, 3 et suivantes, sont quant à elles implémentées dans le noyau de Linux.

Une fonction importante des couches basses est de vérifier si la trame Ethernet est destinée ou non à la machine considérée. Cette opération est effectuée en comparant l'adresse MAC de destination contenue dans la trame avec celle de l'interface. Si les adresses coïncident, la trame est déshabillée de son en-tête Ethernet et le paquet est reconstitué pour être transmis aux couches supérieures implémentées dans le noyau de Linux. Dans le cas contraire, il ne tient pas compte de la trame. Cela ne fait pas l'affaire des mécanismes d'écoute du réseau, quelles qu'en soient les motivations. La mise en mode promiscuous de la carte Ethernet remédie à ce problème en obligeant cette dernière à transmettre toutes les trames au noyau qui se chargera de faire le tri.

Sur le système compromis, le fichier `/usr/src/.puta/system` contient les couples « nom de compte – mot de passe » qui ont été enregistrés par le sniffer lors de l'écoute frauduleuse. L'examen de ce fichier montre que le sniffer a capturé l'identifiant et le mot de passe de deux comptes appartenant à Tamalo.com, sur les machines `dial025.mon-fai.com` et `www.diffusion-tamalo.com.it`, comme le montre la figure 3-8. Les deux connexions qui ont pu être sniffées sont des sessions FTP non chiffrées.

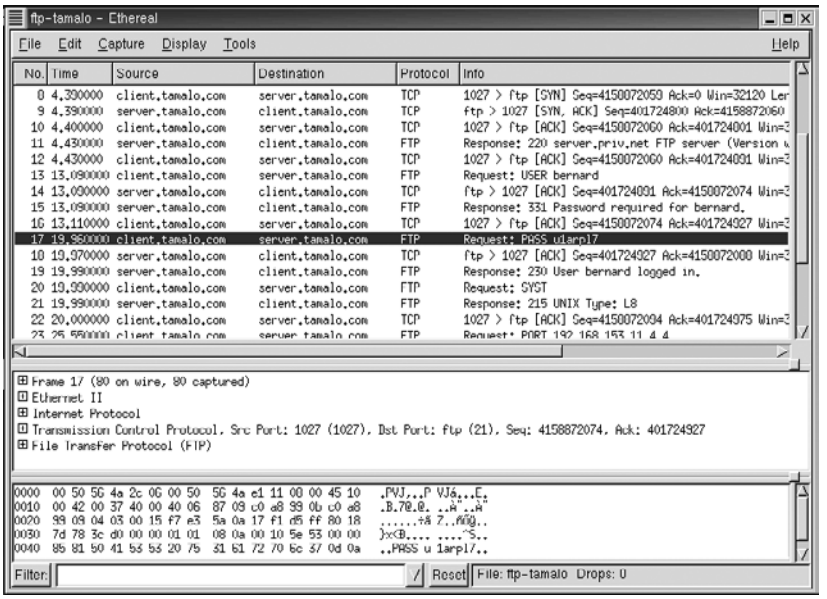


Figure 3–8 Fichier de sortie du sniffer réseau

Action

Quelques mesures doivent être mises en place sans délai :

1. Le changement des mots de passe interne et externe de tous les utilisateurs du réseau.
2. La vérification de l'historique de toutes les connexions sur des sites distants afin de savoir si des connexions frauduleuses ont eu lieu.
3. L'installation d'applications client/serveur mettant en œuvre du chiffrement pour se protéger des écoutes sur le réseau.

Rootkit : effacer les traces et masquer la présence du pirate

Le programme `/usr/src/.puta/t0rnsb` est un nettoyeur de fichiers de journalisation (logs). Son rôle est d'effacer les traces de passage du pirate.

En effet, les services de Linux sont généralement conçus pour enregistrer par l'intermédiaire d'un processus appelé `syslogd` un certain nombre de traces dans des fichiers journaux, configurés grâce au contenu du fichier `/etc/syslogd.conf` (ou `/etc/syslog-ng/`). Le plus souvent, ces fichiers sont situés dans le répertoire `/var/log`. Par défaut, les deux fichiers les plus importants du point de vue de la sécurité sont `/var/log/messages` et `/var/log/secure`.

Exemple de log

```
Jan 12 17:21:26 gw sshd[14168]:
Accepted password for root from
192.168.40.176 port 1034 ssh2
```

Cette ligne de log est produite par le serveur sshd. Elle indique une connexion SSH depuis la machine dont l'adresse IP est `192.168.40.176` sur la machine `gw`. L'heure de la connexion et l'utilisateur, `root`, sont précisés.

Un autre type de trace est enregistré dans le fichier `/var/log/wtmp`. Ce fichier stocke sous forme binaire une trace de chaque connexion sur la machine considérée. Il est possible de visualiser ces connexions avec la commande `last`. Le rootkit fournit donc au pirate des outils pour faire disparaître les traces qui le concernent afin d'éviter que l'adresse de la machine à partir de laquelle il nous a attaqués n'apparaisse dans ces fichiers.

Les outils de nettoyage des logs sont plus ou moins évolués, se situant entre la mise à zéro pure et simple du fichier, pour les rootkits primitifs, et la suppression des lignes concernant la période de présence du pirate pour les plus évolués.

Dans certains cas, l'observation des fichiers de log, même nettoyés par le rootkit, fournira des informations précieuses à l'administrateur de la machine. Par exemple, l'absence totale de logs pendant une période donnée nous indique que le pirate était probablement connecté pendant ce créneau. Si nous disposons des logs du routeur d'entrée pour cette période, ils seront riches d'informations.

À RETENIR

Il faut bien comprendre que tous les programmes, même développés localement, s'ils sont « linkés » dynamiquement, dissimuleront tout ce qui relève du piratage en cours (fichiers…) si le pirate est allé jusqu'à instrumenter la libC.

Le fichier `/usr/src/.puta/.1file` contient la liste des fichiers cachés à l'utilisateur de la machine par les commandes `ls` et `find` modifiées. Il con-

tient en particulier les fichiers du rootkit lui-même : les répertoires `.puta` et `.t0rn`, les fichiers `.1file`, `.1addr`...

Le fichier `/usr/src/.puta/.1addr` contient le début des adresses IP cachées à l'utilisateur de la machine par la commande `netstat` modifiée. Seul le début de l'adresse est donné (`192.168` par exemple) pour que la découverte de ce fichier ne trahisse pas le nom de la machine originaire de l'attaque.

Le fichier `/usr/src/.puta/.1proc` contient la liste des processus cachés à l'utilisateur de la machine par la commande `ps` modifiée. Le sniffer t0rnp est dans la liste, ainsi que la porte dérobée : `nscd`.

Le fichier `/usr/src/.puta/.1logz` contient la liste des adresses IP filtrées par le rootkit et qui n'apparaîtront pas dans les logs.

Rootkit : la porte dérobée (backdoor)

RAPPEL Porte dérobée (backdoor)

Une porte dérobée est un programme qui ménage un accès privilégié, discret et direct à celui qui sait l'employer.

De nombreux ports sont connus pour être utilisés comme portes dérobées. Certains pirates, plus fainéants que les autres, effectuent directement un *scan* de ces ports espérant y trouver une *backdoor* ouverte par un autre !

Le site http://ports.tantalo.net/index.php donne une liste des utilisations des ports, officielles ou comme portes dérobées.

Les rootkits permettent généralement au pirate d'ouvrir une porte dérobée qui écoute sur un port de son choix, supérieur à 1 024 la plupart du temps.

Dans le cas de t0rn, la porte dérobée est constituée par un serveur sshd qui écoute sur le port 15 000. Le démon `ssh` est appelé `/usr/sbin/nscd` (prétendument *Name Server Cache Daemon*).

Notez que la connexion de l'attaquant est chiffrée, ce qui lui évite d'être lui-même sniffé par ses propres outils ou par tout autre outil d'analyse réseau !

Les fichiers de configuration de ce service sont dans le répertoire `/usr/info/.t0rn` :

- `/usr/info/.t0rn/shhk.pub` contient la clé publique du serveur sshd.
- `/usr/info/.t0rn/shhk` contient la clé privée du serveur sshd.
- `/usr/info/.t0rn/shdcf` est le fichier de configuration de sshd. On y découvre qu'il est lancé sur le port 15 000.

En cas de redémarrage du système, ce « `nscd` » est relancé par le script `/etc/rc.sysinit`.

Le processus `nscd` est caché à l'exécution de `ps` grâce au fichier `/usr/src/.puta/.1proc`.

Rootkit t0rn : conclusion

Une analyse rapide du rootkit`t0rn` montre combien la découverte des fichiers de configuration de ce dernier peut être précieuse pour nous. Par exemple, en recoupant les plages d'adresses IP cachées avec les logs des routeurs d'entrée, nous sommes à même de déterminer l'adresse IP de la machine qui nous a attaqués.

Pour cette raison, des rootkits plus évolués fournissent parfois des outils de chiffrement de leurs propres fichiers de configuration. Il existe par exemple des rootkits possédant un fichier de configuration unique chiffré par un « ou

exclusif » de chaque octet avec 255. Heureusement, le décryptage de ces fichiers constitue un défi qui motive de nombreux amateurs, et l'algorithme de chiffrement de ces fichiers ne tarde jamais à être diffusé sur Internet.

Encore une fois, le fait que de nombreux rootkits soient identifiés et leur fonctionnement bien connu ne dispense en aucun cas du reformatage du système de fichiers ni de la réinstallation complète d'une machine compromise.

Détecter la compromission à partir des logs

Depuis quelques semaines, nous avions mis en place chez Tamalo.com une centralisation systématique des logs des machines Linux vers une machine de collecte n'offrant aucun autre service à l'extérieur. Cette centralisation va nous aider à retrouver les logs qui ont étés effacés par le rootkit sur notre serveur.

Copie de log retrouvée sur la machine de collecte : une attaque en deux phases

```
Dec 26 19:02:22 Tamalo1 inetd[158]: printer[5336] from 192.168.20.29 1623 ❶
Dec 26 19:02:22 Tamalo2 inetd[175]: printer[5227] from 192.168.20.29 1627
Dec 26 19:03:55 Tamalo1 SERVER[2320]: Dispatch_input: bad request line ❷
'BBÜóÿ¿Ýóÿ¿Þóÿ¿ßóÿ¿XXXXXXXXXXXXXXXXXX000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000480000000
00001073835088security000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000006??????????????????????????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????????????????????????????????????
??????????????1Û1É1À°FÍ??å1Ò²f?Ð1É?ÈC?]øC?]ôK?Mü?MôÍ?1É?EôCf?]ìfÇEî^O'?Mð?Eì?EøÆEü^P?Ð?MôÍ??ÐCCÍ??ÐC
Í??Ã1É²??ÐÍ??ÐAÍ?ë^X^?u^H1À^F^G?E^L°^K?ó?M^H?U^LÍ?èäÿÿÿ/bin/sh'
```

L'analyse de la copie des logs sur la machine de collecte met clairement en évidence les deux étapes de l'attaque. Dans un premier temps, à 19:02:22 ❶, un scan horizontal de notre réseau sur le port d'impression 515 identifie les machines qui présentent une faille. Dans un deuxième temps, à 19:03:55 ❷, cette faille est exploitée sur Tamalo1. L'enchaînement entre le scan et l'exploit est extrêmement rapide (1 minute 30), ce qui suppose que le pirate a utilisé un script qui lance automatiquement l'opération en fonction du résultat du scan.

Il apparaît que nous avons été la victime d'une attaque sur le service d'impression `lprng`. Remarquez dans la ligne de log précédente la taille importante de l'argument fourni à `lpr`, ainsi que la présence de la chaîne de caractères `/bin/sh` à l'intérieur de cet argument.

Cet argument est passé par `lprng` au `syslogd` par l'intermédiaire de la fonction `use_syslog()`. Cette dernière présente une faille : elle utilise l'argument reçu par le service pour déterminer le format de la chaîne de caractères à transmettre. Un format inattendu contenant un appel à `/bin/sh` sera donc exécuté par le service `lprng` autorisant le pirate à exécuter le code de son choix.

Cette vulnérabilité du service d'impression `lprng` de Linux était parfaitement décrite dans un avis de sécurité de la société Red Hat qui nous était parvenu quelques temps auparavant.

Avis de sécurité de Red Hat sur la vulnérabilité du service lprng

```
-----------------------------------------------------------------
-----
Red Hat, Inc. Security Adviso ry
Synopsis: LPRng contains a critical string format bug
Advisory ID: RHSA-2000:065-04
Issue date: 2000-09-26
Updated on: 2000-10-04
Product: Red Hat Linux
Keywords: LPRng security lpd printing lpr syslog
Cross references: N/A
-----------------------------------------------------------------
-----
1. Topic:
LPRng has a string format bug in the use_syslog function which
could
lead to root compromise.

2. Relevant releases/architectures:
Red Hat Linux 7.0 - i386
3. Problem description:
LPRng has a string format bug in the use_syslog function. This
function returns user input in a string that is passed to the
syslog()
function as the format string. It is possible to corrupt the
print
daemon's execution with unexpected format specifiers, thus
gaining root
access to the computer. The vulnerability is theoretically
exploitable
both locally and remotely.
```

Origine de l'attaque

Une négligence du pirate donnera de façon non ambiguë l'adresse de la machine à partir de laquelle il nous a attaqués. Dans le répertoire `/dev/ptyi` utilisé par le pirate pour déposer quelques outils, nous avons découvert un fichier `core`.

Une simple commande `strings` appliquée à ce fichier extrait l'ensemble des chaînes de caractères contenues dans la mémoire. Elle nous dévoile l'environnement complet dans lequel travaillait le pirate au moment de l'incident !

B.A.-BA Fichier core

Un fichier `core` est créé lorsque le noyau Linux interrompt sans sommation le déroulement d'un processus tentant de commettre une action interdite, par exemple accéder à une portion de la mémoire ne lui appartenant pas. Ce fichier est une image de la mémoire occupée par le processus au moment du problème. Un fichier `core` peut être ouvert avec un débogueur afin de connaître l'endroit exact du plantage, ainsi que l'environnement complet du programme au moment de l'incident.

Chaînes de caractères extraites du fichier core

```
HOME=/root
USER=root
LOGNAME=root
PATH=/usr/sbin:/sbin:/usr/bin:/bin:/usr/X11R6/bin
MAIL=/var/spool/mail/root
SHELL=/bin/tcsh
SSH_CLIENT=192.168.16.58 1029 15000
SSH_TTY=/dev/pts/2 TERM=xterm HOSTTYPE=i386-linux VENDOR=intel
OSTYPE=linux MACHTYPE=i386
```

Pour des raisons de confidentialité, les deux premiers octets de l'adresse source de l'attaque ont été remplacés par `192.168`.

La variable `SSH_CLIENT` indique très clairement que la machine dont l'adresse IP est `192.168.16.58` était connectée sur le port 15 000 de notre serveur, ce qui correspond à la porte dérobée.

Une interrogation des bases *whois* détermine rapidement la provenance de l'attaque.

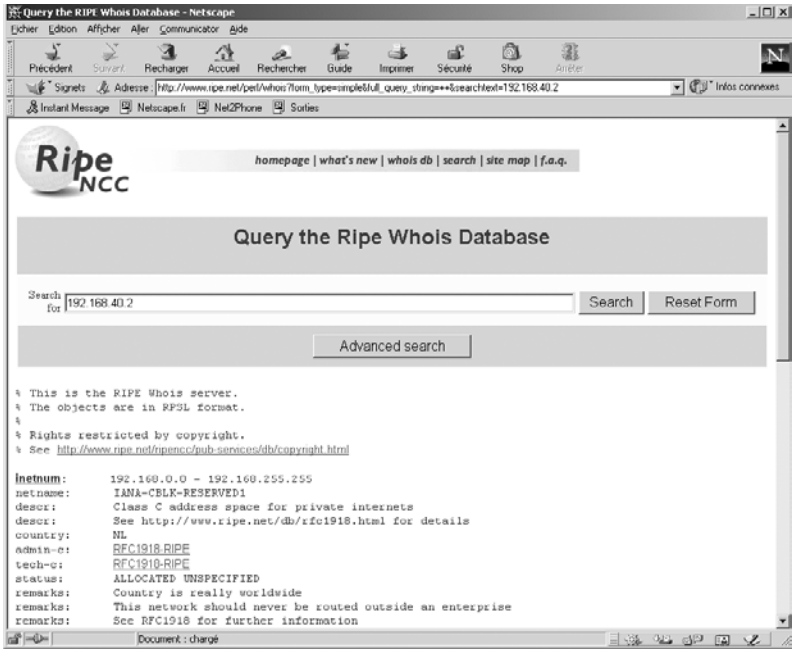


Figure 3–9 Les bases whois : Ripe

L'organisation qui gère l'adresse de notre pirate n'étant pas localisée en France, nous décidons de contacter notre CERT afin qu'il transmette nos récriminations à son homologue.

En parallèle, nous tentons une protestation par courrier électronique à l'adresse `abuse` du réseau source de l'attaque, ainsi qu'avec le contact technique indiqué dans la base whois. Nous obtenons très rapidement une

ORGANISMES Les bases Whois

Les bases whois déterminent à quel organisme a été affecté un domaine IP. Elles fournissent les coordonnées du responsable fonctionnel des adresses attribuées et de la personne à contacter en cas de problème. Ces bases sont au nombre de trois, RIPE pour l'Europe et l'Afrique, ARIN pour les États-Unis et APNIC pour l'Asie. Elles peuvent être interrogées à partir de leur site Web.

- http://www.ripe.net/perl/whois
- http://www.arin.net/whois/index.html
- http://www.apnic.net/

réponse à notre courrier, l'administrateur de la machine concernée nous indiquant qu'il venait de découvrir que sa machine était également compromise.

En résumé…

Les attaques des systèmes informatiques sont de plus en plus automatisées. Leur scénario est assez reproductible : découverte d'une faille dans un service, publication d'un « exploit », scan réseau et tentative de compromission.

Les pirates utilisent des panoplies d'outils qui cachent leur présence, capturent les mots de passe circulant sur le réseau, installent des portes dérobées ou enfin scannent un autre réseau.

L'analyse d'une machine compromise fait apparaître les outils mis en œuvre par le pirate. Elle révèle comment le réseau a été pénétré et si d'autres machines présentent les mêmes failles. Elle permet souvent d'identifier l'origine de l'attaque, mais rarement de remonter jusqu'au pirate qui, en général, se protège par de nombreux rebonds.

Pour se protéger contre ces attaques, deux types d'actions seront décrits dans les chapitres qui suivent : le recours à des applications client/serveur mettant en œuvre du chiffrement pour interdire l'écoute réseau, le filtrage des services vulnérables par la mise en place de pare-feu et la segmentation du réseau.

ORGANISMES Les CERT : Computer Emergency Resource Team

Un CERT est une organisation qui travaille sur les problèmes de sécurité informatique pour une communauté donnée. En France, il y en a quatre. Le plus ancien est le CERT Renater. Il concerne la communauté université-recherche. Chaque CERT dispose de moyens techniques et humains propres. Dans certaines affaires, les CERT peuvent travailler en relation avec les autorités judiciaires.

Au niveau mondial, les CERT sont en relation entre eux par le biais d'un forum appelé FIRST (Forum of Incident Response and Security Team). Les CERT échangent ainsi des informations sur les failles nouvelles et les incidents de sécurité courants.

Les CERT effectuent une veille technologique par rapport aux failles des logiciels pouvant donner lieu à une attaque. Ils diffusent des avis de sécurité à leurs correspondants et les avertissent par des messages d'alerte lorsque certaines attaques prennent des proportions très importantes.

- http://www.cert.org
- http://www.certa.ssi.gouv.fr

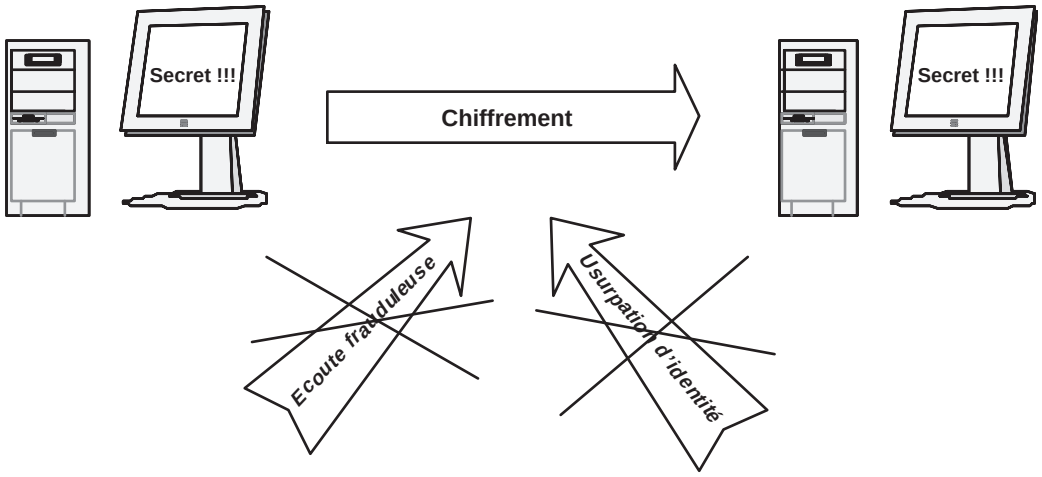
CONVENTION L'adresse abuse

Il est recommandé à l'administrateur d'un domaine nommé `nom.de.domaine` de créer l'adresse abuse électronique correspondante (`abuse@nom.de.domaine`) qui est redirigée vers la sienne.

Une personne qui aurait à se plaindre d'un comportement anormal d'une des machines de `nom.de.domaine` pourrait ainsi le signaler à l'administrateur dudit domaine par l'envoi d'un simple courrier électronique.

L'adresse abuse s'avère également utile pour l'administrateur du réseau concerné. Par exemple, c'est grâce à cette adresse qu'il sera informé en cas de problème avec des machines de son domaine.

chapitre 4


Secret !!!
Chiffrement
Secret !!!
Ecoute frauduleuse
Usurpation d'identité

INCLUDES
OpenSSL
CRYPTOGRAPHY SOFTWARE
OpenSSH

Chiffrement des communications avec SSH et SSL

Pour limiter la portée des écoutes et garantir l'identité des machines qui communiquent entre elles, il est nécessire d'utiliser des solutions de chiffrement.

SOMMAIRE

- Utilisation du chiffrement
- Principe du chiffrement
- Confidentialité et authentification
- Le protocole SSL (Secure Socket Layer)
- Communication sécurisée avec SSH (Secure Shell)

MOTS-CLÉS

- Chiffrement symétrique et asymétrique
- Confidentialité
- Authentification
- Signature électronique
- Infrastructure à gestion de clés
- Secure Socket Layer (SSL)
- Secure Shell (SSH)
- Réseau privé virtuel (RPV)
- Virtual Private Network (VPN)

VOCABULAIRE Chiffrer, déchiffrer, décrypter

Voici les conventions et les termes que nous utiliserons au cours de cet ouvrage.

- *Chiffrer* ou coder est la transformation d'un message (de données) en clair en un message codé compréhensible seulement par celui qui dispose du code ou de la clé.
- *Déchiffrer* ou décoder est l'opération de transformation d'un message chiffré en un message clair en utilisant la clé.
- *Décrypter*, consiste à extraire le texte en clair d'un message chiffré sans en connaître la clé. Cette opération est en général l'œuvre des pirates ou des cryptanalystes.

Comme les administrateurs système de l'entreprise Tamalo.com ont pu le constater lors de leur mésaventure (voir chapitre 3), il est très facile à celui qui a compromis une machine de profiter des faiblesses de l'infrastructure réseau et des protocoles qui l'animent. L'écoute frauduleuse (le *sniff*) du réseau est la première action que le pirate entreprendra sur une machine dont il aura pris le contrôle. Il pourra ainsi capturer certaines informations sensibles tels que les mots de passe qui transitent en clair sur le réseau et les exploiter pour rebondir de système en système, voire de site en site.

Nous étudierons au cours de ce chapitre les bases du chiffrement et les solutions retenues par les informaticiens de Tamalo.com pour sécuriser les communications. Nous aborderons les principes de fonctionnement de la bibliothèque SSL (Secure Socket Layer) et ses applications pour chiffrer une session TCP, ainsi que le déploiement de SSH (Secure Shell) pour accéder à un shell distant de façon chiffrée.

Les quatre objectifs du chiffrement

Authentification

Le chiffrement apporte une réponse au problème d'authentification dans les protocoles de communication qui reposent aujourd'hui pour la plupart sur la confiance dans l'infrastructure réseau. Hélas, les réseaux sont sensibles à l'écoute frauduleuse et à l'usurpation d'identité de machine. En d'autres termes, n'importe qui peut espionner son voisin et se faire passer pour une machine connectée au réseau. Il suffit de posséder les outils adéquats et ceux-ci sont légion.

Défi (challenge)

Un défi met en œuvre une donnée grâce à laquelle un programme peut prouver à un autre qu'il détient une information (par exemple un mot de passe) sans devoir l'expédier.

Le chiffrement des mots de passe ou des informations sensibles nécessaires pour l'authentification diminue le risque lié à l'espionnage (*sniff*) du réseau. Le chiffrement à clé publique décrit un peu plus loin dans ce chapitre, introduit des techniques d'authentification ne reposant plus sur la transmission d'un mot de passe permanent mais sur un « défi » (*challenge*).

Les informations sensibles ne transitent alors plus en clair sur le réseau. Les personnes physiques, les machines ou encore les services réseau peuvent ainsi s'authentifier mutuellement.

Ce point sera abordé plus en détail avec la description du fonctionnement de SSL et la mise en œuvre du produit de connexion sécurisée SSH.

Intégrité

Les techniques de chiffrement permettant la signature de l'empreinte d'une donnée, d'un document ou d'un programme, garantissent qu'aucune modifi-

cation n'a été effectuée par une tierce personne. Cela présente beaucoup d'intérêt dans la lutte contre les virus par exemple. S'il est possible de garantir l'origine et l'intégrité d'un programme diffusé, l'utilisation d'un logiciel antivirus devient obsolète. Ce mécanisme de signature des programmes est aujourd'hui encore peu répandu mais se développe de plus en plus.

Confidentialité

Lors d'un échange de documents, le chiffrement va permettre d'interdire la lecture du contenu par une personne non autorisée qui aurait pu écouter l'échange. Dans un contexte informatique, il n'est pas possible de faire confiance au réseau. Le chiffrement de données sensibles comme les mots de passe permet de rendre plus fiables les mécanismes d'authentification en éliminant le risque d'écoute frauduleuse du réseau. Le chiffrement de tout autre type de données est également parfois nécessaire pour garantir la confidentialité de celles-ci afin de protéger l'outil de travail informatique.

Signature électronique

Le chiffrement utilisé dans le cadre de la signature électronique de documents permet l'identification formelle de leurs auteurs ou expéditeurs. Si la signature authentifiée des courriers électroniques était généralisée, les courriers électroniques non sollicités (SPAM) ne seraient plus un problème. Il suffirait de refuser les courriers anonymes pour ne plus voir sa boîte à lettres électronique polluée.

Facteurs de fiabilité des techniques de chiffrement

La fiabilité du chiffrement repose sur trois éléments :

- la qualité de l'algorithme mathématique utilisé plutôt que le secret de celui-ci ;
- la qualité de l'implémentation de l'algorithme. Il faut savoir que la plupart des tentatives de piratage s'attaquent à l'implémentation qui est faite de ces algorithmes de chiffrement (recherche de *buffer overflow*) plutôt qu'à l'algorithme lui-même ;
- la bonne gestion du secret, c'est-à-dire de la ou des clés de chiffrement. L'algorithme étant la plupart du temps public, si la ou les clés utilisées sont connues, il n'y a bien évidemment plus aucune sécurité.

Intégrité des paquetages

De nombreuses distributions de Linux offrent maintenant des outils facilitant l'installation et la mise à jour des paquetages (APT, RPM…). Ils intègrent des logiques vérifiant automatiquement l'origine et l'intégrité des paquetages téléchargés qui sont « signés et scellés » afin de ne pas exposer l'administrateur au déploiement d'un ensemble apparemment officiel mais modifié de façon dangereuse, par exemple pour accueillir un rootkit.

Algorithmes de chiffrement symétrique et asymétrique

Deux types de chiffrement sont aujourd'hui couramment utilisés. Il s'agit du chiffrement symétrique et du chiffrement asymétrique. Ils sont complémentaires et ont chacun un rôle déterminé dans l'établissement et la gestion des connexions sécurisées.

Chiffrement symétrique

Aussi connu sous le nom de chiffrement à clé secrète, le chiffrement symétrique utilise une seule clé pour chiffrer et déchiffrer les données. Les deux parties doivent partager l'information qui permettra de recouvrer les données. Expéditeur et destinataire doivent s'être échangés au préalable – par un moyen sûr – la clé qui permettra à chacun d'eux de chiffrer ou déchiffrer des données.

DANS LA PRATIQUE Chiffrement symétrique et confidentialité

Dans l'exemple de la figure 4-1, Bill souhaite envoyer un message confidentiel à Monica via un système de messagerie électronique. Pour cela, il chiffre un document avec une clé secrète que détient aussi Monica. Le document chiffré est envoyé à cette dernière. Lorsque Monica reçoit le message, il ne reste plus qu'à utiliser la clé secrète partagée avec Bill pour le déchiffrer.

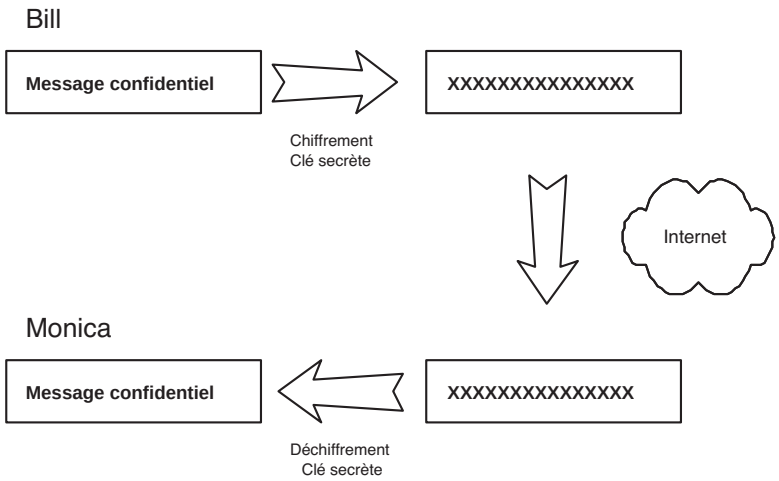


Figure 4–1 Chiffrement symétrique et confidentialité des données

L'intérêt du chiffrement symétrique réside principalement dans sa rapidité d'exécution. Il sera en général utilisé pour chiffrer les transferts réseau. Son inconvénient est qu'il est nécessaire, pour assurer la confidentialité des données avec ses interlocuteurs, de partager une clé sécrète différente avec chacun d'entre eux. Cette contrainte peut rapidement devenir pénalisante avec le nombre de clés à gérer.

Voici quelques exemples d'algorithmes de chiffrement symétrique couramment utilisés.

- Data Encryption Standard (DES), inventé en 1977, utilise une clé de 56 bits.
- 3DES est l'application de l'algorithme DES à trois reprises lors du chiffrement avec trois clés différentes (A–B–C 168 bits) ou deux clés différentes (A–B–A 112 bits).
- RC2, RC4 et RC5 utilisent des clés jusqu'à 1024 bits.
- Blowfish
- International Data Encryption Algorithm IDEA.
- Advanced Encryption Standard (AES), a été proposé en 2001 lors d'un concours international visant à trouver un remplaçant à DES.
- Quand cela est possible, AES est l'algorithme à utiliser.

Chiffrement asymétrique

Pour utiliser le chiffrement asymétrique, également connu sous le nom de chiffrement à clé publique, chaque utilisateur désirant échanger des données doit posséder un couple de clés composé d'une clé privée et d'une clé publique. La clé publique, comme son nom l'indique, est diffusée le plus largement possible, par exemple dans un annuaire, alors que la clé privée doit rester connue de son seul propriétaire. Les données chiffrées avec la clé publique ne peuvent être déchiffrées qu'avec la clé privée correspondante et réciproquement. Il n'est en théorie pas possible de déduire la clé privée de la clé publique. Les utilisations du chiffrement asymétrique sont diverses. Il est adapté et prévu pour assurer la confidentialité et la signature électronique de documents.

DANS LA PRATIQUE Chiffrement asymétrique et confidentialité

Notre ami Bill souhaite de nouveau envoyer un message confidentiel à Monica, mais cette fois en utilisant un algorithme à clé publique (figure 4-2). Pour ce faire, il chiffre son message confidentiel avec la clé publique de Monica. Le message est acheminé à cette dernière, qui utilise sa clé privée, seule capable de déchiffrer les messages chiffrés avec sa clé publique, pour connaître le contenu de cette nouvelle correspondance.

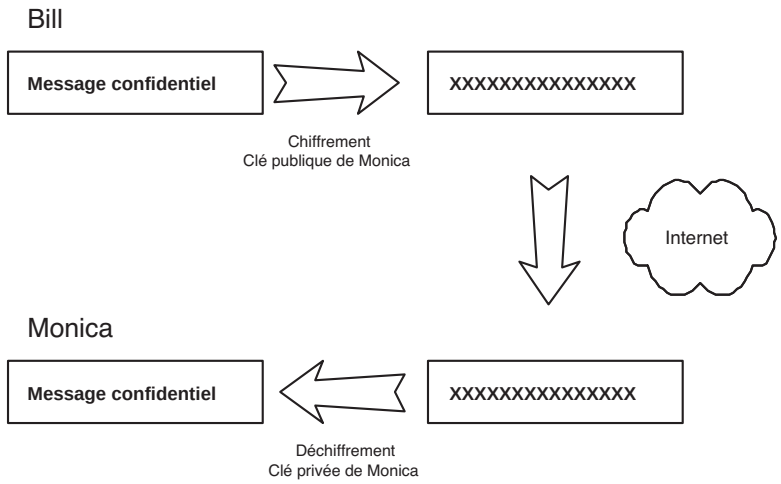


Figure 4–2 Chiffrement asymétrique et confidentialité des données

DANS LA PRATIQUE Chiffrement asymétrique et signature électronique

Après quelques mésaventures, Bill et Monica décident de signer électroniquement leurs messages afin d'en garantir la source (figure 4-3). Bill chiffre l'empreinte de son message avec sa clé privée. Seule sa clé publique détenue par Monica peut déchiffrer cette empreinte permettant de valider l'intégrité de ce nouveau message et donc garantir que Bill en est bien l'expéditeur.

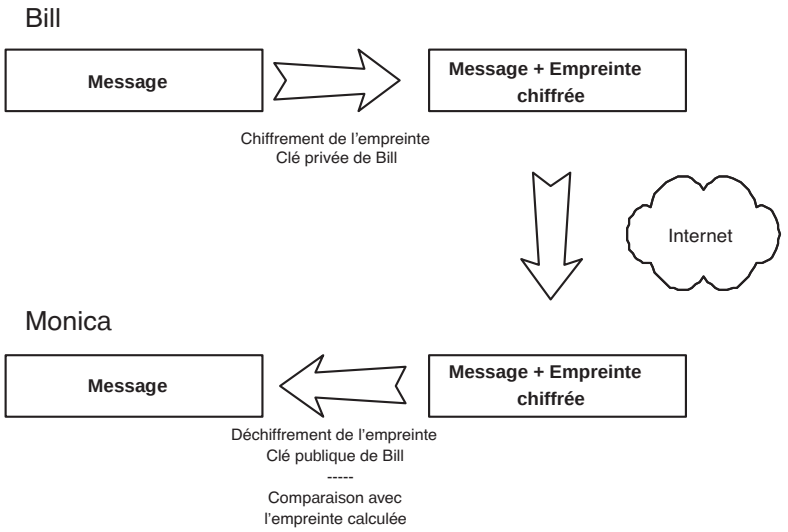


Figure 4–3 Chiffrement asymétrique et signature électronique des données

DANS LA VRAIE VIE...

Le chiffrement à clé publique est couramment utilisé de nos jours dans la correspondance électronique. En utilisant successivement le chiffrement d'un courrier avec la clé privée de l'expéditeur puis la clé publique du destinataire, il est possible de garantir à la fois la confidentialité du document et l'identité de l'expéditeur.

DSA (Digital Signature Algorithm) et RSA, qui tire son nom de ses auteurs Rivest, Shamir et Adleman, sont les deux principaux algorithmes de chiffrement à clé publique, utilisés en particulier par SSL et SSH. Beaucoup moins performants donc plus coûteux que les algorithmes de chiffrement symétrique, ils sont, en général, utilisés dans les phases d'authentification et de négociation préalables à l'établissement de la connexion sécurisée. Une clé de session utilisée pour le chiffrement symétrique de la connexion sécurisée sera échangée entre les deux parties par des mécanismes de chiffrement à clé publique. Voir en annexe A de l'ouvrage, Infrastructure à gestion de clés, la génération d'une bi-clé pour un utilisateur ou un service.

RÉCRÉATION La stéganographie

La stéganographie est une technique qui consiste à dissimuler un message dans un ensemble de données d'apparence anodine. Plus concrètement, il est possible de cacher des informations de tous types dans un fichier contenant une image sans que cela soit détectable lors de la visualisation. Cela peut servir en particulier pour insérer des « copyrights » dans des photos sans que ceux-ci soient visuellement détectables.

Cette technique peut également être utilisée pour échanger des documents confidentiels à partir d'images publiées sur un site Web accessible à tous.

Une liste d'outils de stéganographie, pour la plupart en libre distribution, est disponible à l'adresse :

- http://www.stegoarchive.com

Qui pourrait croire que le fichier original au format JPEG de cette photo contient le numéro et le code de la carte bancaire du personnage aux lunettes noires ?

La clé privée RSA est sauvegardée dans le répertoire `$HOME/.ssh/id_rsa` et la clé publique correspondante dans le fichier `$HOME/.ssh/id_rsa.pub` par défaut.

Dans le cas de l'algorithme DSA, la clé privée est dans le fichier `$HOME/.ssh/id_dsa` et la clé publique correspondante dans le fichier `$HOME/.ssh/id_dsa.pub`.

Au moment de la production du couple de clés, il est demandé une *passphrase*, long mot de passe servant à chiffrer le fichier contenant la clé privée. Cette protection permet d'éviter à quiconque pourrait accéder à vos fichiers d'en extraire le contenu. Souvenez-vous que la clé privée ne doit être connue que de son propriétaire.

La clé publique est copiée dans le fichier `$HOME/.ssh/authorized_keys` du compte distant sur la machine vers laquelle les connexions seront initiées.

```
$ scp -p $HOME/.ssh/id_rsa.pub server:.ssh/authorized_keys
```

Le fichier `authorized_keys` peut contenir plusieurs clés publiques. Lorsqu'une connexion SSH est ouverte vers la machine `server`, l'utilisateur doit fournir sa *passphrase* afin d'accéder à la clé privée nécessaire pour que le mécanisme d'authentification décrit sur la figure 4-9 aboutisse.

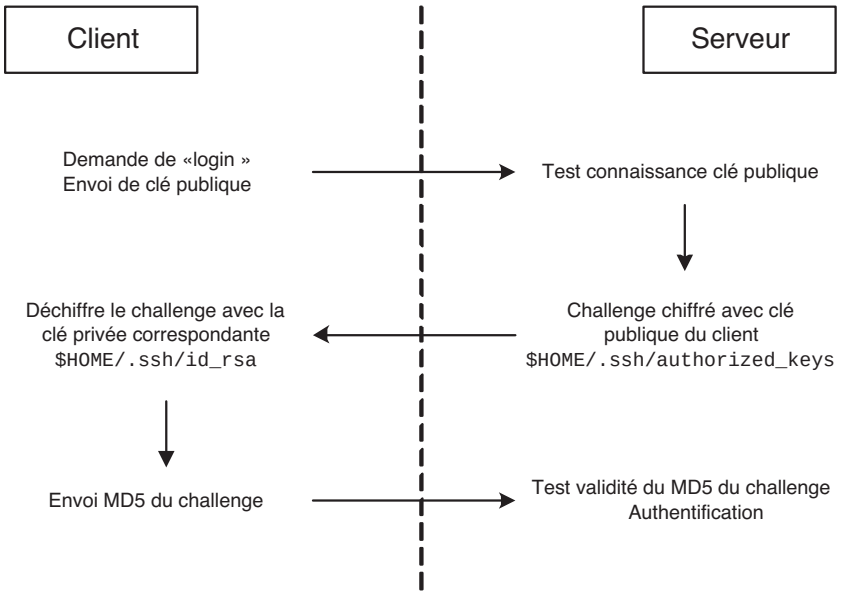


Figure 4–9
Authentification RSA

Cette technique d'authentification par couple de clés est certainement la plus robuste et reste celle à conseiller aux utilisateurs. Elle présente néanmoins un point faible. En effet, un utilisateur souhaitant ne plus saisir de mot de passe ou de *passphrase* pour se connecter aura vite compris qu'il n'est

pas obligé d'en saisir une lors de la génération du couple de clés. Vous l'aurez aisément compris, le danger réside alors dans le risque d'une mauvaise protection du fichier contenant la clé privée de l'utilisateur. Quiconque aura réussi à s'emparer de ce fichier pourra l'utiliser pour se connecter sous l'identité du propriétaire.

Afin d'offrir le confort de ne pas saisir sa *passphrase* 150 fois par jour et pour dissuader la pratique de la *passphrase* vide, les auteurs de SSH ont imaginé une solution. Le programme `ssh-agent` permet de garder en mémoire les clés privées de l'utilisateur pendant toute la durée de sa session X11 ouverte sur un poste de travail. Le démon `ssh-agent` rend disponible automatiquement la clé appropriée à la commande `ssh` lorsqu'elle est exécutée pour établir une connexion distante.

La *passphrase* associée à une clé privée n'est renseignée qu'une seule fois pendant toute la durée de la session par l'utilisateur, lors de l'ajout au magasin à clés privées maintenu par le démon `ssh-agent`. Ce dernier s'exécute en tâche de fond sur le poste de travail. Il est lancé au début de la session X11 avant le démarrage du gestionnaire de fenêtres ou *window manager* en anglais.

Le lancement du démon `ssh-agent` doit apparaître dans le fichier « `~/.xsession` » de l'utilisateur de la manière qui suit, avant le lancement du *window manager* :

```
# Lancement du démon ssh-agent dans le fichier .xsession
eval `ssh-agent`
```

L'exemple suivant décrit les étapes nécessaires à la génération d'un couple de clés, et à l'ajout de la clé privée dans le magasin de `ssh-agent`.

Génération d'un couple de clés RSA :

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/bill/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/bill/.ssh/id_rsa.
Your public key has been saved in /home/bill/.ssh/id_rsa.pub.
The key fingerprint is:
c1:a6:2c:6a:bf:6f:70:cf:95:9f:dd:e2:31:3f:83:2c
bill@client.tamalo.com
```

Consultation du contenu du magasin à clés maintenu par le démon `ssh-agent` :

```
$ ssh-add -L
The agent has no identities.
```

Ajout d'une clé privée au magasin :

```
$ ssh-add .ssh/id_rsa
Enter passphrase for .ssh/id_rsa:
Identity added: .ssh/id_rsa (.ssh/id_rsa)
```

Vérification de la présence de cette clé dans le magasin :

```
$ ssh-add -L
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEA6/tKfSnIRWnD6ILyKpuZFn9IruGXFHh7qEr
EaSw8Wx9Gaxpcfaek+56XXIJFUMvQhjHyEn757s2rL/IDG3tFuNBwhXwkVQQuzx
+bQTZBkzcIuJYHgogs5TR56yqwaqEkegMFo1xulM9Ue6nV5coG7zX1yV/jZ94Cv
b26PvQBmTO= .ssh/id_rsa
```

Une fois que la clé publique correspondante à la clé privée est copiée sur la machine distante dans le fichier `authorized_keys`, il devient possible de se connecter sans donner de mot de passe ou de *passphrase*.

La commande `ssh-add` permet également de supprimer l'ensemble des clés maintenues par le démon `ssh-agent`.

```
$ ssh-add -D
All identities removed.
```

L'arrêt de se dernier est réalisé par l'invocation de la commande `ssh-agent` avec l'option `-k`.

```
$ ssh-agent -k
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 32387 killed;
```

Relais d'affichage X11

Dans le monde Unix, lorsqu'une application graphique s'exécute sur un premier système, il est possible de déporter son affichage sur un autre système doté d'un écran et d'un serveur d'affichage X11. Avec une configuration permissive des accès au serveur d'affichage X11, n'importe qui peut prendre le contrôle de l'affichage et des périphériques associés. Notamment, il devient possible de récupérer les événements clavier et donc d'espionner ce que fait l'utilisateur de la machine. SSH vient combler ce trou de sécurité en offrant la possibilité de rediriger la requête d'affichage dans le tunnel chiffré de la connexion.

EN PRATIQUE Rediriger X11 dans un tunnel SSH

Lors de la connexion depuis `ordi01` sur `ordi02` (figure 4-10), le démon `sshd` positionne une variable d'environnement `DISPLAY` à `ordi02:1`, qui correspond à un écran virtuel dont il a la gestion. Lorsqu'une application graphique lancée sur la machine distante `ordi02` fait une requête d'affichage X11 sur cet écran virtuel, le démon `sshd` intercepte la requête et la redirige dans le tunnel chiffré de la session SSH. La commande client SSH sur la machine `ordi01` reçoit la requête d'affichage et la transfère sur le descripteur d'écran local, `ordi01:0`, géré par le gestionnaire d'affichage X11. Il est donc possible sur `ordi01` de n'accepter plus que les requêtes d'affichage local de la session utilisateur, éliminant par-là même le risque d'espionnage.

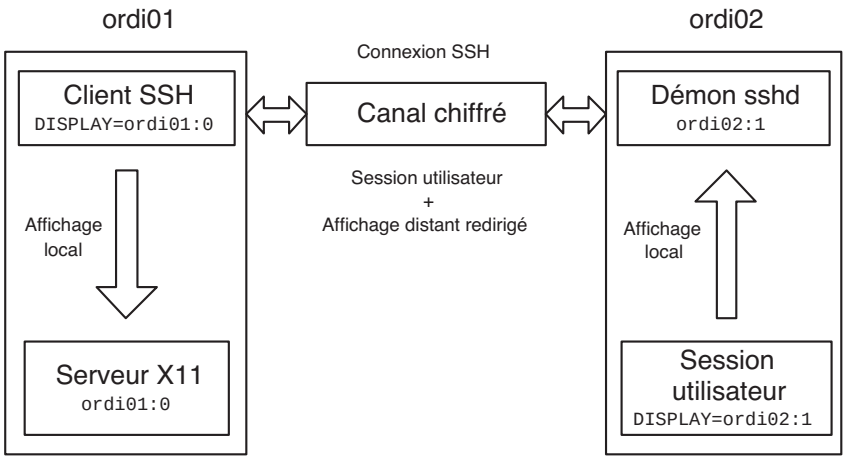


Figure 4–10
Principe du relais X11

Gestion des accès au service SSH

Plusieurs mécanismes coexistent pour gérer les accès au serveur SSH d'une machine.

- Les directives `AllowUsers` et `AllowGroups` du fichier de configuration `/etc/ssh/sshd_config`, permettent la sélection des comptes utilisateur ou des groupes autorisés à se connecter à partir du réseau. L'utilisation de ces paramètres est rudimentaire et exclusive, mais elle peut néanmoins se révéler utile dans quelques cas de figure.
- De la même manière, l'option `PermitRootLogin` offre la possibilité de refuser explicitement les connexions sous le compte administrateur root.

Enfin, l'utilisation conjointe de SSH et des librairies de filtrage réseau TCP Wrapper permet la mise en place de règles d'accès basées sur l'adresse IP, le nom de machine ou encore le nom de domaine Internet de la source initiant la connexion réseau (voir chapitre 5).

Dépannage

En cas d'embarras lors de l'utilisation de ssh, il conviendra d'employer l'option `-v` du client afin de le rendre plus verbeux et de lire à mesure les traces du serveur (sshd) sollicité.

L'alternative VPN

L'utilisation de SSL et de SSH permet de répondre à la grande majorité des besoins de confidentialité et d'authentification nécessaires aux différents types de communications de Tamalo.com.

Néanmoins, dans certains cas, des protocoles réseau applicatifs peuvent ne pas offrir ces possibilités de chiffrement parce qu'elles n'ont, au moment de leur spécification, pas été prévues. Une solution convenable pour sécuriser les transactions de ces services dans de telles situations, réside dans l'utilisation d'un Réseau privé virtuel (RPV) ou *Virtual Private Network* en anglais (VPN).

La technologie des VPN consiste à mettre en œuvre les techniques de chiffrement nécessaires à la sécurisation des communications pour l'authentification et la confidentialité des données, non plus au niveau applicatif comme cela est le cas pour SSL et SSH, mais au niveau de la couche réseau, afin que leur utilisation devienne transparente pour les services réseau de haut niveau.

Plusieurs initiatives, en général complémentaires, ont vu le jour pour répondre au besoin de VPN sur le système Linux. Dans le cas du projet OpenVPN, la fourniture d'outils s'étend même aux systèmes Windows, Solaris, Mac OS X et la lignée des BSD. Il s'agit d'un VPN reposant sur la couche de sécurisation SSL très facile à déployer, donc très populaire. C'est l'outil incontournable pour celui qui souhaite commencer avec les VPN sur le système Linux. Le projet FreeS/WAN propose quant à lui une implémentation des protocoles IPSec et IKE pour les noyaux Linux. Il intervient au niveau de couches plus profondes du kernel que le projet OpenVPN. La description des couches logicielles proposées par ces deux projets, ainsi que leur mise en œuvre, ne seront pas abordées dans cette édition.

ORGANISATION Le consortium VPN

- http://www.vpnc.org

RÉFÉRENCE Le projet FreeS/WAN

Le projet FreeS/WAN propose une implémentation des protocoles IPSEC et IKE pour Linux. Ces protocoles sont les standards utilisés dans le monde VPN. Ils sont implémentés au niveau des couches réseau du noyau Linux. Arrivé au terme de son développement, FreeS/WAN a donné naissance aux projets Openswan et Strongswan.

- http://www.freeswan.org
- http://www.openswan.org
- http://www.strongswan.org

RÉFÉRENCE Le projet OpenVPN

Le projet OpenVPN implémente une extension sécurisée réseau des couches 2 et 3 du modèle OSI en utilisant le standard TLS/SSL. Plus simplement, un tunnel SSL sert à encapsuler du trafic IP afin d'en sécuriser l'échange entre deux points. OpenVPN est très intéressant pour les sites qui souhaitent déployer rapidement et simplement un accès sécurisé VPN.

- http://www.openvpn.net

En résumé…

Un bon nombre de protocoles réseau encore régulièrement utilisés de nos jours ont été développés à une époque où les réseaux locaux et distants étaient considérés comme dignes de confiance. En admettant que cela ait été réellement le cas un jour, ce n'est plus vrai aujourd'hui.

Il est nécessaire de protéger les données véhiculées sur le réseau et de renforcer les mécanismes d'authentification. Après avoir subi une compromission, l'équipe en charge de la gestion du réseau informatique de Tamalo.com a décidé de supprimer l'ensemble des services laissant transiter des données sensibles (notamment les mots de passe) en clair sur le réseau. Ainsi, l'arrêt définitif des services TELNET, FTP et RSH au profit de SSH a été appliqué immédiatement après la réorganisation de l'infrastructure informatique. Il en est de même avec les services HTTP et IMAP, qui sont remplacés chaque fois que cela est nécessaire par les versions SSL de ces mêmes protocoles, introduisant ainsi une couche de transport chiffrée et des mécanismes d'authentification forte.

chapitre 5

Sécurisation des systèmes

Maintenir un ensemble de machines Linux dans un état homogène et cohérent, c'est-à-dire procéder à une administration système rigoureuse, est la condition préalable strictement nécessaire à la sécurisation du système d'information. Elle commence par le déploiement d'un système minimal opérationnel.

SOMMAIRE

- Installation automatisée du système
- Mise en configuration minimale
- Mises à jour
- Sécurisation du système de fichiers
- Configuration sécurisée de la pile TCP/IP

MOTS-CLÉS

- Administration système
- Bonnes pratiques
- KickStart
- BIOS, RPM
- Chkconfig
- Service
- Protection des fichiers
- suid, sgid

Nous allons présenter au cours de ce chapitre les moyens mis en œuvre par les administrateurs de la société Tamalo.com pour maintenir un ensemble de machines Linux dans un état homogène et cohérent. Des opérations aussi simples que la mise à jour régulière des composants du système, la désactivation systématique des services inutiles ou encore la gestion contrôlée des accès aux différentes ressources du système, permettent d'avancer significativement dans le domaine de la sécurité système.

La première phase de la procédure d'installation consiste à déployer un système minimal opérationnel sécurisé, de manière homogène, sur l'ensemble du parc informatique. Nous verrons comment, avec les outils fournis avec la distribution Red Hat, il est possible d'automatiser l'installation des machines, pour limiter l'hétérogénéité des systèmes d'exploitation. Aucun service superflu, réseau ou non, ne doit y être démarré. C'est en général loin d'être le cas lors d'une installation par défaut de la plupart des distributions Linux. Une seconde phase de configuration proportionnée aux besoins en sécurité permet la mise en conformité opérationnelle définitive du système.

Nous allons aborder ces différentes étapes au cours de ce chapitre.

Installation automatisée

Afin de simplifier l'administration du parc, deux profils de machines ont été définis. Le premier concerne les postes de travail à destination des employés, qu'ils soient administratifs, commerciaux ou encore développeurs. Ils seront dotés de PC avec le système Linux Red Hat et les outils nécessaires au travail bureautique et au développement d'applications.

Le deuxième profil concerne les serveurs sur lesquels sont hébergées les applications nécessaires au travail collaboratif (impression, bases de données, messagerie électronique) et aux services Internet (serveur de noms, serveur Web). Un système Linux Red Hat beaucoup plus dépouillé que le premier sert de base à ces serveurs d'applications.

L'objectif de cette vision simplifiée du parc informatique est clairement d'administrer le plus efficacement possible des systèmes très similaires, afin de permettre une plus grande réactivité en cas de besoin (déploiement de mises à jour, incident de sécurité). Un environnement système homogène permettra l'utilisation de procédures automatiques nécessaires aux différentes tâches d'administration des systèmes. Ne l'oublions pas, un système sécurisé est d'abord un système bien maintenu et donc à jour.

L'utilisation de KickStart est un atout pour le déploiement des profils définis par les administrateurs système de Tamalo.com. Le contenu des deux fichiers de profils est présenté ci-dessous. Ils diffèrent principalement dans le nombre de logiciels installés. Les postes de travail sont pourvus d'un envi-

Outils KickStart

KickStart est un outil proposé avec les distributions Linux Red Hat. Il permet l'automatisation des phases d'installation du système et de ses composants. L'installation KickStart se réalise de manière automatique grâce à un fichier de configuration dans lequel est décrit le profil de la machine à installer. Le fichier `/root/anaconda-ks.cfg` contenant le profil KickStart est créé lors de chaque installation d'un système LinuxRed Hat. Après une installation manuelle témoin, il peut être utilisé pour cloner le système.

- http://www.redhat.com/docs/manuals/linux/RHL–7.3–Manual/custom–guide/ch–kickstart2.html

ronnement graphique, d'outils de développement et d'outils de productivité. Aucun compilateur n'est installé sur les serveurs d'applications afin de limiter le champ d'action d'un pirate qui aurait réussi à s'introduire sur les machines. Le niveau de sécurité du pare-feu réseau est par défaut maximal. Voici les profils serveur et poste de travail.

Fichier KickStart du profil serveur

```
#####################################################
# fichier KickStart - profil serveur #
#####################################################
# Kickstart file automatically generated by anaconda.
install
lang en_US
langsupport --default en_US.UTF-8 en_US.UTF-8
keyboard fr
mouse generic3ps/2 --device psaux
skipx
network --device eth0 --bootproto static --ip 192.168.1.99 --netmask 255.255.255.0
➥ --gateway 192.168.1.254 --nameserver 192.168.1.11 --hostname mynewserver
rootpw --iscrypted $1$ùªäÏÔïÍÎ$UGLO4pbdH6PvYq2icErHi/
firewall --high --dhcp
authconfig --enableshadow --enablemd5
timezone Europe/Paris
bootloader --location=mbr --md5pass=$1$AUôBºµ9Þ$s.41wZbdILEw/cm.zu9Z3/
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
clearpart --all
part /boot --fstype ext3 --size=128
part /tmp --fstype ext3 --size=128
part swap --size=256
part / --fstype ext3 --size=1 --grow
%packages
%post
```

Fichier KickStart du profil des postes de travail

```
#####################################################
# fichier KickStart - profil poste de travail #
#####################################################
# Kickstart file automatically generated by anaconda.
install
lang en_US
langsupport --default en_US.UTF-8 en_US.UTF-8
keyboard fr
mouse generic3ps/2 --device psaux
skipx
network --device eth0 --bootproto dhcp
rootpw --iscrypted $1$ýEöó8æÌK$AvsTl13hnwcEuBGMSK8WOO
firewall --high
authconfig --enableshadow --enablemd5
```

```
timezone --utc Europe/Paris
bootloader --location=mbr --md5pass=$1$Ä14uÑøöð$g9r/
➥6myWayVYJHqMxjUql.
# The following is the partition information you requested
# Note that any partitions you deleted are not
expressed
# here so unless you clear all partitions first,
# this is not guaranteed to work
clearpart -linux
part /boot --fstype ext3 --size=128
part swap --size=256
part /tmp --fstype ext3 --size=128
part / --fstype ext3 --size=1 --grow
%packages
@ Development Tools
@ GNOME Desktop Environment
@ Graphical Internet
@ Graphics
@ Kernel Development
@ Office/Productivity
@ Sound and Video
@ Text-based Internet
@ X Window System
gnome-audio
tk
xmms
firstboot
ggv
gtkam
ltrace
xsane-gimp
cdparanoia
indent
xsane
diffstat
gtk-engines
gnome-vfs-extras
mrproject
authconfig-gtk
gnome-media
ImageMagick
fetchmail
automake14
automake15
gdm
gnome-system-monitor
cdp
patchutils
gimp-print-utils
openssh-askpass
xawtv
.../...
.../...
xchat
memprof
magicdev
gcc-g77
sane-frontends
file-roller
hpijs
dia
xpdf
mozilla-psm
redhat-config-services
gcc-gnat
mutt
splint
openssh-askpass-gnome
openoffice
redhat-switchmail-gnome
XFree86-xdm
gftp
gnome-vfs2-extras
hwbrowser
evolution
libvorbis-devel
gtk2-engines
redhat-config-network
mtr-gtk
doxygen
netpbm-progs
cvs
gcc-java
gnome-user-docs
redhat-config-xfree86
grip
redhat-config-users
rcs
redhat-switch-printer-gnome
gaim
gtoaster
gimp-data-extras
cdlabelgen
%post
redhat-config-printer-gui
gedit
gconf-editor
gqview
slrn
redhat-logviewer
rhn-applet
desktop-backgrounds-extra
```

Copié sur une disquette sous le nom de `ks.cfg`, le fichier de profil est utilisé lors de l'invocation de la commande `linux ks=floppy` lors de la phase d'installation du système, comme le montre la figure 5-1.

```
-  To install or upgrade Red Hat Linux in graphical mode,
   press the <ENTER> key.

-  To install or upgrade Red Hat Linux in text mode, type:
   linux text <ENTER>.

-  Use the function keys listed below for more information.

[F1-Main] [F2-Options] [F3-General] [F4-Kernel] [F5-Rescue]
boot: linux ks=floppy_
```

Figure 5–1 Lancement d'une installation utilisant le profil ks.cfg copié sur une disquette

RECOMMANDATIONS **Mises à jour...**

La mise à jour doit être systématique et immédiate quand un problème de sécurité a été détecté dans un composant du système.
Pour les produits critiques comme les applications réseau (DNS, WEB, MAIL…), il est fortement recommandé d'utiliser les distributions originales plutôt que les paquetages fournis dans la distribution Linux Red Hat. Ceci permet notamment une plus grande réactivité quand une nouvelle version de tel ou tel logiciel est disponible (et cela évite d'attendre que Red Hat en ait fait un paquetage).

Mise à jour régulière des systèmes

Les éditeurs de logiciels ne fournissent plus aujourd'hui les supports d'installation (CD ou DVD) contenant les dernières mises à jour du système. Il incombe donc à l'administrateur de compléter son installation en téléchargeant à mesure les plus récentes mises à jour de chacun des composants du système d'exploitation et de les appliquer.

En effet, une mise à jour après installation est indispensable, comme le sont celles qui vont suivre tout au long de la vie du système. Cela ne doit pas être négligé, la sécurité en dépend et un retard dans l'application d'un correctif est très souvent à l'origine de la compromission de machine, comme dans le cas de la société Tamalo.com (chapitre 3).

Concernant la distribution Red Hat, les mises à jour sont téléchargeables à partir du site ftp.redhat.com ou à partir de tous ses sites miroirs officiels (répertoire `/pub/redhat/linux/update/`). Le format dans lequel elles sont distribuées correspond au format de paquetage RPM. Il suffit de copier les fichiers contenant ces paquetages dans un répertoire puis d'exécuter la commande suivante pour appliquer l'ensemble des correctifs :

```
# rpm -Fvh *.rpm
```

RAPPEL IMPORTANT

Il convient de s'assurer de la provenance et de l'intégrité des paquetages par exemple en vérifiant leurs signatures PGP avant installation :
`rpm -K <nom_du_paquetage.rpm>`

Les administrateurs de Tamalo.com ont pris le temps de développer un outil simple, effectuant la comparaison des paquetages installés sur un système

avec les mises à jour disponibles sur un site miroir FTP Red Hat. Cet outil permet le téléchargement et l'installation des nouvelles versions des logiciels.

L'ensemble APT a été adapté de façon à gérer les rpm (la distribution Debian utilise un autre format de paquetage appelé .deb) par Connectiva, éditeur d'une distribution Linux.
- http://freshrpms.net/apt/

Mise à jour et installation optimale avec APT

La série de programmes nommés APT, issue de Debian et fort justement réputée, simplifie considérablement l'installation et la mise à jour de paquetages. Elle gère entre autres les dépendances, les accès à de nombreux sites abritant les versions à jour, et la vérification de leurs signatures.

Nous insistons cependant : il ne faut *pas* penser a priori que des rpm publiés de provenance non déterminée, où qu'ils se trouvent, soient fiables ni sûrs.

Présentation commerciale de Red Hat Network :
- http://www.europe.redhat.com/software/rhen/

Mise à jour avec Red Hat Network

Red Hat propose un service de mise à jour nommées Red Hat Network grâce à des paquetages certifiés effectuée quasi automatiquement et à distance, flanquée d'une modeste console d'administration.

L'indispensable protection par mot de passe au démarrage

Périphériques d'amorçage

Les paramètres d'amorçage, définis dans la configuration SETUP d'un PC, incluent en général par ordre de recherche, le lecteur de disquettes, le lecteur de CD-Rom et enfin le (ou les) disque(s) installé(s) dans l'unité. Pour un système stable et opérationnel, il est recommandé de ne laisser que le disque contenant le système d'exploitation dans la liste des périphériques de démarrage. Cela complique la tâche d'un pirate accédant directement à la machine afin de l'amorcer grâce à son propre support (CD, disquette, clé USB...) pour accéder aux fichiers qu'elle stocke. Le verrouillage de l'unité centrale, interdisant de déconnecter ou de décharger (par court-circuit) la batterie préservant le SETUP offrira une protection complémentaire utile.

Il a été décidé de limiter au maximum la possibilité d'utiliser le matériel de manière non conforme aux objectifs initialement définis. Un ajout de périphérique, une modification de configuration non validés par un administrateur système peuvent amener à des situations où le risque non correctement évalué met en péril le système informatique. On peut protéger simplement le système, en cumulant les différentes possibilités de restrictions d'accès, aussi bien au niveau du BIOS (figure 5-2), du gestionnaire d'amorçage, `boot`

Des informations utiles sur la parution de mises à jour des paquetages Red Hat sont sur le site de Red Hat.
- http://www.redhat.com/apps/support/errata/index.html.

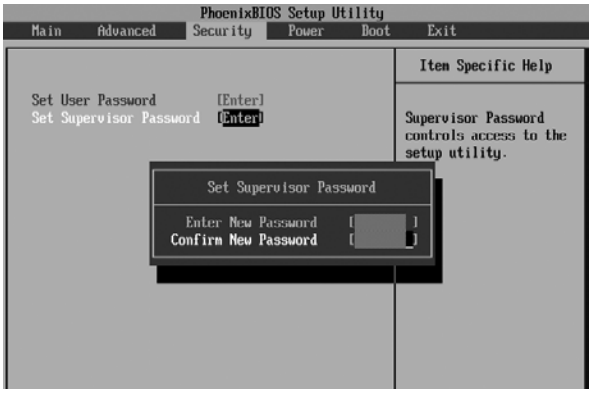


Figure 5–2 Définition d'un mot de passe pour l'accès au panneau de configuration du BIOS

`loader` en anglais (figure 5-3), que du système d'exploitation (figure 5-4). Cette restriction se fait le plus souvent par un mot de passe que seul l'administrateur système doit détenir.

Il convient de ne négliger aucune de ces dispositions car, à défaut, tout individu mal intentionné parvenant à amorcer le système comme il le souhaite pourrait sans mal s'en rendre maître.

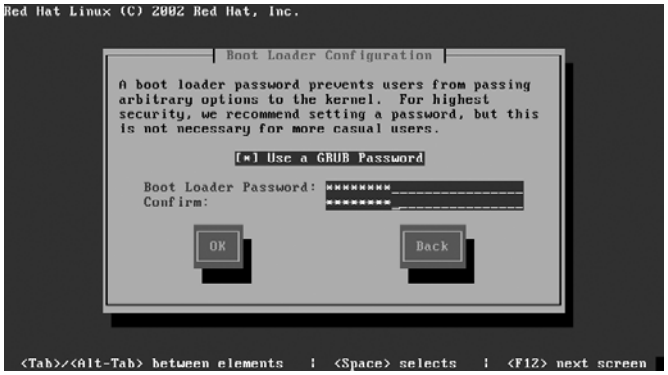


Figure 5–3 Définition d'un mot de passe pour interdire la modification des options par défaut du gestionnaire d'amorçage

Figure 5–4 Définition du mot de passe du compte administrateur « root » du système d'exploitation lors de l'installation manuelle du système

Mise en configuration minimale, limitation des services actifs

Les administrateurs système de Tamalo.com se sont aperçus à leurs dépens que des services inutiles étaient démarrés automatiquement après l'installation. Ainsi, sur bon nombre de machines, le service d'impression distant LPRng était actif bien que non utilisé.

Il y a deux avantages à désactiver systématiquement les services inutiles. Cela augmente d'une part la sécurité potentielle du système : autant de services en moins sont autant de vulnérabilités potentielles en moins. D'autre part, cela permet de libérer quelques mégaoctets de mémoire et quelques cycles de processeur, ce qui est toujours bon à prendre.

La mise en configuration minimale d'une machine installée avec un profil serveur est présentée dans ce qui suit. Sur les serveurs de Tamalo.com, seul le service réseau SSH, ainsi que quelques composants indispensables du système d'exploitation, restent activés à l'issue de cette mise en configuration minimale.

À RETENIR

Tout service étant potentiellement dangereux, il va de soi qu'il faut supprimer ceux qui sont inutiles.

Identification des processus

La liste des processus en cours d'exécution sur le système immédiatement après l'installation peut paraître raisonnable ; pourtant, plusieurs sont inutiles.

Liste des processus exécutés par défaut après installation

```
# ps -edf
UID        PID  PPID C STIME TTY    TIME     CMD
root         1     0 0 14:13 ?      00:00:04 init
root         2     1 0 14:13 ?      00:00:00 [keventd]
root         3     1 0 14:13 ?      00:00:00 [kapmd]
root         4     1 0 14:13 ?      00:00:00 [ksoftirqd_CPU0]
root         5     1 0 14:13 ?      00:00:00 [kswapd]
root         6     1 0 14:13 ?      00:00:00 [bdflush]
root         7     1 0 14:13 ?      00:00:00 [kupdated]
root         8     1 0 14:13 ?      00:00:00 [mdrecoveryd]
root        16     1 0 14:13 ?      00:00:02 [kjournald]
root        71     1 0 14:13 ?      00:00:00 [khubd]
root       164     1 0 14:14 ?      00:00:00 [kjournald]
root       165     1 0 14:14 ?      00:00:00 [kjournald]
root       463     1 0 14:14 ?      00:00:00 syslogd -m 0
root       468     1 0 14:14 ?      00:00:00 klogd -x
rpc        485     1 0 14:14 ?      00:00:00 portmap
rpcuser    504     1 0 14:14 ?      00:00:00 rpc.statd
root       584     1 0 14:15 ?      00:00:00 /usr/sbin/apmd -p 10 -w 5 -W -P
root       634     1 0 14:15 ?      00:00:00 /usr/sbin/sshd
root       657     1 0 14:15 ?      00:00:01 sendmail: accepting connections
smmsp      667     1 0 14:15 ?      00:00:00 sendmail: Queue runner@01:00:00
root       677     1 0 14:15 ?      00:00:00 gpm -t ps/2 -m /dev/mouse
root       686     1 0 14:15 ?      00:00:00 crond
root       695     1 0 14:15 ?      00:00:00 anacron -s
daemon     704     1 0 14:15 ?      00:00:00 /usr/sbin/atd
root       713     1 0 14:15 ?      00:00:00 login -- root
root       714     1 0 14:15 tty2   00:00:00 /sbin/mingetty tty2
root       715     1 0 14:15 tty3   00:00:00 /sbin/mingetty tty3
root       716     1 0 14:15 tty4   00:00:00 /sbin/mingetty tty4
root       717     1 0 14:15 tty5   00:00:00 /sbin/mingetty tty5
root       718     1 0 14:15 tty6   00:00:00 /sbin/mingetty tty6
root       721   713 0 14:15 tty1   00:00:01 -bash
root       813   721 0 14:53 tty1   00:00:00 ps -edf
```

Identification des ports réseau utilisés

À l'exception du port 22 utilisé par le service d'accès sécurisé SSH, aucun autre port ne devrait être utilisé. En effet, aucun autre service réseau n'est supposé être démarré. Ce n'est pas le cas, bien que l'installation soit minimale d'après le fichier de configuration de KickStart.

```
# netstat -atup
Active Internet connections (servers and established)
```

```
Proto Recv-Q Send-Q Local Address    Foreign Address   State     PID/Program name
tcp        0      0 *:1024           *:*               LISTEN    504/rpc.statd
tcp        0      0 *:sunrpc         *:*               LISTEN    485/portmap
tcp        0      0 *:ssh            *:*               LISTEN    634/sshd
tcp        0      0 mynewserver:smtp *:*               LISTEN    657/sendmail
udp        0      0 *:1024           *:*                         504/rpc.statd
udp        0      0 *:sunrpc         *:*                         485/portmap
```

Les services ouverts sont les suivants :

- `rpc.statd`, RPC Network Status Monitor (utilisé par NFS) ;
- `portmap`, qui fournit la correspondance entre le numéro RPC (Remote Procedure Call) et le port IP utilisé en écoute par le service RPC ;
- `sshd`, service de connexion distante sécurisé ;
- `sendmail`, service de messagerie électronique.

Identification des services actifs

Même les services qui ne sont pas des services réseau – et sont donc moins vulnérables à une attaque extérieure – sont bien trop nombreux sur ce système fraîchement installé. Le résultat de la commande `chkconfig` donne l'état de chacun d'eux pour les niveaux d'exécution du système d'exploitation.

Liste des services réseau avec chkconfig

```
# chkconfig --list
syslog        0:off 1:off 2:on  3:on  4:on  5:on  6:off
netfs         0:off 1:off 2:off 3:on  4:on  5:on  6:off
network       0:off 1:off 2:on  3:on  4:on  5:on  6:off
random        0:off 1:off 2:on  3:on  4:on  5:on  6:off
rawdevices    0:off 1:off 2:off 3:on  4:on  5:on  6:off
saslauthd     0:off 1:off 2:off 3:off 4:off 5:off 6:off
portmap       0:off 1:off 2:off 3:on  4:on  5:on  6:off
apmd          0:off 1:off 2:on  3:on  4:on  5:on  6:off
atd           0:off 1:off 2:off 3:on  4:on  5:on  6:off
gpm           0:off 1:off 2:on  3:on  4:on  5:on  6:off
autofs        0:off 1:off 2:off 3:on  4:on  5:on  6:off
irda          0:off 1:off 2:off 3:off 4:off 5:off 6:off
isdn          0:off 1:off 2:on  3:on  4:on  5:on  6:off
keytable      0:off 1:on  2:on  3:on  4:on  5:on  6:off
kudzu         0:off 1:off 2:off 3:on  4:on  5:on  6:off
sshd          0:off 1:off 2:on  3:on  4:on  5:on  6:off
snmpd         0:off 1:off 2:off 3:off 4:off 5:off 6:off
snmptrapd     0:off 1:off 2:off 3:off 4:off 5:off 6:off
sendmail      0:off 1:off 2:on  3:on  4:on  5:on  6:off
iptables      0:off 1:off 2:on  3:on  4:on  5:on  6:off
nfs           0:off 1:off 2:off 3:off 4:off 5:off 6:off
nfslock       0:off 1:off 2:off 3:on  4:on  5:on  6:off
rhnsd         0:off 1:off 2:off 3:on  4:on  5:on  6:off
pcmcia        0:off 1:off 2:on  3:on  4:on  5:on  6:off
crond         0:off 1:off 2:on  3:on  4:on  5:on  6:off
anacron       0:off 1:off 2:on  3:on  4:on  5:on  6:off
```

B.A.-BA Niveaux d'exécution du système Linux

1. Niveau 0 : arrêt du système d'exploitation
2. Niveau 1 : mode de maintenance mono-utilisateur
3. Niveaux 2 et 3 : modes multi-utilisateurs
4. Niveau 4 : non utilisé
5. Niveau 5 : mode multi-utilisateur et affichage graphique
6. Niveau 6 : redémarrage du système

Un poste de travail est normalement en niveau 5 d'exécution du système alors qu'un serveur est au niveau 3. Le niveau de démarrage du système est configurable dans le fichier `/etc/inittab`.

ALTERNATIVE Interface de configuration serviceconf

Sous Red Hat 9 et sous X Window, on invoquera `serviceconf`, interface extrêmement conviviale de configuration des services : démarrage, arrêt, affectation à un niveau d'exécution donné. `Serviceconf` offre en un coup d'œil un panorama des services, et une description détaillée pour chacun.

Suppression du MTA déconseillée sur les serveurs

Dans la plupart des cas, il ne sera pas possible ni souhaitable de supprimer le routeur de messagerie (MTA, ici sendmail) car il véhicule les messages électroniques expédiés par divers programmes (en particulier les travaux sous `cron`) à l'administrateur du système. Pour prévenir une attaque utilisateur depuis un compte interactif (shell), on désactivera les comptes interactifs. Pour prévenir les attaques venant du réseau, on limitera l'exposition grâce à un filtre de paquets IP (voir chapitre 7, « Filtrage en entrée de site ») et par une configuration adéquate de Sendmail.

Désactivation des services inutiles

Sur nos machines, seuls les services `syslog`, `network`, `random`, `rawdevices`, `keytable`, `kudzu`, `sshd`, `iptables` et `crond` doivent rester actifs. Ces services dépendent bien sûr de vos besoins. Par exemple, il sera recommandé de réactiver le service de gestion d'alimentation électrique `apmd` sur un portable.

Les services réseau `rpc.statd`, `autofs` et `portmap` sont désactivés, ainsi que les autres services indésirables, en utilisant la procédure suivante :

1. Désactivation au redémarrage du système d'exploitation :

```
# chkconfig--level 0123456 autofs off
```

2. Suppression du service de la table de configuration :

```
# chkconfig --del autofs
```

3. Arrêt immédiat du service :

```
# service autofs stop
```

Cette opération est réalisée pour les services `anacron`, `pcmcia` (sauf dans le cas d'un portable), `rhnsd`, `nfslock`, `nfs`, `snmpd`, `snmptrapd`, `isdn`, `irda` (sauf en cas d'utilisation d'un port infrarouge), `autofs`, `gpm`, `portmap`, `saslauthd` et `netfs`.

Certains des services pourront être réactivés lors de la phase de configuration du serveur, quand le rôle de ce dernier aura été défini. Nous aborderons cette partie au cours du chapitre 6, « Sécurisation des services ».

Ci-dessous, les résultats des commandes `ps`, `netstat` et `chkconfig` donnent l'état du système après cette phase de désactivation des services inutiles.

Ports réseau utilisés après mise en configuration minimale (seul le port réseau 22 correspondant au service SSH est ouvert)

```
# netstat -atup
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp        0      0 *:ssh         *:*             LISTEN 634/sshd
```

Processus en cours d'exécution après mise en configuration minimale

```
# ps -edf
UID        PID  PPID C STIME TTY    TIME     CMD
root         1     0 0 14:13 ?      00:00:04 init
root         2     1 0 14:13 ?      00:00:00 [keventd]
root         3     1 0 14:13 ?      00:00:00 [kapmd]
root         4     1 0 14:13 ?      00:00:00 [ksoftirqd_CPU0]
root         5     1 0 14:13 ?      00:00:00 [kswapd]
root         6     1 0 14:13 ?      00:00:00 [bdflush]
root         7     1 0 14:13 ?      00:00:00 [kupdated]
root         8     1 0 14:13 ?      00:00:00 [mdrecoveryd]
```

```
root        16     1 0 14:13 ?      00:00:02 [kjournald]
root        71     1 0 14:13 ?      00:00:00 [khubd]
root       164     1 0 14:14 ?      00:00:00 [kjournald]
root       165     1 0 14:14 ?      00:00:00 [kjournald]
root       463     1 0 14:14 ?      00:00:00 syslogd -m 0
root       468     1 0 14:14 ?      00:00:00 klogd -x
root       634     1 0 14:15 ?      00:00:00 /usr/sbin/sshd
root       686     1 0 14:15 ?      00:00:00 crond
root       713     1 0 14:15 ?      00:00:00 login -- root
root       714     1 0 14:15 tty2   00:00:00 /sbin/mingetty tty2
root       715     1 0 14:15 tty3   00:00:00 /sbin/mingetty tty3
root       716     1 0 14:15 tty4   00:00:00 /sbin/mingetty tty4
root       717     1 0 14:15 tty5   00:00:00 /sbin/mingetty tty5
root       718     1 0 14:15 tty6   00:00:00 /sbin/mingetty tty6
```

État des différents services après mise en configuration minimale

```
# chkconfig --list
syslog        0:off 1:off 2:on  3:on  4:on  5:on  6:off
network       0:off 1:off 2:on  3:on  4:on  5:on  6:off
random        0:off 1:off 2:on  3:on  4:on  5:on  6:off
rawdevices    0:off 1:off 2:off 3:on  4:on  5:on  6:off
keytable      0:off 1:on  2:on  3:on  4:on  5:on  6:off
kudzu         0:off 1:off 2:off 3:on  4:on  5:on  6:off
sshd          0:off 1:off 2:on  3:on  4:on  5:on  6:off
iptables      0:off 1:off 2:on  3:on  4:on  5:on  6:off
crond         0:off 1:off 2:on  3:on  4:on  5:on  6:off
```

Pour un profil de poste de travail, un service supplémentaire apparaît dans ce tableau. Il s'agit de `xfs`, le serveur de polices de caractères nécessaire au fonctionnement de X Window.

Sécurisation du système de fichiers

Permissions des fichiers

Les droits positionnés sur l'ensemble des fichiers et des répertoires sont parfois trop permissifs. L'accès à une information ou à une commande sensible peut mettre le système en péril lors d'une attaque. L'utilisation excessive des droits de protection « suid » et, dans une moindre mesure, celle du « sgid » constituent également une cause de vulnérabilité importante. L'exploitation combinée de ces permissions particulières et de la mauvaise implémentation de certains programmes, comme la possibilité de débordement de mémoire, est à l'origine de bon nombre de compromissions.

ALLER PLUS LOIN

Systèmes de fichiers chiffrés

Pour celui qui considère la confidentialité de ses données comme très critique, il est possible d'utiliser des moyens de stockage sécurisé qui offrent une bonne protection contre le vol de matériel ou l'accès frauduleux. Un éventail d'outils permet le chiffrement du système de fichiers. Attention néanmoins, en cas de problème, la réparation d'un tel système de fichiers peut s'avérer parfois longue et difficile.

- http://encryptionhowto.sourceforge.net/
- http://tldp.org/HOWTO/Cryptoloop-HOWTO/
- http://koeln.ccc.de/archiv/drt/crypto/linux-disk.html

B.A.-BA Permissions Unix

Les protections Unix positionnées sur un fichier sont composées de trois triplets qui correspondent aux permissions données au propriétaire du fichier (triplet u pour *user*), au groupe propriétaire (triplet g pour *group*) et au reste du monde (triplet o pour *others*). Chaque triplet donne ou non l'autorisation de lire (droit r), d'écrire (droit w) et d'exécuter (droit x) le fichier.

B.A.-BA Droits spéciaux suid et sgid des exécutables

Ces droits particuliers (droits s) concernent uniquement le bit d'exécution. Ils définissent l'identité sous laquelle l'exécution d'un programme sera réalisée. Ceci ne vaut que pour les binaires et non pour les scripts. Si aucun des deux droits n'est positionné, le programme est exécuté sous l'identité de l'utilisateur qui l'a lancé. Cette identité est définie par un identificateur d'utilisateur (uid) et un identificateur de groupe (gid). Si à l'inverse le droit suid est positionné, le programme est exécuté sous une identité composée du uid propriétaire du fichier et du gid de l'utilisateur appelant le programme.
L'utilisation du sgid permet l'exécution sous le gid du groupe propriétaire du programme. Un utilisateur non privilégié peut par cette technique accéder à des informations sensibles, lisibles et modifiables en théorie seulement par l'administrateur du système. La commande passwd permettant de changer le mot de passe d'un compte est dotée du droit suid parce qu'elle doit écrire dans les fichiers abritant ces informations où seul root peut écrire (/etc/passwd, etc.).

```
# ls -l /usr/bin/passwd
-r-s--x--x 1 root root 15368 May
28 2002 /usr/bin/passwd
```

Détection des fichiers dotés de droits trop permissifs

Voici un ensemble de commandes permettant de détecter les fichiers pour lesquels les protections pourraient être considérées comme trop permissives.

- Fichiers en écriture pour tous

```
# find / -type f -perm -o+w -ls
```

- Fichiers en écriture pour le groupe

```
# find / -type f -perm -g+w -ls
```

- Répertoires en écriture pour tous

```
# find / -type d -perm -o+w -ls
```

- Répertoires en écriture pour le groupe

```
# find / -type d -perm -g+w -ls
```

Notons que tous les utilisateurs doivent pouvoir écrire dans les répertoires (communs) /tmp et /var/tmp avec la restriction que seul le propriétaire d'un fichier peut le modifier et le supprimer. Ce droit particulier est appelé *sticky bit* (droit t) et doit être impérativement attribué à de tels répertoires comme indiqué ci-dessous :

```
# ls -ld /tmp
drwxrwxrwt10root root4096Jun 24 17:33/tmp
# ls -ld /var/tmp
drwxrwxrwt10root root4096Jun 24 12:40/var/tmp
```

Droits suid et sgid

De nombreuses attaques détournent l'utilisation des droits particuliers suid et sgid positionnés sur des fichiers de programmes pour obtenir des accès privilégiés aux ressources du système. Il est impératif de contrôler l'utilisation de ces droits afin de garantir leur utilisation dans les conditions pour lesquelles elles ont été prévues. Cette mission incombe au programme qui bénéficie du suid ou sgid ; celui-ci ne doit pas, par conséquent, présenter de faille.

Il faut limiter au maximum l'utilisation du droit suid lorsque le fichier appartient à un compte privilégié comme le compte administrateur root. Seuls quelques programmes devraient en être dotés.

Recherche des fichiers dotés du droit « suid » :

```
# find / -perm -4000 -type f -ls
```

Recherche des fichiers dotés du droit « sgid » :

```
# find / -perm -2000 -type f -ls
```

Alternative à la protection suid : sudo

Le programme `sudo`, disponible dans toutes les bonnes distributions Linux, est une excellente alternative au positionnement de la protection `suid` sur les programmes. En effet, `sudo` permet de déléguer le droit à un utilisateur sur une machine d'exécuter une commande sous une autre identité, y compris celle de l'administrateur de la machine. Le programme `sudo` est lui-même doté du droit `suid`. Son utilisation permet de restreindre le nombre de commandes possédant cette protection spéciale et d'en limiter les accès.

Le programme `sudo` est très facile à configurer. La commande `visudo` permet notamment d'éditer le fichier de configuration `/etc/sudoers` en étant certain que personne d'autre ne l'édite au même moment. Le fichier `sudoers` donné ci-dessous autorise l'utilisateur `bernard` à exécuter la commande `/usr/sbin/xcdroast` avec les droits `root` sur la machine `rh2.tamalo.com` sans avoir à fournir de mot de passe :

```
# sudoers file.
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the sudoers man page for the details on how to write a sudoers
file.
#
# Host alias specification
# User alias specification
# Cmnd alias specification
# Defaults specification
# User privilege specification
root ALL=(ALL) ALL
# Uncomment to allow people in group wheel to run all commands
# %wheel ALL=(ALL)ALL
# Same thing without a password
# %wheel ALL=(ALL)NOPASSWD: ALL
# Samples
# %users ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users localhost=/sbin/shutdown -h now
# bernard est autorisé à utiliser le graveur de CD sur rh2
bernard rh2.tamalo.com = NOPASSWD: /usr/sbin/xcdroast
```

Lorsque que `bernard` veut graver un CD, il appelle `xcdroast` avec la commande suivante :

```
# sudo /usr/sbin/xcdroast
```

Options de montage des systèmes de fichiers

Pour interdire l'utilisation des programmes dotés de l'attribut de protection `suid`, il est possible de positionner l'option `nosuid` au montage d'un système de fichiers. Cela a pour effet de bloquer l'interprétation par le système de cette protection spéciale. Le programme pourvu du droit `suid` sur un tel système de fichiers sera exécuté sous l'identité de l'utilisateur appelant, et non pas sous celle du propriétaire du fichier.

Les pirates ont souvent pour objectif de lancer un *shell* (interpréteur de commandes) appartenant à l'utilisateur `root` et pourvu du droit `suid`. Le montage du système de fichiers avec l'option `nosuid` permet de contrer ce genre d'initiative malveillante.

Dans le cas des machines de la société Tamalo.com, le montage du système de fichiers `/tmp` et les systèmes de fichiers contenant des données utilisateurs est effectué avec l'option `nosuid`. Le contenu de `/etc/fstab` est modifié en conséquence :

- Les options de montage du système de fichiers `/tmp` sont celles par défaut :

```
LABEL=/tmp /tmp ext3 defaults 1 2
```

- Il faut modifier les attributs de montage de `/tmp` de manière à inclure l'option `nosuid` :

```
LABEL=/tmp /tmp ext3 nosuid 1 2
```

Gestion des accès et stratégie locale de sécurité

La gestion des accès aux différentes ressources du système est capitale. Avec une installation par défaut, tout est permis ou presque. Une bonne politique est au contraire de systématiquement limiter les accès au maximum, pour ensuite ouvrir l'utilisation des services lorsque les besoins s'en font sentir.

Compte privilégié root

Le compte `root` est le compte administrateur par défaut sur le système Linux. Les administrateurs système doivent lui accorder la plus grande attention car sa compromission peut avoir des conséquences graves. Quelques précautions simples permettent d'augmenter la sécurité d'utilisation du compte `root`.

Le répertoire personnel `/root` (*home directory* en anglais) ne doit pas être accessible en écriture ni en lecture pour toute autre personne que `root` lui-même.

BON À SAVOIR Limitation de l'accès privilégié avec securetty

Le fichier `/etc/securetty` définit la liste des pseudo-terminaux via lesquels l'utilisateur `root` peut se connecter. En situation de très haute sécurité, `root` ne devrait pouvoir se connecter qu'à partir de la console du système et non à partir du réseau. Cela permet d'éviter de faire transiter sur le réseau une information aussi critique que le mot de passe de l'administrateur du système.

Le masque de protection pour la création des fichiers ne doit permettre qu'à l'administrateur de modifier ses propres fichiers (`umask 022` ou mieux `077`). Enfin, le contenu de la variable `PATH` définie dans le fichier `/root/.bash_profile` est volontairement limité aux répertoires `/bin`, `/usr/bin`, `/sbin` et `/usr/sbin`.

Blocage des comptes inutiles

À l'installation du système, de nombreux comptes sont créés. Ils correspondent pour la plupart à des comptes d'applications et n'ont pas pour vocation d'être utilisés de façon interactive. À ce stade, seul le compte administrateur `root` est un compte de connexion. Le `shell` de tous les autres comptes doit être positionné à `/sbin/nologin` (sous Red Hat) ou `/bin/false` (sous Debian).

Filtrage réseau avec TCP Wrapper

TCP Wrapper est une bibliothèque permettant de filtrer les connexions réseau au niveau applicatif. Elle est intégrée à la plupart des distributions Linux et la majorité des services réseau utilisent ses fonctionnalités. Par exemple, il est possible de filtrer les connexions à destination du serveur SSH de la machine en modifiant le contenu des fichiers `/etc/hosts.deny` et `/etc/hosts.allow`. Les administrateurs système de Tamalo.com estiment qu'après la phase d'installation, aucune connexion réseau ne doit être autorisée par défaut. Les mots-clés `ALL :ALL` sont donc ajoutés au fichier `/etc/hosts.deny`. On refuse donc les connexions à destination de tous (`ALL:`) les services de la machine en provenance de toutes (`:ALL`) les machines. Pour rendre de nouveau accessibles les services réseau utilisant TCP Wrapper, il sera nécessaire d'intervenir sur le fichier `/etc/hosts.allow`.

L'activité de TCP Wrapper est envoyée dans le système de journal syslog.

Fichier /etc/hosts.deny sur l'ensemble des machines

```
# Interdit toutes les connexions par défaut et envoie un
# courrier électronique d'alerte lors d'une tentative de
# connexion refusée.
ALL:ALL spawn (echo "%s refuses connection from %c" | /bin/mail
-s "tcpd alert" hostmaster@tamalo.com) &
```

Fichier /etc/hosts.allow sur l'ensembles des serveurs

```
# Autorise les connexions SSH en provenance des
# réseaux internes de Tamalo.com et de l'adresse de
# bouclage interne.
sshd: 192.168.154.0/255.255.255.0, 192.168.155.0/255.255.255.0,
127.0.0.1
```

DANGER Variable d'environnement PATH de root

Supposons que `root` ait mis le répertoire `/tmp` dans sa variable PATH. Un utilisateur mal intentionné peut créer une commande `/tmp/moer` qui supprime tous les fichiers du système (commande shell `rm –rf /*`). Sans privilège particulier, cette commande présente peu d'intérêt.

Si maintenant l'administrateur système travaillant sous le compte `root` fait une faute de frappe et appelle la commande `moer` au lieu de `more`, il exécutera la seule commande `moer` présente sur le système, c'est-à-dire celle que le pirate aura mise en place dans le répertoire `/tmp`. L'administrateur aura ainsi, bien malgré lui, effacé l'ensemble du système.

Voilà pourquoi la variable d'environnement PATH de `root` ne doit pointer que sur des répertoires où les utilisateurs n'ont pas accès en écriture. Il faut en particulier éviter les répertoires `/tmp` et `.` (point).

OUTILS TCP Wrapper

TCP Wrapper est un outil de filtrage applicatif développé dans les années 1990 par Wietse Zweitze Venema. Il est intégré à la majeure partie des distributions Linux. Sa simplicité et sa stabilité ont fait son succès. Il est notamment très connu pour permettre de filtrer les accès aux services réseau gérés par le serveur Internet Unix `inetd`.

- http://www.porcupine.org/wietse

Configuration des services système cron et syslog

cron

Le service `cron` est utilisé à Tamalo.com pour exécuter des tâches administratives régulières non interactives. Par défaut à l'installation, tous les utilisateurs définis sur le système peuvent l'utiliser. La procédure qui suit a pour but de n'autoriser que le compte administrateur `root` à utiliser ce service.

```
# umask 022
# echo root > /etc/cron.allow
# chown root:root /etc/cron.allow
# chmod 400 /etc/cron.allow
# rm -f /etc/cron.deny
```

Il est également prudent de vérifier les permissions des répertoires `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly`, `/etc/cron.monthly`, de leur contenu et du fichier `/etc/crontab`. Seul l'utilisateur `root` doit posséder les droits d'écriture, de lecture et d'exécution.

ATTENTION

Certains paquetages logiciels peuvent déployer une `crontab` pour les comptes utilisateur qu'ils déploient, par exemple afin de mener à intervalles réguliers des purges ou vérifications. On prendra donc soin de vérifier le contenu du fichier `/etc/crontab` et de ses dépendances (ce qui s'y trouve cité).

syslog

Le service `syslog` fait partie des rares services qui n'ont pas été désactivés lors de cette mise en configuration minimale. `syslog` gère le journal des systèmes Unix. Il offre la possibilité d'archiver localement sur chaque machine, ou de centraliser sur une machine unique, les messages qui lui sont envoyés par le noyau système ou les services actifs (`crond`, `sshd`…). Il permet d'effectuer un aiguillage en fonction de l'origine des messages et de leur caractère d'urgence. Ce service est indispensable à la surveillance et à l'administration du système.

Le fichier de configuration `/etc/syslog.conf` créé à l'installation est assez complet. Seule la dernière directive a été ajoutée pour que les messages soient également centralisés sur une machine du réseau Tamalo.com.

La machine d'adresse IP 192.168.154.250 joue le rôle de centraliseur de syslog. Son fichier de configuration `/etc/syslog.conf` ne doit évidemment pas contenir la directive `@192.168.154.250` pour éviter des bouclages sans fin. Le démon `syslogd` doit y être également lancé avec l'option `-r` (à modifier dans le fichier `/etc/sysconfig/syslog`), afin d'autoriser la réception des journaux en provenance d'autres machines du réseau.

Pour toutes les machines à l'exception du centraliseur de syslog, le fichier de configuration du démon syslogd est le suivant.

Fichier de configuration du démon syslogd

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                          /dev/console
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none  /var/log/messages
# The authpriv file has restricted access.
authpriv.*/var/log/secure
# Log all the mail messages in one place.
mail.*   /var/log/maillog
# Log cron stuff
cron.*   /var/log/cron
# Everybody gets emergency messages
*.emerg  *
# Save news errors of level crit and higher in a special file.
uucp,news.crit/var/log/spooler
# Save boot messages also to boot.log
local7.* /var/log/boot.log
# Log anything of level info or higher to our remote syslog
server.
*.info; @192.168.154.250
```

Configuration sécurisée de la pile TCP/IP

Certains paramètres de la configuration réseau IP du système doivent être modifiés de manière à en renforcer la robustesse vis-à-vis d'attaques potentielles. Comme c'est souvent le cas, les paramètres par défaut permettent de prendre nativement en charge beaucoup de fonctionnalités, mais ils sont hélas trop permissifs.

Sous Red Hat, la configuration de la couche réseau IP de Linux se fait en modifiant le contenu du fichier de configuration `/etc/sysctl.conf`.

Nous allons énumérer l'ensemble des paramètres nécessitant une modification pour pallier le laxisme de la configuration par défaut.

ALTERNATIVE Autres distributions : /proc

Sous d'autres distributions, on manipulera le contenu de /proc grâce à une commande telle que :

```
echo "1" > /proc/sys/net/ipv4/
conf/all/log_martians
```

Ignorer certains messages ICMP

ICMP Redirect

Le but des messages `ICMP Redirect` est d'indiquer à une machine ou à un routeur qu'il y a un chemin plus court pour joindre une destination donnée. Ils sont en général utilisés dans les réseaux d'interconnexion de routeurs et non au sein de réseaux locaux.

Comme on le voit sur la figure 5-5, la passerelle par défaut de la machine `client.tamalo.com` pointe sur `routeur A`, qui fait suivre tous les paquets à `routeur B` afin de sortir vers Internet.

B.A.-BA Internet Control Message Protocol

ICMP (Internet Control Message Protocol), est un protocole de contrôle de la couche réseau IP. Son but n'est pas le transport des données, mais plutôt la régulation du fonctionnement de la couche réseau par l'envoi de messages de contrôle.

Sur la totalité des systèmes de l'entreprise Tamalo.com, les paquets `ICMP Redirect` sont ignorés pour éviter les attaques de type MiM décrites en aparté. Pour cela, les lignes suivantes sont ajoutées au contenu du fichier `/etc/sysctl.conf` après l'installation du système.

```
# Ignore ICMP Redirect message
net.ipv4.conf.all.accept_redirects = 0
```

ICMP Echo request

Les paquets `ICMP Echo request` envoyés par la commande `ping` peuvent, dans certains cas, être utilisés pour scanner un réseau ou pour provoquer un déni de service. Il est possible de désactiver la prise en compte de ces requêtes. Comme pour le paramètre précédent, il s'agit de modifier le contenu du fichier de configuration `/etc/sysctl.conf`.

```
# Ignore ICMP Echo request message
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.icmp_echo_ignore_all = 1
```

Attention

Cette modification n'est effective qu'après le redémarrage du système !

Si l'on préfère ne pas recourir à cette solution pour le moins extrême, il est possible de spécifier grâce au `threshold` d'IPtables un seuil de quelques paquets par seconde (déterminé en fonction des capacités du réseau et de la machine).

```
iptables -A INPUT -p icmp --icmp-type echo-request -m limit
➥ --limit 8/s -j ACCEPT
```

Cette commande demande à IPtables de créer une règle pour n'accepter qu'un nombre limité de paquets ICMP de type `echo-request` par seconde. Cette règle doit bien entendu être accompagnée d'autres règles qui acceptent le trafic entrant toléré tandis que la toute dernière règle (ou « politique ») doit être d'ignorer (`drop`) le paquet IP.

Nous verrons comment configurer IPtables pour le filtrage de paquets au chapitre 8.

ICMP Ignore Bogus Response

```
# RFC's sending ICMP error replies to a broadcast frame is
forbidden,
# so drop response to them.
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

Interdiction du source routing

Le routage par la source est un mécanisme permettant à un système connecté au réseau d'envoyer des paquets à destination d'une machine en l'adressant à une autre machine, un routeur en général. Ce dernier, en ouvrant le paquet qui lui est explicitement envoyé, y trouve l'adresse de destination finale. Il modifie alors l'adresse IP de destination du paquet qui était

La sécurisation des services réseau est traitée au chapitre 6.

la sienne par cette nouvelle adresse, puis achemine le paquet réseau vers le destinataire. Dans le cas de la société Tamalo.com, le routage par la source est désactivé sur l'ensemble de machines et principalement sur les routeurs de réseaux.

```
# Disabling source routing
net.ipv4.conf.all.accept_source_route = 0
```

Surveillance des martiens !

Quand un paquet arrive sur une interface réseau d'une machine avec une adresse source falsifiée ou non routable, il est appelé martien. Sur le routeur d'entrée, il est utile de garder une trace de ces paquets, qui révèlent en général un problème de configuration réseau ou une tentative de piratage.

```
# Enabling loggig of martian for all network interfaces
net.ipv4.conf.all.log_martians = 1
```

Protection contre les attaques IP spoofing et SYN flooding

L'*IP spoofing* est une technique d'attaque réseau qui consiste à créer un paquet IP de toutes pièces (*forge*), avec une adresse source falsifiée afin de passer la barrière des filtres réseau. Lorsque le paquet vient de l'extérieur de votre réseau mais arbore une adresse source interne, situation bien évidemment anormale, il faut pouvoir le filtrer.

Le paramètre `rp_filter` permet d'éviter dans une certaine mesure les attaques par IP spoofing. L'activation de ce paramètre est obligatoire sur une machine Linux qui fait office de routeur. Par défaut, les administrateurs de Tamalo.com ont décidé de l'inclure sur l'ensemble des machines, même si pour la plupart cela n'a pas vraiment de conséquence.

```
# Enabling address spoofing detection
net.ipv4.conf.all.rp_filter = 1
```

La couche réseau du noyau Linux permet également de se défendre contre les attaques de type *SYN flood*. Ces attaques, dont le but est de provoquer un déni de service sur la machine cible, fonctionnent en envoyant une très grande quantité de requêtes d'ouverture de connexions TCP sans suite, dans le but de saturer la table des connexions de la couche réseau du système d'exploitation, paralysant ainsi l'activité de ce dernier. Le paramètre `tcp_syncookies` agit sur la protection du système. Toutes les machines de la société sont configurées pour activer cette protection.

```
# Enabling tcp_syncookies
net.ipv4.tcp_syncookies = 1
```

Filtrage sans état

En analysant le contenu du paquet, il s'agit de décider de l'accepter ou de le refuser. Les informations exploitables pour la décision sont détaillées dans les paragraphes suivants.

Adresses IP source et destination

Seules certaines machines du réseau sont des serveurs et ont, par conséquent, vocation à offrir un service visible de l'extérieur. Il faudra donc autoriser les tentatives de connexions à destination de ces serveurs et filtrer les autres.

Protocole, ports source et destination

Un service utilise son protocole propre grâce à des paquets réseau véhiculés par TCP/IP, en mode TCP (sessions) ou UDP (non connecté) vers un port donné. L'exemple le plus classique est celui du Web, qui utilise le protocole TCP et dont le service écoute sur le port 80. La plupart des services réseau utilisent un port connu, ce qui permet à une machine externe de les contacter sans autre information que le nom du serveur. Cette possibilité est également utilisée par les pirates qui établissent ainsi facilement la liste des serveurs Web d'un réseau donné par un simple scan du port 80 !

Le pare-feu va s'appuyer sur les informations lues dans le paquet – à savoir le protocole, le port source et le port destination – afin de garantir que, pour une machine donnée, seuls les services voulus seront accessibles de l'extérieur. Par exemple, pour une machine assurant un service Web, seul le port 80 TCP sera rendu accessible de l'extérieur par le pare-feu.

Parmi les 65 535 ports possibles, les ports 1 à 1023 ont une caractéristique particulière dans un environnement Unix, donc Linux. En effet, avec ces systèmes, pour ouvrir un port inférieur à 1024, le service doit tourner en mode privilégié, c'est-à-dire sous le compte administrateur `root`.

B.A.-BA Connexions réseau

Une connexion réseau IP est définie par une adresse source, un port source, une adresse de destination et un port de destination, que le protocole soit UDP ou TCP.

NORMALISATION Well known ports

Les *well known ports* sont les ports applicatifs dont les numéros ont été fixés par l'IANA (Internet Assigned Numbers Authority) pour un service donné. Sous Linux, la correspondance entre le numéro de port et le nom du service est donnée dans le fichier `/etc/services`.

À l'intérieur d'un paquet, le numéro de port est stocké dans un entier de 2 octets. La valeur d'un port est donc comprise entre 0 et 65 535. L'IANA a prévu différentes significations d'un port applicatif en fonction de sa valeur.

- 1 à 1 023 : well known ports. Ces « ports bien connus » sont associés à un service de base, par exemple, 21 : FTP, 23 : TELNET, 25 : SMTP (Simple Mail Transfert Protocole), 80 : HTTP…
- 1 024 à 49 151 : registered ports. Ces ports ont été réservés pour un service donné, ce service n'étant pas forcément exécuté dans un environnement privilégié. Citons par exemple 3 306 : MySQL ou 2 401 : CVS.
- 49 152 à 65 535 : dynamic ports utilisés par les sessions.

➤ http://www.iana.org/assignments/port-numbers

INCONTOURNABLE Règles de base du filtrage

La rigueur de la politique de filtrage mise en place dépend bien sûr du risque informatique que l'on est en mesure d'assumer, comme cela a été décrit au chapitre 1. Dans ce qui suit, nous décrivons cependant un certain nombre de règles auxquelles il serait quelque peu téméraire de déroger.

Filtrer les adresses internes

Une adresse IP falsifiée (IP spoofing) est une adresse fournie dans un paquet qui a été modifiée par un pirate. Il va de soi que l'on aimerait bien refuser tout paquet contenant une telle adresse, mais malheureusement il n'y a pas de moyen de distinguer un paquet falsifié d'un vrai paquet, sauf dans de rares cas comme ceux décrits ci-dessous.
Imaginez que sur l'interface externe de votre routeur d'entrée vous voyez arriver un paquet présentant une adresse IP source appartenant à votre propre réseau ! Il ne peut s'agir que de l'œuvre d'un pirate et il faut clairement refuser de tels paquets. En maquillant ainsi les adresses sources de ses paquets, le pirate cherche tout simplement à contourner votre politique de sécurité. En effet, en prétendant avoir une adresse IP interne, le pirate obtient des droits forcément plus importants que ceux associés à une adresse IP externe !
La première règle de filtrage va donc consister à refuser tous les paquets qui se présentent sur l'interface externe de notre routeur, dans le cas où l'adresse source prétendrait être une adresse interne à notre réseau. En particulier, il faudra refuser les paquets contenant `localhost 127.0.0.1` comme adresse source !

Refuser le routage par la source

Le routage par la source, en anglais source routing, est une option TCP qui permet à l'émetteur d'indiquer à l'intérieur du paquet le chemin qu'il doit prendre pour rejoindre sa destination. Une telle option permet de maîtriser le cheminement du paquet. Elle peut malheureusement être utilisée par les pirates pour contourner votre politique de filtrage. Par exemple, pour protéger un serveur qui n'a pas besoin d'être visible de l'extérieur, vous lui affectez une adresse non routable `10.10.0.1`. Pourtant, grâce au source routing, un pirate pourra envoyer à votre routeur d'entrée un paquet dont la destination finale est le serveur sensible `10.1.1.1`. Si le source routing est autorisé, le routeur fera aimablement suivre le paquet vers le serveur comme indiqué sur la figure 7-1.

Interdire l'ICMP Redirect

Ce point particulier est traité dans le chapitre 5.

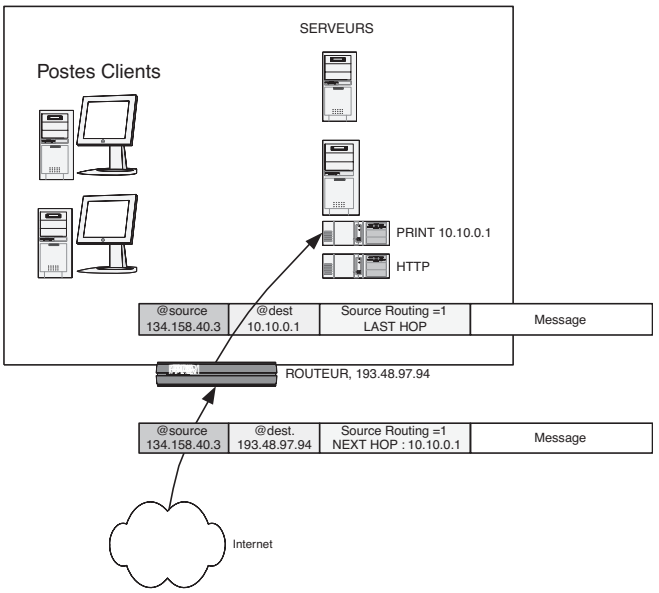


Figure 7–1 Les dangers du source routing

Ceci aggrave les conséquences de la présence de vulnérabilités du service (voir chapitre 3.). En effet, si le pirate parvient à en prendre le contrôle pour lui faire exécuter un code arbitraire, ce code s'exécutera sous le compte `root`, ouvrant ainsi un accès privilégié sur la machine. Il existe des mécanismes permettant de contourner cette lacune et d'exécuter des services sur des ports privilégiés sans être au moment précis de la connexion sous l'identité du compte `root`. Ces possibilités seront abordées au chapitre 6.

Drapeaux TCP et filtrage en entrée

L'objectif poursuivi par la mise en place d'un pare-feu est le plus souvent de permettre à un utilisateur interne de sortir librement, tandis qu'on veut

interdire à un utilisateur externe d'accéder aux ressources internes. Cette politique de filtrage est qualifiée de filtrage en entrée.

Le filtrage en entrée ne doit donc pas être confondu avec le filtrage de tous les paquets entrants. Il s'agit de refuser un paquet entrant s'il ouvre une connexion vers une machine située à l'intérieur du site. En revanche, le pare-feu devra accepter les paquets qui entrent, s'ils sont des réponses à une requête initialisée depuis l'intérieur.

Dans le cas d'une connexion TCP, un certain nombre de drapeaux, *flags* en anglais, contenus dans l'en–tête TCP du paquet permettent de connaître l'état de la connexion. Le routeur filtrant va donc pouvoir s'appuyer sur ces drapeaux pour savoir comment traiter le paquet.

Le tableau ci-dessous donne les différents drapeaux contenus dans l'en-tête TCP et l'état des bits à l'ouverture d'une connexion.

```
Flags: 0x0002 (SYN)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...0 .... = Acknowledgment: Not set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..1. = Syn: Set
.... ...0 = Fin: Not set
```

À l'aide de Ethereal, une analyse des paquets échangés au moment de l'ouverture d'une connexion TCP permet d'identifier les configurations de bits qui la caractérisent (figure 7-2).

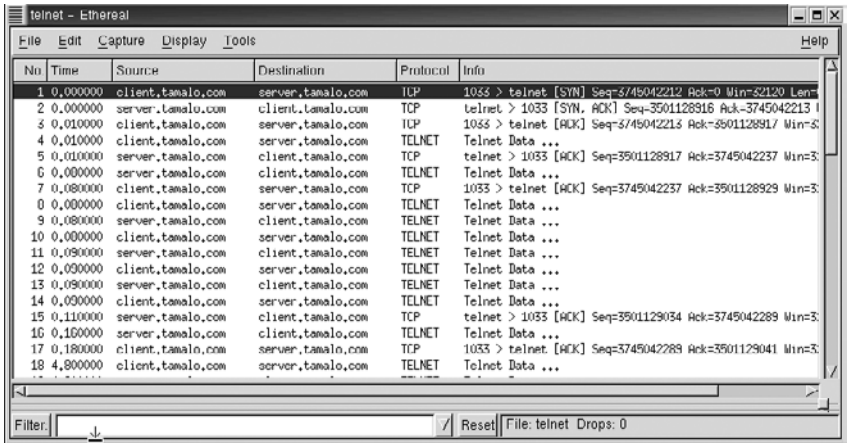


Figure 7–2
Analyse d'une connexion TCP avec Ethereal

Comme cela est schématisé sur la figure 7-3, à l'ouverture de la connexion, le premier paquet qui part du client vers le serveur possède la configuration de drapeaux suivante : `SYN=1`, `ACK=0`. Le paquet renvoyé en retour a `SYN=1`,

ACK=1. Les paquets qui suivent ont tous les bits SYN=0 et ACK=1. Il est donc possible de distinguer l'ouverture de connexion par la configuration SYN=1 ACK=0.

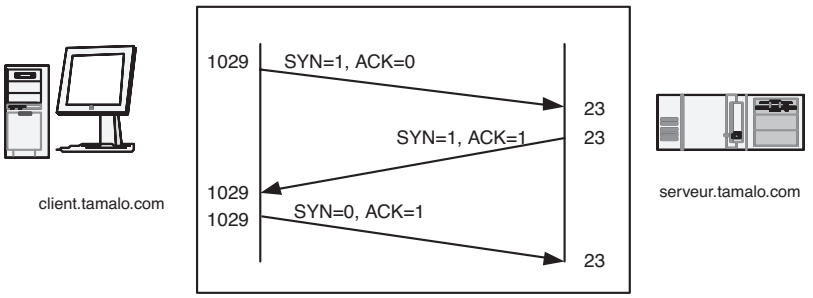


Figure 7–3
Ouverture d'une session TCP, exemple de TELNET

Les limites du filtrage sans état

Le filtrage sans état, *stateless* en anglais, prend la décision d'accepter ou de refuser un paquet en se basant uniquement sur les informations que ce dernier contient. L'avantage d'une telle méthode est qu'elle est très peu gourmande en ressources CPU et mémoire.

En effet pour chaque paquet, il suffit de regarder :

- les adresses IP source et destination ;
- les ports source et destination ;
- pour une connexion TCP, l'état des bits, en particulier SYN et ACK.

À partir de ces informations, la passerelle d'entrée peut prendre la décision d'accepter ou de refuser le paquet.

DANGER Les règles de filtrage sans état peuvent être contournées avec un paquet falsifié

Le routeur d'entrée d'un site est configuré pour effectuer un filtrage de paquet sans état. Le filtrage en entrée est effectué en laissant passer les paquets qui présentent ACK=1 ou RST=1. (*)

- ACK=1 permet de n'accepter que les retours de connexion, ce qui implique que la connexion a été initialisée depuis l'intérieur du site.
- RST=1 permet de laisser passer les demandes de fermeture (RESET) de connexion, ce qui permet sur le serveur une terminaison propre du processus correspondant.

Un paquet présentant la configuration falsifiée suivante : SYN=1, ACK=0, RST=1, va donc franchir le routeur ainsi configuré. Cette configuration est incohérente puisque qu'elle demande à la fois une ouverture de connexion SYN=1 et sa fermeture RST=1. Elle est quand même acceptée comme une ouverture de connexion par certaines piles TCP/IP, dont celle des noyaux Linux 2.4.

En voici un exemple pratique. Un serveur Web basé sur un noyau Linux 2.4 est destiné à fournir un service intranet. Celui-ci est protégé contre les ouvertures de connexion par un filtrage sans état autorisant les connexions ESTABLISHED comme décrit ci-dessus. Un pirate pourra malgré tout se connecter sur ce serveur depuis l'extérieur ! Il lui suffira pour cela de recompiler sa pile TCP/IP pour positionner à 1 le bit RST à chaque ouverture de connexion !

(*) Cette configuration est celle qui correspond au mode ESTABLISHED avec un routeur CISCO. Sur de tels routeurs, pour parer à cette défaillance, il est recommandé d'utiliser le mode ACK en lieu et place du mode ESTABLISHED. En complément, il est possible de paramétrer explicitement le pare-feu Iptables pour refuser de tels paquets.

L'inconvénient du filtrage sans état est qu'il fait confiance au contenu du paquet sans recouper les informations de ce dernier avec une autre source. Or le paquet est émis par l'expéditeur qui, s'il est mal intentionné, a pu modifier à volonté le contenu de celui-ci. Un tel paquet modifié par un pirate est qualifié de paquet falsifié.

ATTENTION Mode avec états

La mise en place d'un filtrage avec états (module *connection tracking* avec IPtables) nécessite des ressources CPU et mémoire augmentant avec l'activité réseau, en particulier avec le nombre de sessions TCP simultanément actives. Si le pare-feu est sous-dimensionné, ce mode peut entraîner une détérioration des temps de réponse du réseau.

Filtrage avec états

Un pare-feu avec états, *stateful* en anglais, garde en mémoire les sessions logiques (échanges de paquets) en cours. Sur cette base, il est capable de modifier dynamiquement ses règles de filtrage. Ainsi, il lui sera possible d'autoriser l'entrée de paquets qui sont des réponses à des connections initialisées depuis l'intérieur du réseau comme cela est décrit sur la figure 7-4.

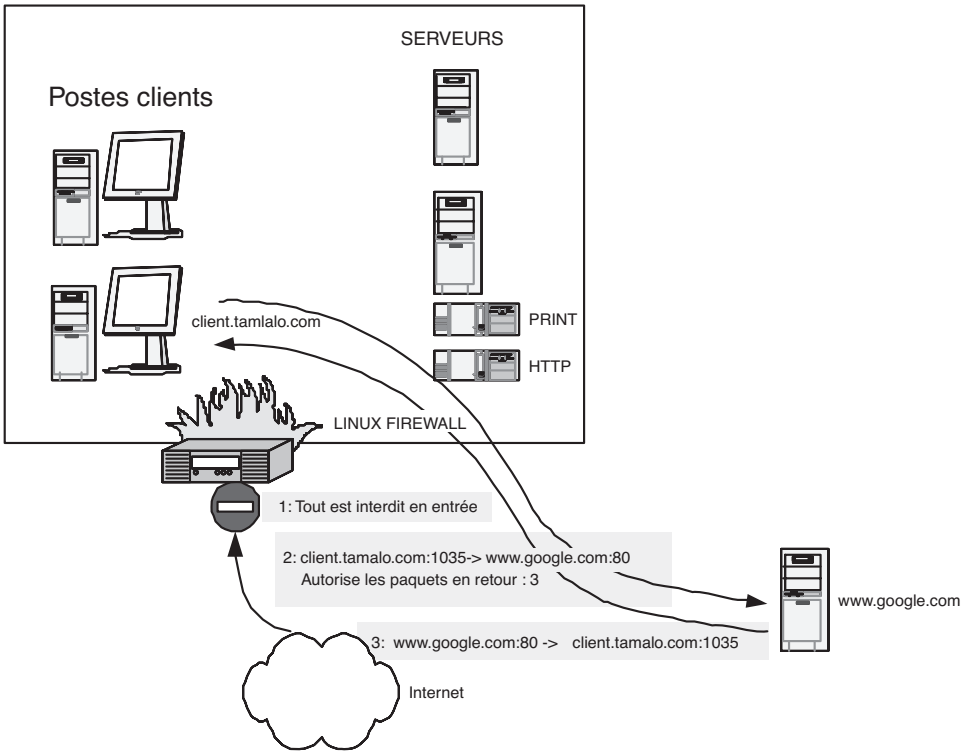


Figure 7–4 Pare-feu avec états, règles de filtrage dynamique

Les avantages du filtrage avec états sont de plusieurs ordres. Comme le pare-feu ne se fie pas seulement au contenu du paquet, il est plus difficile de le tromper avec des paquets falsifiés.

En outre, il rend possible l'autorisation des retours de connexions (connexions dites *established)* en mode TCP, mais aussi en mode UDP, ce qui n'est pas le cas des pare-feu sans états.

Enfin, il sera également possible de faire du suivi de connexion pour prendre en charge des protocoles complexes où les retours de connexions n'utilisent pas le même couple de ports que la connexion aller. C'est le cas par exemple du protocole FTP décrit ci-après, mais aussi de la plupart des protocoles UDP utilisés pour transférer des flux vidéo ou audio (comme le protocole H323 utilisé en visioconférence).

Politique de filtrage : avant la compromission, « tout ouvert sauf »

Les informaticiens de Tamalo.com étant très attachés à l'ouverture sur Internet, la seule politique envisagée dans un premier temps avait été basée sur une ouverture par défaut.

Le principe en est simple : seuls les ports pour lesquels une vulnérabilité est connue sont fermés par défaut. Ces ports sont ouverts seulement vers les machines qui offrent volontairement le service correspondant ; ils sont fermés vers toutes les autres machines.

Tableau 7–1 Politique de filtrage « tout ouvert sauf » : liste des services fermés

Protocole	Port	Service	Interdit vers
TCP	21	FTP	Interdit vers tous sauf ftp.tamalo.com
TCP	23	TELNET	Interdit pour toutes les machines, le service est inactivé par défaut
TCP	25	SMTP	Interdit vers tous sauf smtp1.tamalo.com
UDP	53	domain	Interdit vers tous sauf ns1.tamalo.com et ns2.tamalo.com
TCP	69	TFTP	Interdit pour toutes les machines
	79	Finger	Interdit pour toutes les machines
TCP	80	Web	Interdit vers tous sauf www.tamalo.com
TCP	109	Pop2	Interdit pour toutes les machines
TCP	110	Pop3	Interdit pour toutes les machines
	111	Sunrpc	Interdit pour toutes les machines
	119	NNTP	Interdit pour toutes les machines
TCP	137, 138, 139	SMB-CIFS (Samba)	Interdit pour toutes les machines
TCP	143	IMAP	Interdit vers tous sauf imap.tamalo.com
TCP	512	Exec	Interdit pour toutes les machines
UDP	512	Biff	Interdit pour toutes les machines

Tableau 7–1 Politique de filtrage « tout ouvert sauf » : liste des services fermés (suite)

Protocole	Port	Service	Interdit vers
TCP	513	Login	Interdit pour toutes les machines
UDP	513	Who	Interdit pour toutes les machines
TCP	514	cmd	Interdit pour toutes les machines
UDP	514	syslog	Interdit pour toutes les machines
	2049	Nfsd	Interdit pour toutes les machines

À la date du piratage, la politique était donc celle de l'ouverture par défaut. En particulier, comme on le voit dans le tableau 7-1, aucun filtre ne s'appliquait sur le port TCP 515. Malheureusement, une vulnérabilité de type *buffer overflow* sur le service d'impression `LPRng` en écoute sur ce port a été découverte entre-temps. Comme cela est décrit au chapitre 3, cette vulnérabilité a pu être exploitée sur nos machines avant la mise en place d'une règle de filtrage adaptée.

Les déboires de l'équipe de Tamalo.com montrent bien les limites d'une politique de filtrage ouverte par défaut. Elle exige des administrateurs une vigilance accrue ajoutée à un circuit d'information très rapide pour mettre à jour les filtres aussitôt qu'une vulnérabilité est annoncée. De plus, ces mêmes administrateurs devront avoir une connaissance parfaite de l'ensemble des services ouverts sur les machines de leur réseau pour les tenir à jour. Comment gérer alors le développeur qui lancera un serveur Tomcat en écoute sur le port 8180 ou mySQL en écoute sur le port 3306 ? Il n'y a que deux solutions : « marquer à la culotte » les développeurs pour s'assurer qu'ils ne lancent aucun service vulnérable, ou bien protéger les utilisateurs malgré eux en rendant par défaut leur réseau inaccessible depuis l'extérieur.

Politique de filtrage : du « tout ouvert sauf » au « tout fermé sauf »

Le premier niveau de protection déployé chez Tamalo.com a donc été la mise en place d'une politique de filtrage ou tout est fermé par défaut.

Le déploiement d'une telle politique a des conséquences très positives. Elle permet une meilleure connaissance du réseau en forçant à établir, et à maintenir à jour, la liste des services utiles depuis l'extérieur. Il s'agit d'un aspect très important, un réseau non documenté n'étant pas gérable du point de vue de la sécurité. La documentation est également le premier élément qui sera étudié pour effectuer un audit de sécurité.

Elle force également la mise en place d'un protocole pour l'ouverture d'un nouveau service à l'extérieur. En effet, cette ouverture demande l'interven-

tion du responsable du réseau qui, avant d'ouvrir le port considéré, effectue un certain nombre de vérifications : existence de vulnérabilité connue sur ce service, mise à jour éventuelle de la version du service correspondant.

Le tableau 7-2 donne la liste des couples machines/service qui ont été maintenus ouverts avec la mise en place de cette nouvelle politique.

Tableau 7–2 Politique de filtrage « tout fermé sauf » : liste des services ouverts

Protocole	PORT	Service	Autorisé vers
TCP	21	FTP	ftp.tamalo.com
TCP	25	SMTP	smtp1.tamalo.com
UDP	53	domain	ns1.tamalo.com ns2.tamalo.com
TCP	80	HTTP	www.tamalo.com
TCP	443	HTTPS	www.tamalo.com
TCP	993	IMAPS	imap.tamalo.com
TCP	8180	tomcat	www.tamalo.com

Déploiement de service FTP avec (et malgré) les filtres

Le protocole de transfert de fichier FTP (File Transfer Protocol) est tellement diffusé qu'il est presque inavouable d'avoir envisagé un instant de s'en passer. Et pourtant, il aurait été plus facile et plus sûr d'en priver nos utilisateurs ! Comme cela est décrit dans ce qui suit, le déploiement de FTP et son utilisation dans ses différents modes a été une source de difficultés quant à la mise en place des filtres, et sera très souvent à l'origine de comportements dont l'interprétation est malaisée.

Dans le cadre de l'activité de Tamalo.com, FTP est utilisé dans les deux sens. D'une part, l'entreprise offre un service FTP pour le téléchargement des programmes diffusés. Ce service doit évidemment être accessible depuis l'extérieur. D'autre part, la plupart des utilisateurs du site utilisent un client FTP pour rapatrier des fichiers depuis Internet. La mise en place de filtres ne doit pas contraindre cette possibilité.

FTP fonctionne exclusivement en mode TCP. Il est différent des autres services car il a deux modes de fonctionnement, le mode actif et le mode passif. De plus, FTP met en œuvre deux ports : le port 21, utilisé pour les commandes ou contrôles ; le port 20 – sauf en mode passif ! – utilisé pour le transfert des données.

LE PROTOCLE FTP FTP actif

Le fonctionnement de FTP en mode actif est décrit sur la figure 7-5.

1. Le serveur FTP écoute sur le port 21.
2. Le *client amorce une connexion* depuis un port N (1 029 sur le schéma) non privilégié vers le port 21 du serveur.
3. Le client écoute sur le port N+1 (port de rendez-vous : 1 030 sur le schéma). Il l'indique au serveur par la commande PORT N+1.
4. Le *serveur initie une connexion* depuis le port 20 vers le port N+1 du client.

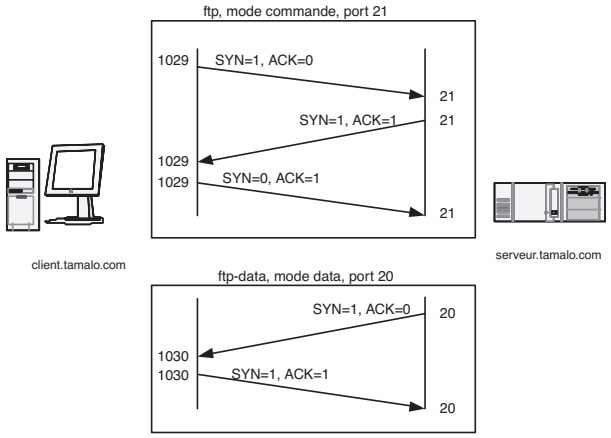


Figure 7–5 FTP actif : connexion du client vers le serveur, puis du serveur vers le client

Le mode actif met en œuvre une double connexion : d'abord pour les commandes du client vers le serveur, puis pour le transfert des données du serveur vers le client.

Analyse du protocole

Comme toute application client-serveur, FTP s'appuie sur un protocole bien défini. Ce dernier utilise un certain nombre de mots-clés (USER, PASS, PORT, LIST, QUIT), qui sont échangés entre le client et le serveur pour effectuer les actions correspondantes.

Ethereal permet d'analyser finement le fonctionnement de FTP en mode actif. La copie d'écran de la figure 7-6 représente les diffé-

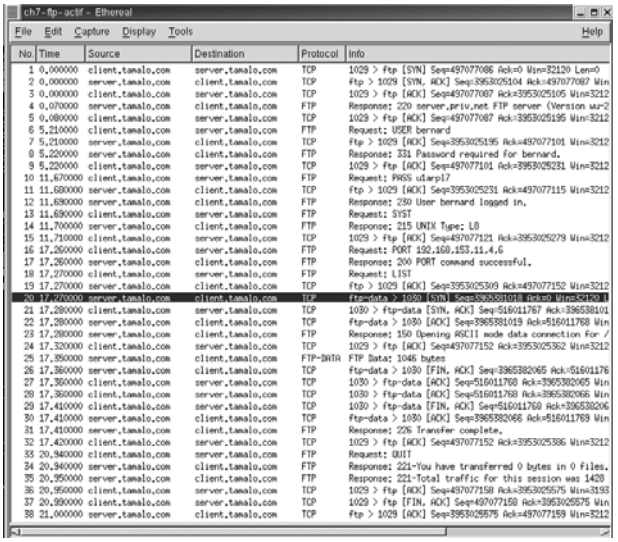


Figure 7–6 Analyse d'une session FTP en mode actif avec Ethereal

rentes étapes de la connexion FTP. Paquet n°1 : `client.tamalo.com` ouvre une connexion vers `server.tamalo.com`. Il est possible de vérifier au passage l'état des drapeaux (*flags* en anglais) : SYN=1, ACK=0. L'authentification est effectuée dans les paquets qui suivent ; notez, au paquet 10, le mot de passe qui transite en clair ! Paquet n°16 : le client demande au serveur, par la commande `PORT 192,168,153,11,4,6`, de le re-contacter sur le port `4*256+6=1030`, à l'adresse `192.168.153.11` (celle de `client.tamalo.com`). Cette demande est acquittée, puis le client demande un transfert par la commande `LIST`. Paquet n°20 : le serveur ouvre alors une connexion sur le port 1030 du client. Cette connexion est normale en regard du protocole, mais elle a pour effet de transformer en serveur une simple machine de bureau, ce qui ne va pas sans poser quelques problèmes de filtrage.

Filtrage d'un client FTP actif

Comme cela est décrit en aparté, en mode actif le client initie une connexion TCP vers le port 21 du serveur, puis quand un transfert est demandé, c'est le serveur qui se connecte à son tour depuis son port 20 vers un port du client.

Considérons le réseau de Tamalo.com. Nous voulons permettre aux employés d'utiliser – en tant que clients – le mode FTP actif. Il faudrait donc rendre possible une ouverture de connexion depuis le port 20 d'une machine externe quelconque vers tous les ports supérieurs à 1 024 des postes de travail du réseau !

Émulation du protocole FTP actif avec une connexion TELNET

Pour les applications qui fonctionnent en mode TCP, il est possible d'émuler le protocole à partir d'une connexion TELNET sur le port du service considéré.
Ainsi, pour émuler une connexion FTP mode actif entre `client.tamalo.com 192.168.153.10` et `ftp.tamalo.com 192.168.153.11`, on exécutera les commandes du tableau ci-dessous (voir aussi figure 7-7).
Pour être capable de recevoir la connexion en retour du serveur FTP vers la machine `client.tamalo.com`, il faut ouvrir sur celle-ci un service en écoute sur le port de rendez-vous, à savoir `192.168.153.11 :44 847`
Cela peut être fait à l'aide de la commande `nc` (utilitaire `netcat`), déjà utilisée dans un tout autre contexte au chapitre 3. Sur le client `client.tamalo.com`, on lancera donc la commande suivante pour écouter sur le port 44 847 (figure 7-7).
`nc -l 44847`

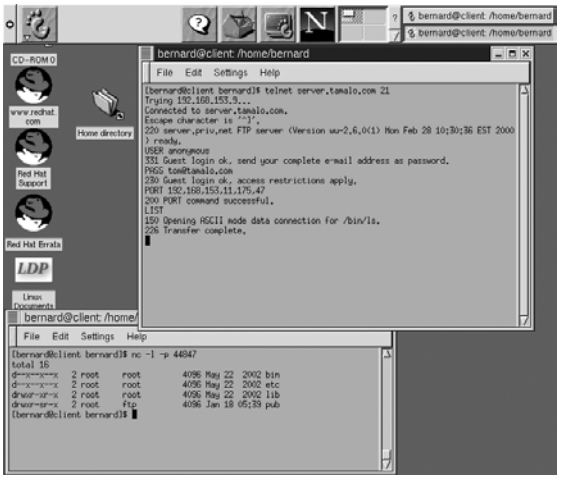


Figure 7–7 Émulation du protocole FTP actif

`telnet ftp.tamalo.com 21`	Connexion sur le port 21 FTP du serveur ftp.tamalo.com
`USER anonymous`	Transmet le nom de l'utilisateur.
`PASS tom@tamalo.com`	Transmet le mot de passe.
`PORT IP1,IP2,IP3,IP4,P1,P2`	IP1,IP2,IP3,IP4 est l'adresse IP de votre machine, P1*256+P2 est le numéro du port de rendez-vous. Ainsi, PORT 192,168,153,11,175,47 indique au serveur de se connecter sur la machine 192.168.153.11 sur le port 44 847=175*256+47
`LIST`	Demande la liste des fichiers. Cette liste sera transmise en mode DATA. Elle sera envoyée depuis le port 20 du serveur sur le port 44 847 du client.
`QUIT`	Ferme la connexion.

Ce faisant, nous ouvririons une brèche considérable dans notre réseau. En effet, un pirate pourrait se connecter sur n'importe quel service interne correspondant à un port supérieur à 1024, en forçant simplement le port source à 20.

Deux types de parade seront décrits dans ce qui suit, afin de ne pas ouvrir une telle brèche tout en conservant la possibilité d'utiliser le mode FTP actif.

- L'une consiste à utiliser un filtrage avec états, doté de la possibilité d'effectuer du suivi de connexion, ou *connection tracking* en anglais.
- L'autre consiste à mettre en œuvre un mandataire (*proxy*) applicatif qui relaiera le protocole FTP. Cette solution sera détaillée au chapitre 8.

PROTOCOLE FTP passif

Le fonctionnement de FTP en mode passif est décrit à la figure 7-8.

5. Le serveur écoute comme en mode actif sur le port 21.
6. Le *client initie une connexion* depuis un port non privilégié vers le port 21 du serveur.
7. Le client demande au serveur de passer en mode passif avec la commande PASV.
8. Le serveur se met en écoute sur le port M et transmet ce numéro de port au client.
9. Le *client initie une connexion* depuis un port non privilégié vers le port M du serveur.

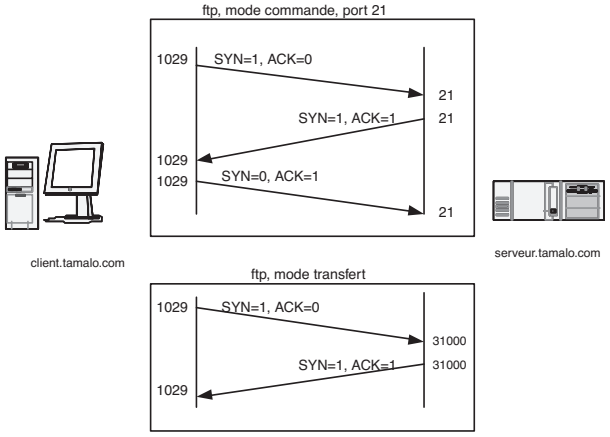


Figure 7–8 FTP mode passif : ouvertures de connexions toujours du client vers le serveur

Analyse du protocole

De la même façon que précédemment, il est possible de se connecter en utilisant la commande `telnet` sur le serveur et d'émuler le protocole FTP en mode passif (voir l'écran de la figure 7-9).

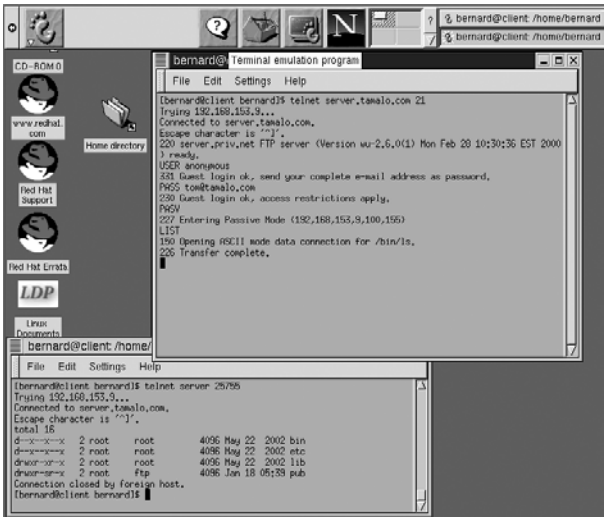


Figure 7–9 Émulation du protocole FTP passif

`telnet ftp.tamalo.com 21`	◄	Connexion sur le port 21 ftp du serveur ftp.tamalo.com
`USER anonymous`	◄	Transmet le nom de l'utilisateur.
`PASS tom@tamalo.com`	◄	Transmet le mot de passe.
`PASV`	◄	Demande au serveur de passer en mode passif. Le serveur retourne le numéro de port sur lequel il s'est mis en écoute : 100*256+155=25 755
`LIST`	◄	Demande la liste des fichiers. Cette liste sera transmise en mode DATA. Elle sera transmise dans une nouvelle connexion effectuée dans une deuxième fenêtre (voir figure 7–9) depuis le client sur le port 25 755 du serveur.
`QUIT`	◄	Ferme la connexion.

Filtrage d'un serveur FTP destiné à fonctionner en mode actif

Le filtrage du serveur ne pose pas de problème particulier. En effet, pour le canal de commandes, le pare-feu devra autoriser les ouvertures de connexions uniquement vers le port 21 du serveur FTP. Pour le canal des données, c'est le serveur qui initie la connexion ; il n'y a donc pas de règle particulière à ajouter.

Filtrage d'un client FTP passif

Le mode FTP passif est bien mieux adapté à la problématique du filtrage. En effet, en mode passif, c'est toujours le client qui effectue les ouvertures de connexion vers l'extérieur. Une règle de filtrage du type `ESTABLISHED`, qui autorise les retours de connexion si elles ont été initialisées depuis l'intérieur, permet d'utiliser librement des clients FTP passifs situés à l'intérieur du site. Il est fortement recommandé de n'utiliser FTP qu'en mode passif et de ne pas déroger à la règle de filtrage total en entrée des adresses de machines internes du réseau.

Néanmoins, cette règle pourra dans certains cas impliquer des dysfonctionnements et il faudra l'adapter aux besoins :

- Les vieux clients FTP ne permettent pas toujours le mode passif.
- Certains clients FTP, ce n'est pas le cas sur Linux, sont souvent configurés par défaut pour travailler en mode actif. Si un utilisateur non averti tente une connexion en mode actif il va donc constater un fonctionnement anormal : FTP fonctionne, les commandes sont acceptées, tout se passe bien... sauf le transfert ! Plus ennuyeux encore : les navigateurs prennent nativement en charge le protocole FTP, ce qui permet de rapatrier des fichiers depuis une URL du type : ftp:// server.tamalo.com/. Malheureusement, sur un navigateur Web, il n'est pas possible de choisir le mode passif plutôt que l'actif. Si le navigateur tente la connexion en mode actif tout semble marcher mais aucun octet ne sera transféré !

Filtrage du serveur FTP passif, limitation du serveur à une plage de ports

Le serveur FTP doit prendre en charge les modes actif et passif. Cela lui permettra de s'adapter à tous les types de clients, quel que soit le mode de fonctionnement qu'ils implémentent. Comme cela a été décrit ci-dessus, le filtrage - d'un serveur - en mode actif ne pose pas de problème.

En mode passif, le client ouvre deux connexions vers le serveur, l'une sur le port 21, l'autre sur un port non privilégié qui est choisi arbitrairement par le démon `ftpd`.

Il faudra donc ouvrir les ports correspondants pour les ouvertures de connexion à destination du serveur FTP :

- le port 21 pour la connexion commande ;
- tous les ports situés au delà de 1 024, pour la connexion *data* (données).

Sur la plupart des serveurs FTP, il est possible de limiter la plage de ports de la connexion *data*.

Il est intéressant de limiter cette plage et d'autoriser les connexions *data* sur des ports qui ne sont réservés par aucun service. Ainsi, on évitera les « well known ports » et les « registered ports » pour se placer entre 49 152 et 65 535.

Avec le service `WU-FTPD`, très répandu sous Linux, il faut modifier le fichier `/etc/ftpaccess`. La plage de ports est limitée par la ligne suivante :

```
passive port 0.0.0.0/0 61000 61100
```

Cette commande indique au service `ftpd` d'ouvrir les ports `ftp-data` dans la plage `61 000 – 61 100`.

Attention, si votre serveur FTP est très sollicité, prévoyez une plage de ports assez importante. En effet, chaque transfert d'une même connexion va ouvrir un nouveau port qui mettra un certain temps à être libéré. Pour accueillir par exemple une dizaine de connexions simultanées, il vaut mieux prévoir une plage de quelques centaines de ports.

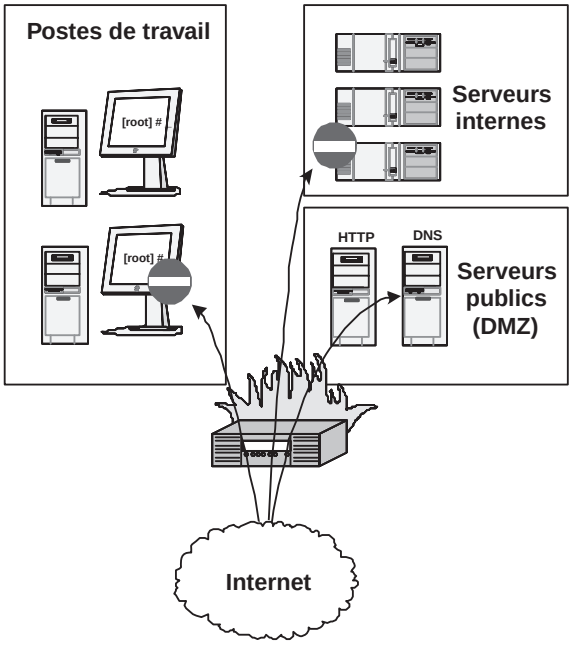
En résumé…

En introduisant un pare-feu entre le réseau local de l'entreprise et l'Internet, il est possible de contrôler le trafic entre ces deux mondes.

Le choix d'une politique de sécurité est indissociable des enjeux pour l'entreprise. Pourtant, même dans des contextes peu sensibles, il est indispensable aujourd'hui d'effectuer un minimum de filtrage en entrée de site. Dans les cas où cela est possible, une politique où tout est fermé par défaut est même vivement recommandée. Malgré tout, la mise en place des filtres doit être faite avec prudence car elle peut engendrer des dysfonctionnements, comme cela a été décrit dans le cas du protocole FTP. En cas de problème, l'analyse des protocoles qui sont mis en œuvre dans le réseau, à l'aide d'outils comme Ethereal, permettra dans la plupart des cas de contourner les difficultés.

Au moment de choisir un matériel, les possibilités de celui-ci en matière de filtrage avec états et de suivi de connexion devront être étudiées. Le couple Linux/IPtables bien configuré présente toutes les caractéristiques d'un excellent pare-feu.

chapitre 8


Postes de travail
[root] #
[root] #
Serveurs internes
HTTP
DNS
Serveurs publics (DMZ)
Internet

Topologie, segmentation et DMZ

La segmentation du réseau en de multiples sous-réseaux est un moyen de lutte efficace contre la propagation d'une compromission. Ce chapitre décrit et illustre les techniques nécessaires à cette compartimentation des réseaux.

SOMMAIRE

- Définition des zones et flux à l'intérieur du réseau de Tamalo.com
- Établissement de la topologie du réseau
- Segmentation du réseau grâce aux VLAN
- Protection des postes de travail : *proxy* et traduction d'adresse IP.
- Configuration des pare-feu avec IPtables
- Protection du réseau sans fil

MOTS-CLÉS

- DMZ, zones, flux et pare-feu
- Subnet, 802.1Q, 802.1X
- ARP cache poisoning
- arpwatch, proxy, SOCKS
- journalisation, IPtables
- source NAT, IP Masquerading
- PAT (Part Address Translation), destination NAT

Nous allons maintenant cloisonner le réseau d'entreprise. Cela évitera qu'un incident bénin, comme la compromission d'un serveur Web, puisse avoir pour conséquence la compromission complète du réseau. Avec ce cloisonnement, des filtres seront mis en place pour limiter l'accès non seulement depuis l'extérieur, mais aussi entre les différentes zones à l'intérieur même du réseau local de l'entreprise.

L'analyse des flux permet de définir la topologie réseau offrant la protection la plus efficace. À partir d'un exemple concret, nous décrirons comment segmenter le réseau et quels sont, parmi les services du réseau, ceux qui doivent prendre place dans la DMZ, de l'anglais *DeMilitarized Zone*.

Pour la topologie adoptée par Tamalo.com, nous étudierons quels bénéfices, pour protéger les postes de travail, il a été possible de tirer des mécanismes de traduction d'adresses (NAT : Network Address Translation) et comment cette technique est complémentaire de celle plus ancienne des *proxys*.

Enfin, nous verrons comment le couple Linux/IPtables a été utilisé pour construire les deux pare-feu utilisés chez Tamalo.com. Nous décrirons en détail la configuration IPtables de ces pare-feu positionnés en entrée de site, ainsi que celle utilisée pour protéger des serveurs spécifiques.

Pourquoi cloisonner ?

Après avoir vécu la prise de contrôle de plusieurs machines de Tamalo.com par un pirate situé à l'extérieur, l'intérêt de protéger le système informatique du monde Internet devient particulièrement évident. Le processus de sécurisation doit être complété par un cloisonnement interne pour répondre parfaitement aux nouvelles exigences de sécurité de l'entreprise.

La compromission décrite au chapitre 3 a montré une défaillance fondamentale dans les mécanismes de protection : après avoir pris le contrôle d'une machine qui offrait un service d'impression ouvert à l'extérieur de notre site, le pirate a eu la possibilité d'écouter l'ensemble des communications du réseau interne. Ce cas de figure pourrait se reproduire malgré la présence d'un pare-feu, si le réseau n'est pas cloisonné. Il faut donc appliquer la règle de bon sens qui consiste à ne donner, à chaque utilisateur et à chaque service, que les droits dont il a besoin pour travailler. Dans notre exemple, le serveur d'impression n'avait certainement pas besoin de pouvoir ouvrir des connexions vers les machines de bureau !

Le cloisonnement consiste à définir des zones isolées les unes par rapport aux autres, auxquelles sont affectés des groupes de machines ou de serveurs.

Les zones seront reliées entre elles par un routeur filtrant ou un pare-feu qui autorisera seulement un certain nombre de flux identifiés à passer d'une zone à l'autre.

Définition des zones du réseau de Tamalo.com

Trois groupes de machines ont été identifiés. Ils seront classés dans trois zones différentes, comme l'indique le tableau 8-1.

Tableau 8–1 Zones du réseau tamalo.com

Postes de travail	Serveurs applicatifs internes SQL, LDAP, NFS, PRINT, IMAPS	Serveurs applicatifs publics DNS, HTTP, FTP

Définition des flux à l'extérieur et à l'intérieur du réseau de Tamalo.com

La connaissance des flux entre les différentes zones aide à configurer les filtres des pare-feu entre zones. C'est une étape essentielle dans la définition de la topologie réseau.

Chacune des trois zones est présentée dans les paragraphes suivants. Nous verrons les flux mis en œuvre vers et depuis l'extérieur, ainsi qu'entre les zones elles-mêmes.

Postes de travail

Aucun accès n'est nécessaire depuis l'extérieur de cette zone, que ce soit depuis la zone des serveurs internes ou depuis celle des serveurs publics. Ces machines doivent en revanche avoir accès sans restriction au monde extérieur, à tous les services possibles, y compris FTP en mode actif.

Serveurs applicatifs internes

Ces services ne doivent pas être accessibles depuis l'extérieur du réseau local. Les utilisateurs extérieurs pourront malgré tout accéder indirectement – jamais directement – à ces ressources par le biais des serveurs publics, grâce au modèle d'architecture trois tiers qui a été retenu. Certains services seront accessibles par les postes de travail et/ou par les serveurs publics.

Serveurs accessibles depuis l'extérieur et l'intérieur : DMZ

Ces services sont accessibles depuis l'extérieur et l'intérieur. Ils seront donc très exposés. Il est possible que l'un d'eux soit un jour ou l'autre l'objet d'une compromission. Pour limiter les conséquences de cette éventualité, ils seront rassemblés dans une zone particulière très surveillée, dont la prise de contrôle ne compromettrait pas tout le réseau. Cette zone s'appelle la DMZ : DeMilitarized Zone en anglais, ou simplement zone démilitarisée en français.

Topologie du réseau

Chacun des réseaux internes se voit affecter une plage d'adresses IP qui constitue un sous-réseau (*subnet*) indépendant. Pour que les machines puissent communiquer entre elles, un mécanisme de routage doit être mis en place entre ces différents sous-réseaux.

Il est également nécessaire d'effectuer un filtrage de paquets entre les sous-réseaux afin de ne laisser passer que les flux autorisés.

Ces deux fonctions de routage et de filtrage peuvent être réalisées par deux boîtiers séparés, un routeur et un pare-feu, ou bien rassemblées en un seul.

Les différentes zones du réseau étant définies, ainsi que les flux de données transitant entre elles, il reste à choisir la meilleure topologie de réseau pour les interconnecter.

Topologie à un seul pare-feu

Cette solution, représentée sur la figure 8-1, met en œuvre un seul pare-feu avec quatre interfaces réseau - ou seulement deux interfaces si on opte pour la mise en place de VLAN comme cela est décrit un peu plus loin dans ce chapitre.

Le pare-feu est connecté à chacun des trois réseaux internes et au réseau Internet.

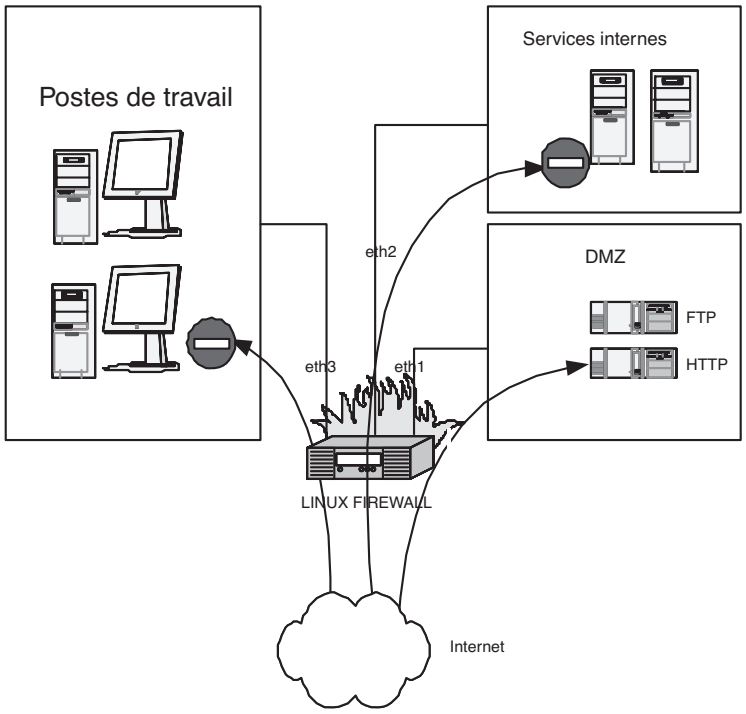


Figure 8–1
Topologie avec un seul pare-feu

Les avantages de cette topologie sont :

- son coût réduit ;
- sa facilité d'exploitation, puisqu'il n'y a à gérer qu'un seul boîtier pare-feu dans le réseau, un seul endroit ou déployer les règles de filtrage et de routage.

Son principal inconvénient est d'offrir une seule barrière de protection par rapport à l'extérieur.

Topologie à double pare-feu adoptée pour le réseau de Tamalo.com

La deuxième topologie mettant en jeu deux pare-feu est celle que nous avons finalement retenue pour Tamalo.com (voir la figure 8-2). Elle nécessite un budget un peu plus important, mais l'implémentation étant réalisée sur une base Linux, le surcoût de cette deuxième solution est finalement acceptable. L'avantage de ce choix sera mesurable en cas d'attaque réussie : il subsistera alors une protection des serveurs et des machines internes, même en cas de compromission de tous les éléments de notre réseau visibles depuis Internet.

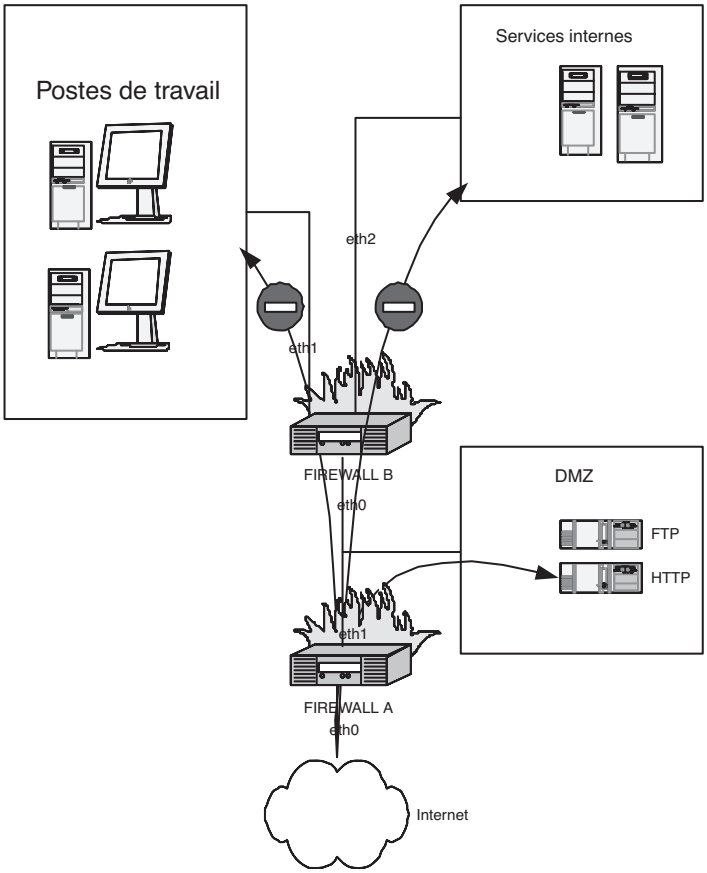


Figure 8–2
Topologie avec deux pare-feu et passage obligé par la DMZ

LE NEC PLUS ULTRA

Deux pare-feu différents en A et B

La méthode la plus sûre aurait consisté à mettre en place deux modèles différents de pare-feu en A et en B afin d'empêcher que la compromission de l'un permette celle de l'autre. En effet, si les pare-feu sont issus de constructeurs différents, ils ne présenteront pas les mêmes vulnérabilités.

En contre-partie, un tel choix nécessite l'administration de matériels différents, ce qui suppose des ressources humaines plus importantes.

Cette solution n'a pas été retenue pour Tamalo.com en raison de son coût et des difficultés d'exploitation qu'elle aurait engendrées.

Arrivé à ce stade de la compromission, le pirate aura besoin d'un peu de temps - précieux pour nous - pour scanner le réseau afin d'être en mesure de préparer une nouvelle attaque en direction des postes de travail ou des serveurs internes. En effet, avant d'avoir pris le contrôle de certaines machines, il ne disposait a priori d'aucun élément sur la topologie du réseau après la DMZ. Il lui restera donc à déterminer les services disponibles et leurs défaillances possibles. Si les outils de surveillance et d'audit du réseau que nous avons mis en place jouent leur rôle, cela devrait nous laisser un temps suffisant dans la plupart des cas pour détecter la compromission et la contrer rapidement en corrigeant le problème.

Détails de la configuration réseau de Tamalo.com

La configuration IP du réseau est décrite sur la figure 8-3.

DMZ

Les serveurs de la DMZ doivent être visibles depuis l'extérieur. Par conséquent, leurs adresses sont situées dans la plage d'adresses routables affectée au réseau de l'entreprise : `193.48.97.64/27`. `193.48.97.64` est l'adresse du réseau ; elle ne peut pas être affectée à une machine. `193.48.97.95` est l'adresse de *broadcast*, elle ne doit pas non plus être affectée à une machine. `255.255.255.224` est le masque de sous-réseau (*netmask*) : il comprend 27 bits à 1 comme l'indique la notation `/27`.

Nous avons affecté les adresses de la façon suivante :

- `193.48.97.91` est l'adresse du routeur par défaut et sera donc affectée à l'interface `eth1` du pare-feu A.
- `193.48.97.65` est l'adresse du serveur DNS primaire : `ns1.tamalo.com`
- `193.48.97.66` est l'adresse du serveur DNS secondaire : `ns2.tamalo.com`
- `193.48.97.67` est l'adresse du serveur Web : `www.tamalo.com`
- `193.48.97.68` est l'adresse du serveur SMTP : `smtp1.tamalo.com`
- `193.48.97.69` est l'adresse du serveur FTP : `ftp.tamalo.com`

Route par défaut

La route par défaut est dirigée vers `193.48.97.91`. Il est possible de positionner cette route de façon dynamique par la commande :

```
route add default gw 193.48.97.91
```

Route statique

Une route statique est ajoutée vers les serveurs internes :

```
route add net 192.168.154.0 netmask 255.255.255.0 gw
193.48.97.92
```

Proxy versus NAT

Le *proxy* et la traduction d'adresses sont deux mécanismes complémentaires pour limiter la visibilité du réseau interne.

Le NAT a l'avantage de s'appliquer quelle que soit l'application considérée, alors qu'il faut un *proxy* différent pour chaque type d'application. Dans la plupart des cas, la technique du NAT est donc plus rapide à mettre en œuvre pour masquer le réseau interne et protéger les postes de travail.

Pourtant, pour disposer de certaines fonctionnalités telles que l'utilisation de FTP en mode actif, la fonction de cache, la possibilité de filtrer les serveurs externes accessibles, il faut mettre en œuvre un proxy.

Proxy et NAT sont donc complémentaires.

Netfilter/IPtables

IPtables et Netfilter fournissent les fonctionnalités de filtrage disponibles sous Linux à partir du noyau 2.4.

Netfilter est implémenté au niveau des couches réseau du noyau Linux. Il effectue le filtrage proprement dit, tandis que IPtables fournit les commandes nécessaires à la programmation des filtres. IPtables est le successeur d'IPchains qui était disponible avec les versions 2.2 du noyau de Linux.

Fonctionnalités d'IPtables

IPtables fournit trois types de fonctionnalités :

- filtrage : statique (*stateless*) ou dynamique (*stateful*) et *connection tracking* ;
- NAT : traduction d'adresses IP, Source NAT, Masquerade, Destination NAT, redirection de port (il faut noter que IPtables ne permet pas à ce jour d'effectuer du NAT statique – un pour un – de façon simple) ;
- marquage et manipulation de paquets.

Parmi ces trois fonctionnalités, les deux premières ont été exploitées pour la sécurisation du réseau de Tamalo.com.

OUTILS Le projet Netfilter/IPtables

Rusty Russel est l'auteur et l'animateur du projet actuel. IPtables est disponible en standard dans les distributions de Linux.

- http://www.netfilter.org
- http://www.iptables.org

Tables et chaînes

À chacune des trois fonctions d'IPtables correspond une table qui sert à programmer cette fonction.

- FILTER : pour les règles de filtrage ;
- NAT : pour les règles de traduction d'adresses ;
- MANGLE : pour la manipulation et le marquage de paquets.

Chaque table contient un certain nombre de chaînes qui contiennent à leur tour une série de règles. La chaîne et les règles qu'elle contient s'appliquent à

un moment précis (avant le routage, après le routage…) du parcours du paquet (figure 8-13) et déterminent le futur du paquet (transmis, intercepté…).

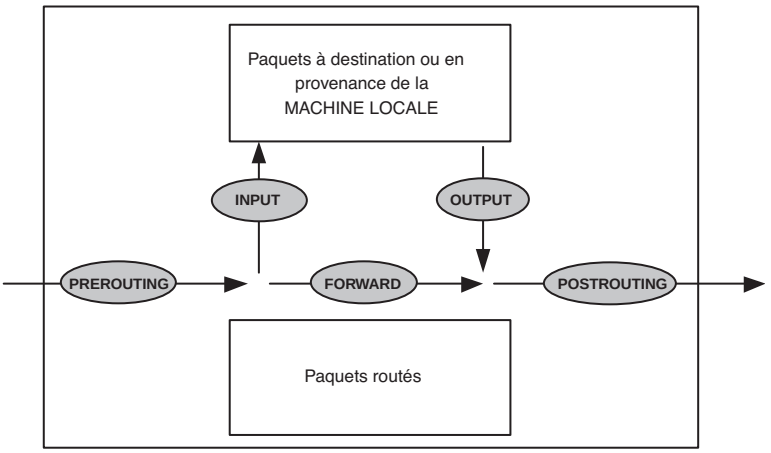


Figure 8–13
IPtables : les cinq points où peuvent s'appliquer les chaînes

Les chaînes peuvent s'appliquer à cinq moments dans la vie du paquet :

- `PREROUTING` : le paquet se présente sur une interface de la machine. Les chaînes appliquées sur `PREROUTING` seront exécutées avant d'analyser l'adresse IP du paquet pour prendre la décision de routage. Le cas d'utilisation typique est le DNAT.
- `POSTROUTING` : les chaînes appliquées sur `POSTROUTING` seront exécutées juste avant d'envoyer le paquet sur l'interface de sortie, alors que le paquet est déjà routé. Le cas d'utilisation typique est le SNAT.
- `FORWARD` : le paquet n'est pas destiné à la machine locale, mais il doit être relayé sur une autre interface. Les chaînes appliquées sur `FORWARD` ne concerneront pas les paquets à destination ou venant de la machine locale.
- `INPUT` : les chaînes appliquées sur `INPUT` concerneront tout paquet à destination de la machine locale.
- `OUTPUT` : les chaînes appliquées sur `OUTPUT` concerneront tout paquet en provenance de la machine locale.

Tableau 8–5 IPtables : points d'application des chaînes en fonction des tables

NAT	MANGLE	FILTER
PREROUTING	PREROUTING	
POSTROUTING	POSTROUTING	
	INPUT	INPUT
OUTPUT	OUTPUT	OUTPUT
	FORWARD	FORWARD

Écriture des règles

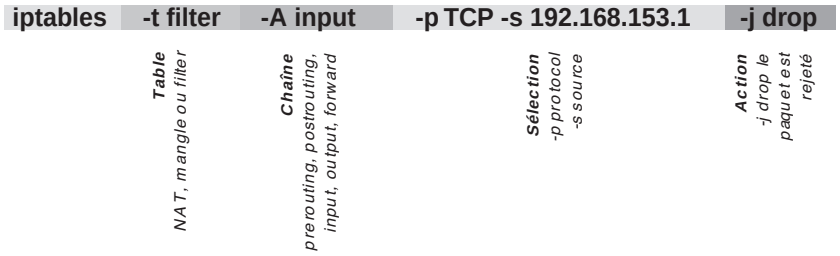


Figure 8–14
IPtables : écriture des règles

La figure 8–14 donne la syntaxe générale pour écrire une règle IPtables. La liste des actions possibles est indiquée ci-dessous :

`-j ACCEPT`	Le paquet est accepté.
`-j DROP`	Le paquet est rejeté.
`-j REJECT`	Le paquet est rejeté, l'expéditeur est averti de l'indisponibilité du service.
`-j QUEUE`	Le paquet est envoyé à une application.
`-j LOG`	Le paquet est envoyé au système « syslog ».
`-j MARK`	Le paquet est marqué.
`-j TOS`	Modifie le « Type Of Service » du paquet.
`-j MIRROR`	Renvoie le paquet à l'expéditeur.
`-j SNAT`	L'adresse source du paquet est translatée.
`-j DNAT`	L'adresse destination du paquet est translatée.
`-j MASQUERADE`	L'adresse source du paquet est translatée.
`-j REDIRECT`	Redirection d'un port vers un autre.

Suivi de connexion

Le suivi de connexion est un procédé qui maintient à jour une table des connexions qui ont traversé le pare-feu. Ce dernier prend la décision de filtrage en fonction du contenu de cette table qui constitue son état. Le pare-feu est alors qualifié de *state-full* (avec états). Sous Linux, cette table est stockée dans le répertoire `/proc/net/ip_conntrack`.

Journalisation

La journalisation est une fonction importante d'un pare-feu. Elle sera très utile dans la phase de mise au point des règles de filtrage pour étudier l'effet des filtres mis en place.

IPtables permet d'envoyer un message au système `syslog` de Linux quand un paquet coïncide avec une règle de type `LOG`. Cette règle doit être positionnée avant toutes les règles capables de stopper le parcours d'un paquet.

Lorsqu'une règle de type `LOG` est vérifiée, le processus de parcours des règles n'est pas arrêté. Il est possible de spécifier le niveau avec l'option `--log-level` (`info, notice, warning`...) et d'ajouter un préfixe au message grâce à l'option `--log-prefix`.

```
iptables -I -p tcp -s 192.168.155.1 -d 192.168.155.2 \
-dport 80 -j LOG --log-levelinfo info --log-prefix WEB
```

Pour les paquets concernés par cette règle, un message sera inscrit dans les fichiers de trace du `syslog`. Ils seront préfixés par le mot-clé `WEB` et auront un degré de priorité `INFO`.

Traduction d'adresses – NAT

Source NAT

La commande ci-dessous est utilisée pour remplacer les adresses sources par `193.48.97.70` :

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --
to 193.48.97.70
```

La commande suivante est utilisée pour remplacer les adresses sources par `193.47.97.70`, `193.47.97.71`, `193.47.97.72`, `193.47.97.73` :

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT \
--to 193.47.97.70-193.47.97.73
```

Masquerade

C'est une forme de « source NAT », mais dans ce cas l'adresse de translation est toujours la même, celle de l'interface de sortie.

```
iptables -t nat -I POSTROUTING -o eth0 -j MASQUERADE
```

Tous les paquets sortiront avec l'adresse IP de l'interface `eth0`.

Destination NAT

La commande ci-dessous remplace l'adresse de destination des paquets :

```
iptables -t nat -A PREROUTING -p tcp -d 193.48.97.80 \
-j DNAT --to-destination 192.168.154.
```

Filtrage

IPtables peut être utilisé efficacement pour configurer une machine Linux en routeur filtrant d'entrée de site. Les règles de filtrage devront être appliquées à tout paquet relayé par la machine (chaîne `FORWARD`), ainsi qu'à tout paquet destiné à la machine locale (chaîne `INPUT`).

Configuration IPtables des deux pare-feu Linux

Nous décrivons ci-après la configuration des pare-feu A et B du site Tamalo.com. Pour chacun d'eux, nous avons créé de nouvelles chaînes nommées respectivement `fwA` et `fwB`. La chaîne `fwA` (ou `fwB`) est appelée à la fois par les chaînes `INPUT` et `FORWARD`.

Firewall A

```
# Configuration de l'interface interne
ifconfig eth1 193.48.97.91 netmask 255.255.255.224 broadcast 193.48.97.95 up
# Activation du routage
echo 1 > /proc/sys/net/ipv4/ip_forward
# Mise en place de la route par défaut vers l'interface externe
route add default gw X.Y.Z.T (adresse sur le réseau d'interconnexion)
# Route vers le réseau des serveurs internes. Il vaut mieux définir cette route
au niveau de chaque machine de la DMZ plutôt qu'au niveau du routeur.
route add net 192.168.154.0 gw 193.48.97.92
# Flush les filtres
iptables -F
# Efface la chaîne fwA si elle existe déjà.
iptables -X fwA
# Crée une nouvelle chaîne fwA
iptables -N fwA
# Refuse les paquets en provenance de l'extérieur pour lesquels l'adresse
# source est définie dans notre réseau (lutte contre IP spoofing).
iptables -A fwA --in-interface eth0 --source 193.48.97.64/27 -j DROP
# Refuse les paquets en provenance de l'extérieur pour lesquels l'adresse
# source est définie dans les réseaux privés de classe A.
iptables -A fwA --in-interface eth0 --source 10.0.0.0/8 -j DROP
# Refuse les paquets en provenance de l'extérieur pour lesquels l'adresse
# source est définie dans les réseaux privés de classe B.
iptables -A fwA --in-interface eth0 --source 172.16.0.0/12 -j DROP
# Refuse les paquets en provenance de l'extérieur pour lesquels l'adresse
# source est définie dans les réseaux privés de classe C.
iptables -A fwA --in-interface eth0 --source 192.168.0.0/16 -j DROP
# Refuse les paquets en provenance de l'extérieur pour lesquels l'adresse
# source est définie dans le réseau de loopback.
iptables -A fwA --in-interface eth0 --source 127.0.0.0/8 -j DROP
# Refuse les paquets dont l'adresse source est une adresse de broadcast.
iptables -A fwA --in-interface eth0 --source 255.255.255.255 -j DROP
# Refuse les paquets dont l'adresse de destination est une adresse de broadcast.
iptables -A fwA --in-interface eth0 --destination 0.0.0.0 -j DROP
# Refuse les paquets dont l'adresse de destination est l'adresse de broadcast
# ou l'adresse du réseau de Tamalo.com.
iptables -A fwA --in-interface eth0 --destination 193.48.97.95 -j DROP
iptables -A fwA --in-interface eth0 --destination 193.48.97.64 -j DROP
# Refuse les paquets dont l'adresse source est une adresse de classe D (multicast).
iptables -A fwA --in-interface eth0 --source 224.0.0.0/4 -j DROP
```

```
# Refuse les paquets multicast si le protocole n'est pas UDP.
iptables –A fwA –-in-interface eth0 –-destination 224.0.0.0/4 –p ! UDP –j DROP
iptables –A fwA –-in-interface eth0 –-destination 224.0.0.0/4 –p UDP –j ACCEPT
# Accepte les connexions ESTABLISHED et RELATED
iptables -A fwA -m state --state ESTABLISHED,RELATED -j ACCEPT
# Accepte les ouvertures de connexion depuis la DMZ 193.48.97.64/27 vers l'extérieur
iptables -A fwA -m state --state NEW --source 193.48.97.64/27 -j ACCEPT
# Accepte les ouvertures de connexion vers le serveur Web port 80 situé en DMZ
iptables -A fwA -m state --state NEW --destination 193.48.97.67 -p TCP --dport 80 -j ACCEPT
# Accepte les ouvertures de connexion vers le serveur SMTP port 25 situé en DMZ
iptables -A fwA -m state --state NEW --destination 193.48.97.68 -p TCP --dport 25 -j ACCEPT
# Accepte les ouvertures de connexion vers le serveur FTP port 21
iptables -A fwA -m state --state NEW --destination 193.48.97.69 -p TCP --dport 21 -j ACCEPT
# Grâce à la technique de connection tracking implémentée dans IPtables, une connexion ftp
# passif sur 65000 sera acceptée parce que reliée, related, à la connexion initiale sur le
# port 21.
# Accepte les ouvertures de connexion vers les serveurs DNS port 53
iptables -A fwA -m state --state NEW --destination 193.48.97.65 -p UDP --dport 53 -j ACCEPT
iptables -A fwA -m state --state NEW --destination 193.48.97.66 -p UDP --dport 53 -j ACCEPT
iptables -A fwA -m state --state NEW --destination 193.48.97.65 -p TCP --dport 53 -j ACCEPT
iptables -A fwA -m state --state NEW --destination 193.48.97.66 -p TCP --dport 53 -j ACCEPT
# Accepte les icmp
iptables -A fwA -p icmp -j ACCEPT
# Rejette les autres connexions avec une erreur ICMP, port unreachable
# Cette règle étant la dernière, elle donne la politique par défaut qui consiste à tout
# rejeter sauf…, ce qui a été accepté précédemment.
iptables -A fwA -j REJECT
# Affiche les règles de filtrage de la chaîne fwA
iptables -L fwA
# Appelle la chaîne fwA à partir des chaînes INPUT et FORWARD
iptables -A INPUT -j fwA
iptables -A FORWARD -j fwA
# iptables –L
```

Firewall B

```
# Configuration de l'interface eth0 connectée à la DMZ
ifconfig eth0 193.48.97.92 netmask 255.255.255.224 broadcast 193.48.97.95 up
# Configuration de l'interface eth1 connectée aux postes clients
ifconfig eth1 192.168.155.254 netmask 255.255.255.0 broadcast 192.168.155.255 up
# Configuration de l'interface eth2 connectée aux services internes
ifconfig eth2 192.168.154.254 netmask 255.255.255.0 broadcast 192.168.154.255 up
# Activation du routage
echo 1 > /proc/sys/net/ipv4/ip_forward
# Mise en place de la route par défaut vers le firewall A
route add default gw 193.48.97.91
# Flush la table NAT si elle est déjà remplie
iptables -t nat -F
```

```
# NATe les adresses sources avec 3 adresses réelles : 193.48.97.91-92-93
iptables -t nat -A POSTROUTING -i eth1 -j SNAT --to-source 193.48.97.91-193.48.97.93
# Flush les filtres
iptables -F
# Efface la chaîne fwB si elle existe déjà
iptables -X fwB
# Cree une nouvelle chaîne fwB
iptables -N fwB
# Accepte les connexions ESTABLISHED et RELATED
iptables -A fwB -m state --state ESTABLISHED,RELATED -j ACCEPT
# Accepte les ouvertures de connexion de smtp1 vers imap sur le port SMTP
iptables -A fwB -m state --state NEW --source 193.48.97.68 --destination 192.168.154.140 -p TCP --dport 25 -j ACCEPT
# Accepte les ouvertures de connexion vers toutes les machines pour les réseaux internes
iptables -A fwB -m state --state NEW --source 192.168.154.0/24 --destination 0/0 -j ACCEPT
iptables -A fwB -m state --state NEW --source 192.168.155.0/24 --destination 0/0 -p TCP -j ACCEPT
# Accepte les icmp
iptables -A fwB -p icmp -j ACCEPT
# Rejette les autres connexions avec une erreur ICMP, port unreachable
iptables -A fwB -j REJECT
# Affiche les règles de filtrage
iptables -L fwB
# Appelle la chaîne fwB à partir des chaînes INPUT et FORWARD
iptables -A INPUT -j fwB
iptables -A FORWARD -j fwB
```

Configuration IPtables de chaque poste de travail

```
# Supprime les filtres actifs
iptables --flush
# Rejette tous les types de trafic
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
# Autorise tous les trafics sur l'interface loopback
iptables -A INPUT -i lo --source 127.0.0.1 --destination 127.0.0.1 -j ACCEPT
# Autorise l'ouverture de connexion de la machine locale vers le reste du monde
iptables -A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
# Autorise la réception de connexions initiées localement
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
# Autorise la réception de messages "ICMP echo request"
iptables -A INPUT --protocol icmp --icmp-type echo-request -j ACCEPT
# Rejette les demandes de connexion provenant de l'extérieur
iptables -A INPUT -m state --state NEW -j REJECT
```

Configuration IPtables du serveur SMTP

La configuration d'IPtables pour le serveur SMTP est transposable à l'ensemble des machines proposant un service réseau. Seuls le protocole réseau et le numéro de port sont susceptibles de varier suivant le service concerné.

```
# Supprime les filtres actifs
iptables --flush
# Rejette tous les types de trafic
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
# Autorise tous les trafics sur l'interface loopback
iptables -A INPUT -i lo --source 127.0.0.1 --destination 127.0.0.1 -j ACCEPT
# Autorise l'ouverture de connexion de la machine locale vers le reste du monde
iptables -A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
# Autorise la réception de connexions initiées localement
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
# Autorise la réception de connexions sur le port SMTP en TCP
iptables -A INPUT -m state --state NEW -p TCP --dport 25 -j ACCEPT
# Autorise la réception de messages "ICMP echo request"
iptables -A INPUT --protocol icmp --icmp-type echo-request -j ACCEPT
# Rejette les demandes de connexion provenant de l'extérieur
iptables -A INPUT -m state --state NEW -j REJECT
```

Marquage de paquets avec IPtables

La table MANGLE sert à modifier des paquets. Elle peut-être utilisée pour changer les champs TOS et TTL ou pour le marquage du paquet comme cela est décrit dans un exemple ci-après.

Modification des champs TOS, TTL

La figure 8-15 décrit comment modifier le champ TOS (Type Of Service) des paquets sortant de la machine. Cette possibilité peut être utilisée pour faire de la qualité de service en réservant des bandes passantes différentes par type de service.

Le TTL (Time to Live) peut être modifié en utilisant la cible TTL et les options `--ttl-set`, `--ttl-inc` ou `--ttl-dec` pour respectivement initialiser, incrémenter ou décrémenter d'une valeur donnée.

Notez que la modification par la table MANGLE des champs TOS ou TTL correspond à une véritable altération du paquet tandis que le marquage simple de celui-ci en utilisant l'option `--set-mark` s'effectue uniquement au niveau des tables du noyau du pare-feu Linux. Le marquage simple est donc exploitable seulement pendant la durée de vie du paquet dans la machine, mais pas en dehors de celle-ci.

À RETENIR TOS remplacé par DiffServ

L'octet TOS est désormais remplacé par le champ DiffServ, Differentiated Services. Cette nouvelle définition distingue trois probabilités – `low`, `medium`, `high` – que les paquets soient écartés (*drop*) pour chacune des quatre classes de service.

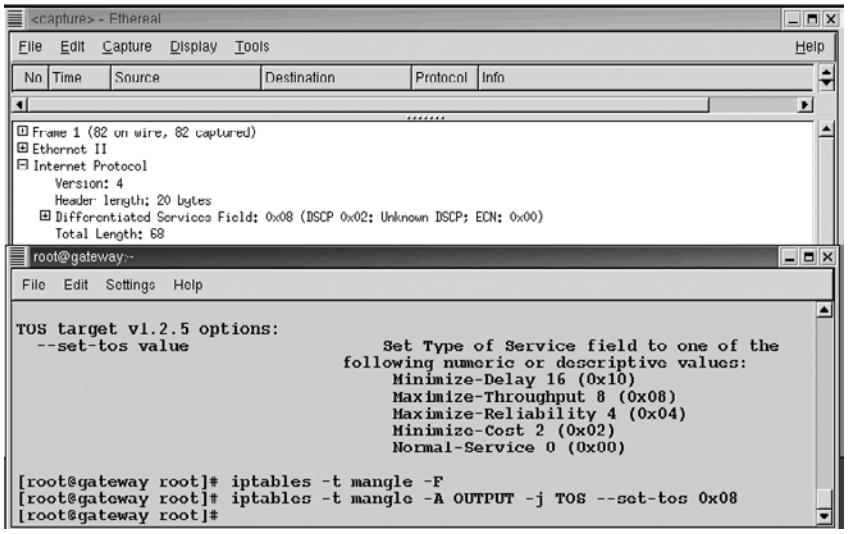


Figure 8–15 Modification du champ TOS et visualisation avec ethereal

Marquage simple du paquet

Le marquage effectué avec l'option `--set-mark` pourra être utilisé pour toute décision que doit prendre le pare-feu dans la suite de la vie du paquet. Une utilisation classique de ce type de marquage a pour objectif la qualité de service (QOS). Dans ce cas, le marquage est effectué avec IPtables pour signaler la priorité du paquet ; il est lu par `IProute2` qui lui affecte une bande passante en fonction de sa priorité.

L'exemple qui suit décrit comment utiliser le marquage des paquets pour réaliser un écran captif basé sur les adresses MAC.

L'adresse MAC de la source d'une trame n'est disponible qu'au moment où la trame entre dans le routeur, c'est-à-dire en PREROUTING, en INPUT et en FORWARD. En effet, dès que le paquet est dans le routeur, il est débarrassé de son en-tête Ethernet qui n'a plus de signification à ce stade. On voudrait pourtant pouvoir diriger - grâce à une substitution de type DNAT effectuée en POSTROUTING - les paquets vers la sortie si leur adresse MAC est autorisée ou vers un écran captif dans le cas contraire. Une solution consiste à utiliser la table MANGLE pour marquer « 1 » les paquets contenus dans les trames dont les adresses MAC sont autorisées et « 0 » dans le cas contraire. Il sera alors possible de prendre la décision de redirection NAT, non pas en fonction de l'adresse MAC, qui n'est pas disponible au stade du POSTROUTING, mais en fonction du marquage effectué à « 0 » ou « 1 ».

DANGER Incrémentation du TTL

Attention, n'augmentez jamais la valeur du champ TTL d'un paquet qui sort de votre réseau au risque de redonner vie à des paquets zombies. En effet, le rôle du TTL (Time to Live) est d'éviter qu'en cas d'erreur de routage, des paquets puissent tourner en rond dans Internet. Le TTL étant décrémenté à chaque routeur, ces paquets seront écartés (*drop*) dès que leur TTL sera à 0.

En revanche, il peut être envisagé d'incrémenter le TTL des paquets entrants pour masquer l'existence du pare-feu à la commande `traceroute`.

- http://www.linuxexposed.com/Articles/Networking/Traffic-shaping-and-bandwidth-management.html

Script de commandes IPtables

```
#!/bin/sh
iptables="/sbin/iptables"
################### NAT ###########################
$iptables -t nat -F
$iptables -t mangle -F
# NAT est l'adresse du routeur NAT de sortie
NAT=193.48.97.91
# CONF adresse du réseau sur lequel s'applique le filtrage par
adresses MAC
CONF=192.168.156.0/24
# Marque 1 les paquets autorisés, 0 les paquets non autorisés
$iptables -t mangle -A PREROUTING -j MARK --set-mark 0x0
# /etc/MAC-autorise contient les machines autorisées # indique
un commentaire
grep -v "#"  /etc/MAC-autorise | sort -u | \
( while read MAC; do
  echo $MAC;
  $iptables -t mangle -A PREROUTING -m mac --mac-source $MAC -s
$CONF -j MARK --set-mark 0x1;
done)
# les paquets TCP non autorisés (marqués 0) sont dirigés vers
l'écran captif
$iptables -t nat -A PREROUTING -m mark --mark 0x0 -p tcp -s
$CONF -j DNAT --to 192.168.2.24:80
# les paquets autorisés (marqués 1) sont natés vers la sortie
$iptables -t nat -A POSTROUTING -m mark --mark 0x1 -s $CONF -j
SNAT --to-source $NAT
```

Pare-feu transparent, mode bridge

Un pare-feu transparent s'insère dans un réseau existant sans modification sur le routeur ni sur les postes et serveurs installés ! Ainsi, sa mise en place est grandement facilitée. De plus, en cas de panne ou de dysfonctionnement du pare-feu, il suffit de l'enlever pour revenir à l'état initial.

Positionnement du pare-feu transparent

Le pare-feu est inséré entre le routeur initial et le reste du réseau comme indiqué sur la figure 8-16. Il voit ainsi passer l'ensemble du trafic entre les réseaux interne et externe.

Adressage IP

Dans notre exemple, le routeur a comme adresse `193.48.97.1`. Nous allons donc réserver les quatre (obligatoirement une puissance de deux) premières adresses du réseau, `193.48.97.0` à `193.48.97.3`, pour la zone d'intercon-

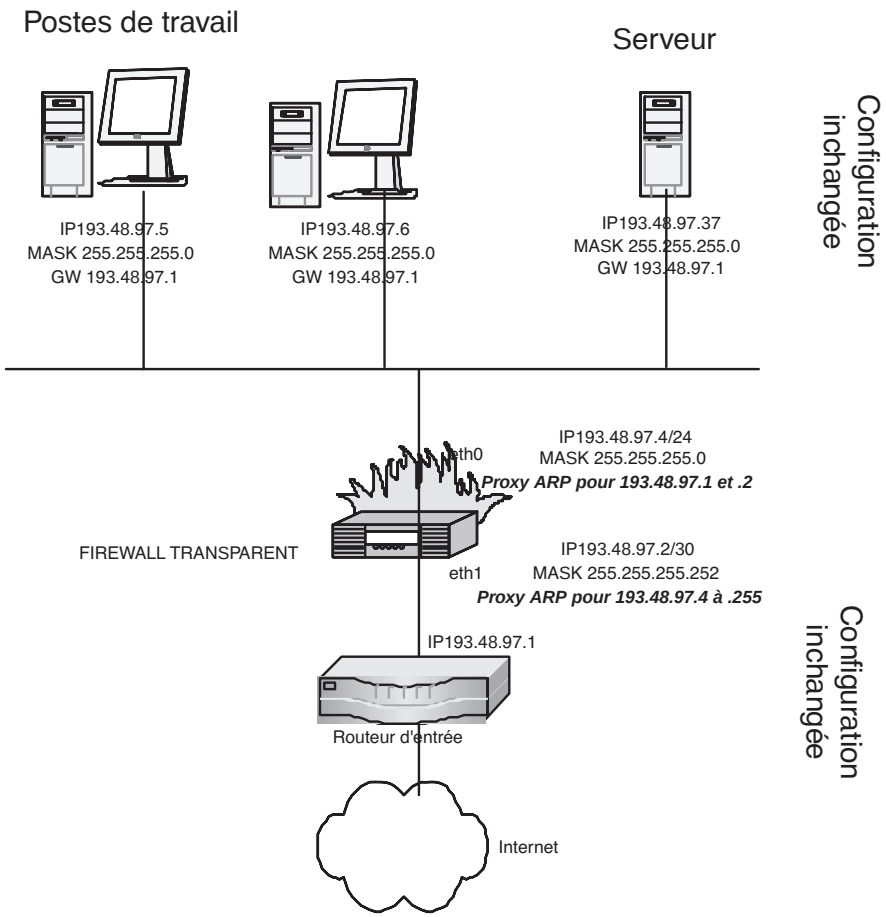


Figure 8–16 Pare-feu transparent

nexion entre le routeur et le pare-feu. Notez que cette zone pourra contenir des serveurs ; elle constituera alors une véritable DMZ.

- `193.48.97.0` est l'adresse du réseau.
- `193.48.97.1` est l'adresse – inchangée - du routeur
- `193.48.97.2` est l'adresse du pare-feu côté DMZ.
- `193.48.97.3` pourra être utilisée pour un serveur qui serait placé dans la DMZ.

Proxy ARP

Le pare-feu s'insère dans le réseau sans modification de celui-ci grâce à la technique du proxy ARP :

- Côté DMZ, le pare-feu est proxy ARP pour l'ensemble des machines du réseau interne, soit `193.48.97.4` à `193.48.97.255`.

- Côté interne, le pare-feu est proxy ARP pour les quatre adresses de la DMZ, `193.48.97.0` à `193.48.97.3`.

Ainsi, chaque fois qu'une machine interne demande, par une requête ARP, l'adresse MAC du routeur `193.48.97.1`, le pare-feu donne l'adresse MAC de sa propre interface `eth0`. Réciproquement, chaque fois que le routeur demande, par une requête ARP, l'adresse MAC d'une machine interne, le pare-feu lui fournira l'adresse MAC de sa propre interface `eth1`.

Le pare-feu se trouve donc en position de « Man in the Middle » ; il ne lui reste plus qu'à router le trafic entre les réseaux `193.48.97.0/30` et `193.48.97.0/24` pour rétablir la connectivité. Ce faisant, il sera possible d'appliquer des règles de filtrage entre les machines internes et externes.

Configuration pratique du pare-feu transparent

Configuration en proxy ARP coté DMZ

```
arp -v -i eth0 -Ds 193.48.97.1 eth0 pub
arp -v -i eth0 -Ds 193.48.97.2 eth0 pub
```

Configuration en proxy ARP coté interne

```
arp -v -i eth1 -Ds 193.48.97.128 eth1 netmask 255.255.255.128
pub
arp -v -i eth1 -Ds 193.48.97.64 eth1 netmask 255.255.255.192
pub
arp -v -i eth1 -Ds 193.48.97.32 eth1 netmask 255.255.255.224
pub
arp -v -i eth1 -Ds 193.48.97.16 eth1 netmask 255.255.255.240
pub
arp -v -i eth1 -Ds 193.48.97.8 eth1 netmask 255.255.255.248 pub
arp -v -i eth1 -Ds 193.48.97.4 eth1 netmask 255.255.255.252 pub
```

Configuration des interfaces et mise en place des routes

```
ifconfig eth0 193.48.97.4 netmask 255.255.255.0 broadcast
193.48.97.255
route add -net 193.48.97.0

ifconfig eth1 193.48.97.2 netmask 255.255.255.252 broadcast
193.48.97.3
route add -net 193.48.97.0

route add default gw 193.48.97.1
```

Configuration IPtables

Les règles sont les mêmes que pour un pare-feu classique, sauf qu'il faut traiter d'abord les adresses de la « DMZ » :

```
# Refuse les ouvertures de connexion depuis la DMZ 193.48.97.0/30
iptables -A FORWARD -m state --state NEW -s 193.48.97.0/24 -j
REJECT
# Accepte les ouvertures de connexion depuis la production
193.48.97.0/24
iptables -A FORWARD -m state --state NEW -s 193.48.97.0/24 -j
ACCEPT
# Accepte les retours de connexions ESTABLISHED et RELATED
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j
ACCEPT
# Rejette les autres connexions avec une erreur ICMP port unreachable
iptables -A FORWARD -j REJECT
```

Sécurité du réseau sans fil

Le réseau sans fil ou Wi-Fi, pour *Wireless Fidelity*, est devenu incontournable par la souplesse qu'il offre aussi bien pour l'accueil des nomades que pour la mobilité des personnels du site. Il ne doit pourtant pas être déployé sans prendre en compte plusieurs impératifs de sécurité.

Risque d'accès frauduleux au réseau

Les réseaux Wi-Fi posent le problème de l'authentification pour l'accès au réseau. En effet, pour se connecter frauduleusement à un réseau filaire, le pirate doit d'abord franchir les mécanismes de contrôle d'accès aux bâtiments afin de s'introduire physiquement dans les locaux. Ce faisant, il prend le risque d'être surpris en flagrant délit, par exemple connecté à une prise informatique. En comparaison, le pirate du Wi-Fi installé dans sa voiture, devant une entreprise, occupé à pianoter sur son ordinateur portable, agit en toute impunité.

Les risques d'attaque du réseau local de l'entreprise à partir du réseau Wi-Fi seront limités en isolant le sans fil dans un segment à part. Il est alors possible d'appliquer les mêmes règles de filtrage depuis ce segment que depuis l'extérieur. Le réseau de l'entreprise n'a ainsi pas plus à craindre du Wi-Fi que de n'importe quel site externe.

Le danger est plus difficile à contrer dans le cas où le pirate rebondirait à partir de l'accès Wi-Fi de notre entreprise pour s'attaquer à un site externe. En effet, dans l'hypothèse où le site attaqué déposerait une plainte, notre

responsabilité se trouverait engagée et il ne nous serait pas possible de prouver par les *logs* que l'attaque vient de l'extérieur de notre réseau.

Il est donc impératif de se protéger contre un accès abusif au réseau sans fil. Malheureusement, à l'exception de l'authentification 802.1X décrite ci-après, aucune des techniques à notre disposition aujourd'hui (WEP, identification par adresse Mac) n'est vraiment sûre.

Les limites du WEP

Les systèmes sans fil intègrent un chiffrement appelé WEP (Wired Equivalent Privacy). Comme son nom l'indique, ce système de chiffrement est câblé dans le matériel des cartes et des bornes sans fil. Une conséquence est que pour le remplacer par une technique plus performante il faudra renouveler l'ensemble des cartes et bornes du réseau ! Ainsi, la technique WPA, Wireless Protected Access, issue de la norme 802.11i, prometteuse du point de vue de la sécurité n'est pas compatible avec les « vieilles » cartes et bornes à base de WEP.

Le chiffrement WEP est utilisé à la fois pour protéger l'accès au réseau et pour chiffrer les communications :

- Pour limiter l'accès frauduleux au réseau, bien que le WEP ait ses limites, il est recommandé de le mettre en place de façon systématique… Il compliquera la vie des pirates, du moins pour les moins téméraires, puisqu'une clé WEP statique sera cassée en moins de 10 minutes par un attaquant doté des bons outils.
- Pour chiffrer les communications, il faut lui préférer SSH ou l'utilisation systématique de VPN.

Une parade à la faiblesse des clés WEP consiste à utiliser des clés dynamiques, remplacées avec une fréquence inférieure à 10 minutes.

RÉFÉRENCE RADIATOR

RADIATOR est une implémentation Open Source commerciale compatible avec le serveur d'authentification RADIUS, Cette version est stable et elle est dotée de nombreuses fonctionnalités.
http://www.open.com.au/radiator/

RÉFÉRENCE FreeRADIUS

Une implémentation libre FreeRADIUS compatible avec le serveur d'authentification RADIUS est disponible sur le site : http://www.freeradius.org

Le protocole 802.1X

Le protocole 802.1X a été défini par l'IEEE pour prendre en charge les phases d'authentification en vue d'autoriser l'accès au réseau. Ce protocole peut être mis en place pour l'accès à un réseau sans fil mais également pour un réseau filaire.

Le serveur, *authenticator system*, 802.1X est implémenté par le point d'accès (AP) au réseau : commutateurs ou bornes sans fil.

Le client, ou *supplicant*, est implémenté par la couche TCP/IP de la machine qui se connecte au réseau. Pour Linux, un client est disponible sur le site http://www.open1x.org.

La mise en place d'une authentification 802.1X nécessite également un serveur d'authentification, tel que FreeRADIUS, qui sera interrogé par le commutateur ou la borne sans fil pour accorder ou non l'accès.

Comme le montre la figure 8-17, tant que le client n'est pas authentifié, seuls les paquets à destination du serveur d'authentification sont transmis par le point d'accès.

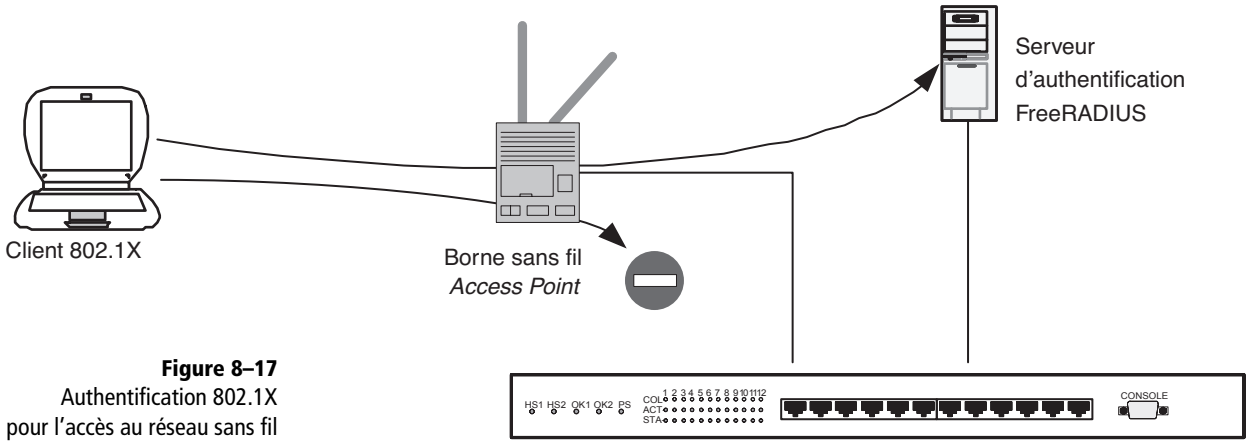


Figure 8–17
Authentification 802.1X
pour l'accès au réseau sans fil

La méthode utilisée pour l'authentification est définie par le protocole EAP (Extended Authentication Protocol). Différents types d'authentification sont possibles :

- EAP TLS – Le client et le serveur s'authentifient mutuellement par le biais de certificats électroniques. Ce mode est le plus sûr, mais il nécessite la mise en place d'une Infrastructure à gestion de clés (PKI) pour la distribution des certificats électroniques à tous les utilisateurs du réseau.
- EAP TTLS – Le serveur s'authentifie par un certificat tandis que le client s'authentifie par un mot de passe. Seuls les serveurs doivent posséder des certificats, ce qui allège la mise en œuvre par rapport à ce qui est décrit précédemment. Dès que le serveur est authentifié par le client, un tunnel TLS est créé entre le client et le serveur qui protège le mot de passe envoyé par le client.
- EAP MD5 – Le client s'authentifie par un mot de passe, mais il n'y a pas d'authentification du serveur. Le seul chiffrement possible utilise les clés WEP non dynamiques. De plus, cette méthode est sensible aux attaques qui consistent à rejouer une session. Pour ces raisons, elle ne doit pas être recommandée sur un réseau sans fil.
- Si l'authentification a réussi, le point d'accès autorise les trames émises par le client vers le reste du réseau ; dans le cas contraire, le client est isolé du réseau.

Risque d'écoute du réseau

Une borne sans fil diffuse les trames qu'elle émet à tous les postes qui lui sont associés. Les possibilités d'écoute sont donc très importantes ; on peut considérer que le risque est du même niveau qu'avec un bon vieux *hub* (concentrateur) ! De ce fait, il est indispensable de déployer sur ce type de

réseau des techniques de chiffrement des communications. Plutôt que de faire confiance au chiffrement WEP disponible sur les cartes, il faut préférer des protocoles applicatifs comme SSH, ou réseau comme IPsec qui est mis en œuvre si on établit un VPN.

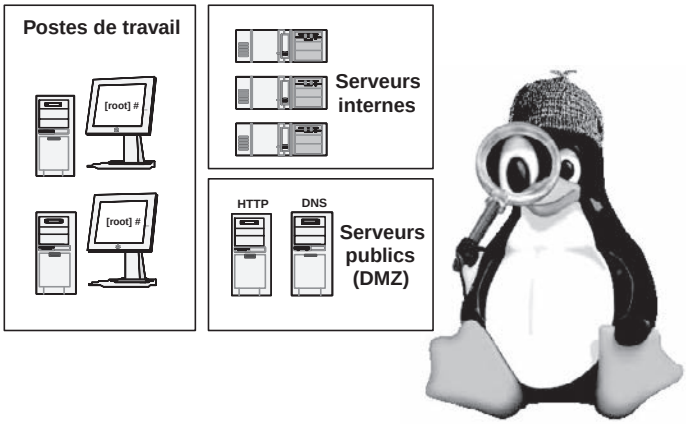
Notez par ailleurs que la possibilité d'écoute existe pour toute personne située à portée de l'émetteur et capable de s'associer à la borne, qu'elle soit hors ou dans les locaux de l'entreprise !

En résumé…

La topologie du réseau est un élément clé de la sécurité du site. La connaissance précise des flux à l'intérieur du réseau permettra d'identifier les zones qui peuvent utilement être créées. Parmi ces zones, la DMZ ou zone démilitarisée abritera les services accessibles depuis Internet. Elle devra faire l'objet d'une surveillance particulière car elle est visible depuis l'extérieur, donc plus exposée. Les moyens à mettre en œuvre pour assurer cette surveillance seront décrits au chapitre 9.

Pour ce qui concerne l'implémentation du ou des pare-feu, Linux doté du tandem Netfilter/IPtables offre des fonctionnalités très complètes, aussi bien pour les fonctions de NAT que de filtrage. Avec cet outil, la mise en place de règles de base simples procure déjà une excellente protection.

chapitre 9


Postes de travail
[root] #
[root] #
Serveurs internes
HTTP
DNS
Serveurs publics (DMZ)

```
root@rh71: /root/SNORT
File  Edit  Settings  Help
# portscan: detect a variety of portscans
# ----------------------------------------
# portscan preprocessor by Patrick Mullen <p_mullen@linuxrc.net>
# This preprocessor detects UDP packets or TCP SYN packets going to
# four different ports in less than three seconds. "Stealth" TCP
# packets are always detected, regardless of these settings.

preprocessor portscan: $HOME_NET 4 3 portscan.log

# Use portscan-ignorehosts to ignore TCP SYN and UDP "scans" from
# specific networks or hosts to reduce false alerts. It is typical
# to see many false alerts from DNS servers so you may want to
# add your DNS servers here. You can all multiple hosts/networks
# in a whitespace-delimited list.
#
#preprocessor portscan-ignorehosts: 0.0.0.0
"snort.conf" 486L, 18487C written
```

Figure 9–10 Configuration de Snort pour détecter les scans

```
root@gateway:~
File  Edit  Settings  Help
[root@gateway root]# nmap -P0 -p1-1023 dns.tamalo.com/27

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Strange error from connect (101):Network is unreachable
All 1023 scanned ports on tamalo.com (193.48.97.64) are: closed
The 1 scanned port on dns.tamalo.com (193.48.97.65) is: closed
Interesting ports on www.tamalo.com (193.48.97.66):
```

Figure 9–11 Lancement du scan avec NMAP

```
root@rh71: /root/SNORT
File  Edit  Settings  Help
[root@rh71 SNORT]# cat /var/log/snort/alert
[**] [100:1:1] spp portscan: PORTSCAN DETECTED from 193.48.97.94 (THRESHOLD
 4 connections exceeded in 3 seconds) [**]
06/23-23:16:24.490825

[root@rh71 SNORT]#
```

Figure 9–12 Alerte Snort en cas de scan

Le pot de miel

Comme son nom l'indique, le pot de miel, en anglais *honeypot*, est un système leurre, mis en place par un administrateur système, destiné à attirer la convoitise des pirates. L'objectif du stratagème est d'observer les crackers pendant qu'ils tentent de prendre le contrôle d'une machine afin de bien connaître leurs outils et leurs méthodes dans le but d'arriver à mieux les contrer.

Le pot de miel est donc configuré pour présenter un certain nombre de défaillances dont on espère qu'elles vont être détectées puis exploitées par un pirate.

RÉFÉRENCES **Pot de miel**

- http://www.citi.umich.edu/u/provos/honeyd/
- http://project.honeynet.org/

OUTIL Deception toolkit

Ce kit fournit un certain nombre de faux services, afin de simuler le fonctionnement par exemple d'un serveur de messagerie ou d'un serveur HTTP présentant des défaillances. Ces leurres ont pour objectif de faire perdre du temps au pirate qui doit choisir une vulnérabilité parmi tous les services disponibles puis l'exploiter. Si par exemple la vulnérabilité lui donne accès à un (faux) fichier `passwd` il devra encore faire tourner *crack* sur ce fichier avant de constater que les mots de passe *craqués* sont inutilisables. De plus, dès que le pirate s'attaque à un faux service, sa présence est connue de l'administrateur qui dispose en particulier de son adresse IP et pourra organiser la riposte en filtrant cette adresse. Enfin les statistiques plaident pour ce type de solution : si 50 % des services sont des faux et si 80 % parmi les vrais ne présentent pas de vulnérabilité, il ne reste que 10 % de chance au pirate d'attaquer un service vulnérable !

▸ http://www.all.net/dtk/index.html

Pendant toutes les étapes de la compromission, les actions du pirate seront tracées le plus finement possible :

- Avant l'intrusion, le système de détection d'intrusion permettra d'identifier la signature de l'attaque et de capturer le trafic réseau.
- Au moment de l'intrusion, les *logs* produits par chaque service seront envoyés à un serveur distant grâce à la mise en place de *logs* centralisés avec syslog-ng par exemple.
- Une fois dans la place, un shell `root`, tel que le shell opérateur `osh` disponible avec Debian, permettra de tracer les actions du pirate dans le but de connaître les *rootkits* installés.
- Quand il estimera que le pirate est allé assez loin, l'administrateur de la machine, prévenu dès la première phase par les alertes fournies par le système de détection d'intrusion, isolera la machine du réseau.
- Il ne restera qu'à effectuer une analyse *a posteriori* (analyse *forensic*) avec le même type d'outils que ceux qui ont étés décrits au chapitre 3 pour recueillir un maximum d'informations sur la compromission.

Tableau de bord de la sécurité

L'objectif d'un tableau de bord de la sécurité informatique est de faire remonter des informations pertinentes aux décideurs concernant la sécurité du système d'information. La mise en place d'un tableau de bord implique la définition d'un certain nombre d'indicateurs mesurant chacun un écart par rapport à un objectif élémentaire de sécurité. L'ensemble de ces données est pondéré et moyenné afin de remonter sous une forme synthétique une information claire aux dirigeants de l'entreprise. Pour atteindre les objectifs de sécurité un certain nombre de plans d'action seront alors mis en œuvre. Le tableau de bord permettra de mesurer leur efficacité.

Les indicateurs de sécurité

Il existe différents types d'indicateurs de sécurité. Les premiers mesurent un écart par rapport à un objectif de sécurité et donnent une idée de la vulnérabilité du système d'information. D'autres indicateurs sont mis en place pour mesurer la menace qui pèse sur le système d'information. Enfin, il est très utile de remonter des informations sur la sensibilisation des différents intervenants, ce qui donne la mesure de l'élément clé que constitue le facteur humain.

La valeur des indicateurs peut être obtenue par un traitement manuel ou automatique.

Mesure de la vulnérabilité

Ces indicateurs permettent de mesurer la vulnérabilité du système d'information. Ils sont le plus souvent renseignés manuellement, suivant un protocole qui doit être le plus précis possible. Ils permettent de connaître l'état du système d'information à un certain moment et sont très utiles pour mesurer l'efficacité d'un plan d'action.

Exemples d'indicateurs de vulnérabilités :

- pourcentage de machines accessibles de l'extérieur dont le service `openssh` est à jour dans la version recommandée : X.Y.Zp1 ;
- pourcentage de machines dotées du logiciel `tripwire` pour la prise d'empreinte et la détection de compromission ;
- pourcentage de services jugés dangereux par un *scan nessus*.

Mesure de la menace

Ces indicateurs permettant de mesurer la menace sont le plus souvent statistiques et peuvent en général être remontés de façon automatisée. Ils n'ont d'intérêt que comparés aux valeurs précédentes du même indicateur. La procédure de collecte doit donc être stable et bien documentée.

Exemples d'indicateurs de la menace :

- Nombre de *scans*/jour, *top ten* des services *scannés*. Ces indications sont généralement remontées de façon automatisée à partir des *logs* fournis par le routeur d'entrée de site ou par une sonde *snort* placée à l'intérieur du site.
- Nombre de connexions par jour, nombre de connexions en dehors des heures ouvrables. Cette indication est remontée par les *syslog*.
- Volume de transfert réseau, *top ten* machine, *top ten* service.
- Nombre d'avis de sécurité en provenance du CERT. Parmi ces avis, comptabilisez également le nombre d'alertes de sécurité.

Facteur humain

La mesure de l'état de la sensibilisation des troupes au risque informatique est certainement la plus difficile à réaliser de façon pratique. Malgré tout, un certain nombre d'éléments existent qui peuvent être mesurés comme :

- le nombre de participants à des stages de sécurité ;
- le nombre de participants à des séminaires ou des journées d'information organisés par l'entreprise sur la sécurité ;
- le nombre de collaborateurs ayant quitté l'entreprise, qui sera comparé utilement au nombre d'accès informatiques fermés pendant la même période ;
- la compatibilité de divers incidents : perte de carte d'accès à la salle informatique, vols de portables, etc.

Synthèses des indicateurs dans un tableau de bord

Le tableau de bord a pour rôle l'affichage synthétique de l'ensemble des données remontées par les indicateurs. Suivant la taille de l'entreprise et le nombre des intervenants, il peut y avoir plusieurs niveaux de tableaux de bord.

Le tableau de bord doit faire apparaître de la manière la plus synthétique possible l'état des indicateurs qui ont été définis. Ces indicateurs seront comparés à l'objectif de sécurité qui a été fixé pour donner une information lisible, par exemple une signalisation du type feu vert – feu orange – feu rouge. On y ajoutera toute information utile telle que des pointeurs sur les plans d'action déployés pour atteindre l'objectif, le budget alloué à ces actions, les personnes impliquées dans leur mise en œuvre.

En résumé…

Plusieurs outils sophistiqués existent désormais dans le monde du logiciel libre et permettent une analyse très fine et une surveillance accrue des réseaux. Ces outils ont atteint un bon niveau de maturité. En fonction des enjeux de la sécurité informatique de l'entreprise, il sera possible de baser la surveillance sur la simple utilisation des logiciels tels que le syslog ou tripwire. L'administrateur réseau tirera également un large bénéfice de l'utilisation de MRTG pour la métrologie du réseau.

Les outils d'audit et les systèmes de détection d'intrusion génèrent une quantité importante d'informations, qui peut vite s'avérer indigeste pour les administrateurs réseau. L'utilisation systématique de ces outils doit donc être réservée aux zones sensibles du réseau, où une surveillance accrue est indispensable.

chapitre 10

RÉPUBLIQUE FRANÇAISE
CARTE NATIONALE D'IDENTITÉ N
Nationalité Française
BB
Nom :
??????????
AUTHENTIFIER
AUTHENTIFIER
AUTHENTIFIER
AUTHENTIFIER
AUTHENTIFIER
AUTHENTIFIER

Gestion des comptes utilisateur et authentification

Identifier et authentifier ses utilisateurs est une fonction vitale pour le système d'information. Trois outils de gestion centralisée et d'authentification seront présentés : NIS, LDAP et Kerberos.

SOMMAIRE

- Gestion centralisée des comptes utilisateur
- Authentification et identification
- Linux et authentification
- Network Information Service
- OpenLDAP
- Kerberos
- Authentification et interopérabilité

MOTS-CLÉS

- Authentification
- Identification
- PAM
- NSS
- NIS
- LDAPDirectory Access Protocol)
- Kerberos

L'authentification est un des maillons essentiels du système d'information. Avec un environnement informatique réduit à la fois en nombre de postes de travail mais également en nombre d'utilisateurs, la mise en œuvre du service d'authentification est en général limitée à la création de groupes et de comptes. Il n'y a pas de déploiement spécifique et les mécanismes prévus de base dans les systèmes d'exploitation, tout particulièrement Linux, suffisent. Dans un environnement plus complexe, comme celui de la société Tamalo, il devient nécessaire de déployer des outils pour une gestion centralisée plus souple des comptes utilisateur et proposant des mécanismes de sécurité plus importants.

Nous étudierons dans ce chapitre les possibilités offertes aux administrateurs système de Tamalo pour mettre en œuvre une gestion des comptes utilisateur centralisée et pour l'intégrer aux mécanismes d'authentification du système Linux dans des conditions de sécurité acceptables.

Gestion centralisée des comptes utilisateur

Le nombre de collaborateurs de la société Tamalo étant en hausse, la gestion traditionnelle des comptes utilisateur, c'est-à-dire localement à chacun des systèmes, devient lourde et inadaptée à une vue unique et cohérente. Il devient en effet difficile de garantir l'homogénéité de la définition des comptes utilisateur sur l'ensemble du parc. L'arrivée d'une nouvelle personne implique un travail pénible de création de compte sur l'ensemble des serveurs et des postes de travail.

Tamalo doit adopter une solution technique pour répondre à son besoin de gestion centrale et d'authentification des comptes utilisateur. Même si ses équipes techniques n'ont pas encore arrêté leur choix, trois logiciels sont candidats pour répondre à ce besoin. NIS (Network Information Service) est le plus intégré au système Unix. LDAP (Lightweight Directory Access Protocol) est un service de répertoire dont l'implémentation libre OpenLDAP est particulièrement populaire dans le monde du logiciel libre. Enfin, Kerberos, réputé pour la robustesse de ses mécanismes d'authentification, répond quant à lui à une problématique forte de sécurité. Ces trois produits seront décrits et comparés dans ce chapitre et leur mise en œuvre sera présentée à l'annexe B.

Identification et authentification

- *Identification* : action qui consiste à renseigner l'identité sans aucune action de vérification. L'utilisateur s'identifie lorsqu'il saisit son identifiant sur la mire de connexion d'un système. De même, il y a identification lorsque le système fait la correspondance entre un identificateur Unix (UID) et un nom de compte.
- *Authentification* : dans le contexte informatique, l'authentification est l'action qui garantit l'identité d'un utilisateur, d'un service ou d'un ordinateur. Cette opération est souvent réalisée en fournissant un mot de passe ou encore par la présentation et la validation d'un certificat électronique.

Authentification et identification

La problématique de la gestion des comptes utilisateur porte en général sur le déploiement d'outils capables d'agir de manière centralisée. Il est aisé de comprendre l'inconfort d'une situation où des dizaines, voire des centaines

de postes de travail doivent être reconfigurés pour ajouter le compte d'un nouvel utilisateur. Il ne faut néanmoins pas oublier le but principal de la base de données contenant les informations de comptes, qui est d'authentifier et d'identifier les utilisateurs du système en toute sécurité, que la gestion en soit centrale ou non.

Pourquoi authentifier ?

L'accès à une ressource informatique, c'est-à-dire à un service ou encore à un ordinateur, doit être contrôlé. Pour comparaison, seul le détenteur des clés de votre voiture peut s'en servir et c'est en général mieux ainsi. L'authentification garantit que l'utilisateur qui s'identifie n'est pas un usurpateur qui cherche à détourner les ressources que vous avez mises en place. Une fois cette authentification réalisée, il est possible de différencier les utilisateurs entre eux. Une gestion de privilèges devient donc possible et permet aux différents systèmes de contrôle d'accès de limiter l'utilisation des ressources aux personnes dûment habilitées.

Le système d'authentification

Un système d'authentification propose une gestion des données utilisateur dans laquelle sont associées des informations permettant de contrôler l'identité de chacun, c'est-à-dire au minimum un identifiant et un secret. Plus concrètement, on a coutume de manipuler un nom de compte et un mot de passe. Le système d'authentification doit également offrir un protocole fiable de vérification des identités. Son fonctionnement est binaire, comme cela est représenté sur la figure 10-1.

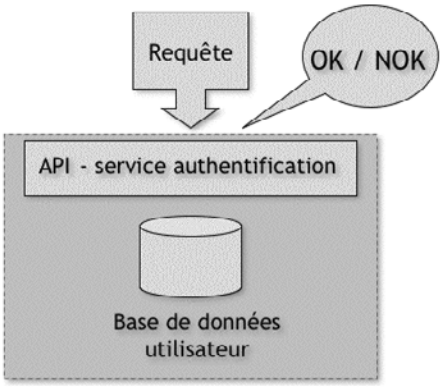


Figure 10–1
Fonctionnement du système d'authentification

Des technologies de chiffrement plus ou moins rudimentaires sont utilisées pour garantir la sécurité et la confidentialité du mécanisme d'authentification. Reportez-vous au chapitre 4 pour une description détaillées des mécanismes de chiffrement adaptés à l'authentification.

Challenge-response

Le protocole d'authentification utilise en général un mécanisme d'interrogation dit de *challenge-response* (question-réponse en français), pour valider l'identité d'un utilisateur. La réponse à une interrogation donnée n'est connue que par le système d'authentification et par l'utilisateur. Ce *challenge-response* peut être une simple demande de mot de passe ou une demande plus évoluée n'impliquant pas le transfert d'information sensible sur un réseau hostile.

L'authentification repose toujours sur un secret (mot de passe, clé privée, etc.). La détention et l'utilisation de ce secret sont les phases critiques du système. Il est souvent nécessaire de protéger la base de données contenant les informations d'authentification en la chiffrant. De même, le protocole d'authentification utilisé ainsi que les protocoles adjacents, en général des protocoles de communication réseau, doivent se prémunir contre la possibilité d'être détournés (lutte conte l'écoute frauduleuse et le re-jeu).

Linux et l'authentification

Linux utilise un système de gestion des comptes à la couleur Unix. Les utilisateurs sont identifiés par leur nom de compte et par leur appartenance à au moins un groupe. Sur un système autonome, ces informations sont contenues dans quatre fichiers au format texte.

Le fichier /etc/group

Le fichier `group` contient l'ensemble des groupes définis sur le système. Une ligne correspond à un groupe. Elle est elle-même composée de quatre champs séparés par le caractère « : » :

- 1er champ : nom du groupe ;
- 2e champ : lettre « x » ;
- 3e champ : identificateur numérique unique du groupe sur le système (GID ou Group Identifier en anglais) ;
- 4e champ : liste des utilisateurs du groupe séparés des virgules « , ».

```
root:x:0:root
bin:x:1:root,bin,daemon
whouse:x:1000:bill
```

Le fichier /etc/passwd

Le fichier `passwd` contient les informations des comptes utilisateur. Là encore, une ligne composée de sept champs définit un compte :

- 1er champ : nom du compte ;
- 2e champ : « x » si le mot de passe associé est contenu dans le fichier `/etc/shadow`, « * » si le compte est désactivé ;
- 3e champ : identificateur numérique utilisateur (UID ou User Identifier) ;
- 4e champ : identificateur de groupe primaire (GID ou Group Identifier) ;

- 5e champ : description du compte ;
- 6e champ : répertoire personnel (« home directory » en anglais) ;
- 7e champ : interpréteur de commandes (« shell » en anglais).

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
bill:x:1000:1000:Bill:/home/bill:/bin/bash
```

Le fichier /etc/shadow

Ce fichier contient l'information la plus sensible du compte utilisateur : le mot de passe permettant son authentification. Comme pour le contenu du fichier `passwd`, chaque ligne correspond à un compte utilisateur. Elle est composée de 9 champs séparés par le caractère « : ». À chaque compte défini dans le fichier `passwd` correspond une entrée dans le fichier `shadow` :

- 1er champ : nom du compte ;
- 2e champ : mot de passe chiffré ;
- 3e champ : date du dernier changement de mot de passe en nombre de jours depuis le 1er janvier 1970 ;
- 4e champ : nombre de jours avant que le mot de passe ne puisse être changé ;
- 5e champ : nombre de jours après lesquels le mot de passe doit être changé ;
- 6e champ : nombre de jours pour la notification d'expiration du mot de passe ;
- 7e champ : nombre de jours avant de désactiver le compte après expiration du mot de passe ;
- 8e champ : date à laquelle le compte expire en nombre de jours depuis le 1er janvier 1970 ;
- 9e champ : champ réservé pour une utilisation future.

```
root:$1$D9gthGPx$v7PXCcRfYrdpghPMLNaze0:12487:0:99999:7:::
bin:*:12487:0:99999:7:::
bill:$1$D9RTYgpx$V8pxcCrFyRDPGHpmlMofe1:12487:0:99999:7:::
```

Le format du mot de passe est variable suivant l'algorithme utilisé pour le chiffrer. Sur la plupart des systèmes Unix, c'est l'algorithme irréversible Data Encryption Standard (DES) qui est utilisé. Sur les systèmes Linux récents, un algorithme de type MD5, plus robuste, est en général proposé.

Le fichier /etc/gshadow

Un quatrième fichier, `/etc/gshadow` est également présent sur la plupart des distributions Linux. Il est pour le fichier `group` ce que `shadow` est à `passwd`. En général peu ou pas utilisé, sa présence passe inaperçue.

Ces quatre fichiers sont le cœur du système d'authentification du système autonome Linux. Il est primordial que les permissions d'accès du système de fichiers ne puissent être modifiées par un autre compte que celui de l'administrateur du système.

Format du mot de passe chiffré

Dans l'exemple suivant, le mot de passe qui apparaît dans une entrée tirée du fichier `shadow` a été chiffré avec un algorithme de type Data Encryption Standard (DES) de la fonction `crypt` :

```
bill:BDtcCCEJ.jUAM:9479:0:10000::::
```

La chaîne de caractères « `BD` » est la semence du générateur pseudo-aléatoire qui a fourni le mot de passe crypté « `tcCCEJ.jUAM` ».

Dans cet autre exemple, le mot de passe a été chiffré avec un algorithme de type MD5 :

```
bill:$1$T19w51$H8amev19P3UQrFfHowctM/:9479:0:10000::::
```

Avec un chiffrement MD5, le format du 2e champ d'une ligne du fichier `shadow` est caractéristique. La chaîne de caractères « `$1$` » indique le chiffrement MD5. La semence, chaîne de caractères « `T19w51` », suit cette séquence jusqu'au prochain caractère « `$` ». Vient enfin le mot de passe chiffré : « `H8amev19P3UQrFfHowctM/` ».

RÉFÉRENCES DES et MD5

- http://en.wikipedia.org/wiki/Data_Encryption_Standard
- http://en.wikipedia.org/wiki/MD5

B.A.-BA Sécurité et robustesse du système d'authentification Linux

La sécurité du mot de passe stocké dans le fichier `/etc/shadow` repose sur deux éléments primordiaux :

- D'abord, la confidentialité du contenu du fichier `/etc/shadow`. Celui-ci ne doit être lisible que par l'administrateur de la machine.
- Ensuite, l'algorithme utilisé pour chiffrer le mot de passe doit donner un résultat unique et être irréversible. Il ne doit pas être possible de retrouver le mot de passe en appliquant un algorithme inverse sur la chaîne cryptée et il ne doit pas non plus être possible d'obtenir une chaîne cryptée identique avec un mot de passe différent.

Gestion des comptes utilisateur

Sur le système Linux RedHat, un jeu de commandes en ligne facilite la manipulation du contenu des fichiers `passwd`, `group`, `shadow` et `gshadow`. Ceci évite l'édition manuelle, parfois dangereuse sur des fichiers aussi critiques. En voici quelques exemples :

- Pour créer le groupe d'utilisateurs « whouse » dont l'identificateur numérique est 1000 :

```
# groupadd -g 1000 whouse
```

- Pour créer le compte utilisateur « bill » dont l'identificateur est 1000 et dont le groupe d'appartenance est « whouse » :

```
# useradd -u 1000 -g whouse bill
```

- Changement du mot de passe du compte « bill » et modification de l'entrée correspondante dans le fichier `shadow`.

```
# passwd bill
```

Les commandes `userdel`, `usermod`, `groupdel` et `groupmod` respectivement suppriment ou modifient les attributs d'un compte utilisateur ou d'un groupe dans les fichiers de définitions des comptes.

Principe de l'authentification par mot de passe

Le principe de l'authentification par mot de passe est simple (figure 10-2). L'utilisateur donne son mot de passe, qui est transmis au système d'authentification. Ce mot de passe est alors chiffré en utilisant le même algorithme et la même semence que pour celui qui est contenu dans le fichier `/etc/shadow`. Si la chaîne obtenue est identique, l'utilisateur est alors authentifié. Sur la figure 10-2 par exemple, la chaîne cryptée correspond au mot de passe « banana ».

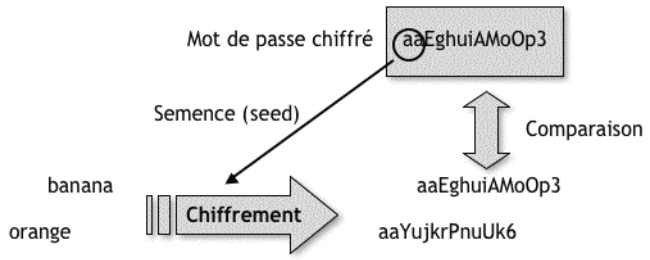


Figure 10–2
Principe de l'authentification

Linux et PAM

RÉFÉRENCE PAM Linux

- http://www.kernel.org/pub/linux/libs/pam/

PAM, qui est l'acronyme de Pluggable Authentication Modules, propose un mode de fonctionnement très modulaire rendant transparent à l'application l'utilisation de tel ou tel système d'authentification. Par le biais d'un fichier de configuration qui sert d'aiguillage et de bibliothèques dynamiques proposant une interface standard et normalisée, l'application sera rendue complètement indépendante de la manière dont seront gérées les informations relatives aux comptes utilisateur. Des modules PAM existent pour NIS, LDAP et Kerberos, rendant ainsi les trois systèmes d'authentification interchangeables sans que cela nécessite une quelconque reconfiguration ou encore recompilation des services. La figure 10-3 présente le mode de fonctionnement modulaire de PAM.

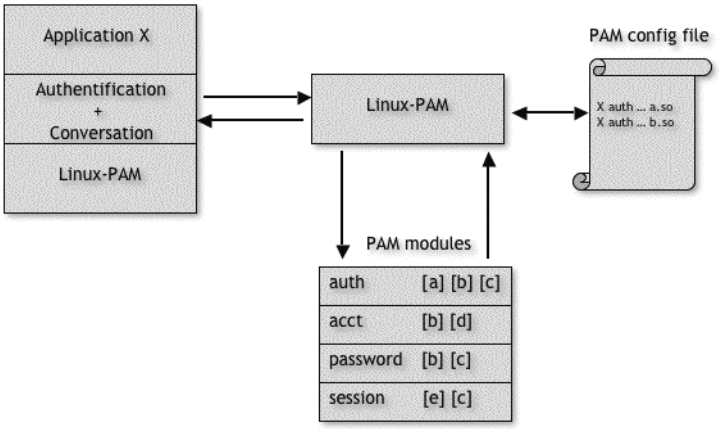


Figure 10–3
Fonctionnement PAM

Sur la distribution Linux utilisée par Tamalo, les fichiers de configuration PAM sont contenus dans le répertoire `/etc/pam.d`. Pour chaque méthode (`auth`, `account`, `password` et `session`), un module est associé avec un niveau de dépendance comme cela est présenté dans l'exemple qui suit. Le répertoire `/lib/security` contient les modules PAM proposés par les systèmes d'authentification.

Contenu du fichier de règles PAM /etc/pam.d/system-auth

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is
run.
auth      required      /lib/security/$ISA/pam_env.so
auth      sufficient    /lib/security/$ISA/pam_unix.so likeauth
auth      required      /lib/security/$ISA/pam_deny.so
```

```
account   required      /lib/security/$ISA/pam_unix.so

password  required      /lib/security/$ISA/pam_cracklib.so
retry=3
password  sufficient  /lib/security/$ISA/pam_unix.so md5 shadow
password  required      /lib/security/$ISA/pam_deny.so

session   required      /lib/security/$ISA/pam_limits.so
session   required      /lib/security/$ISA/pam_unix.so
```

Linux et Name Service Switch

Name Service Switch, aussi connu sous l'acronyme NSS, est un ensemble de bibliothèques servant au système Linux pour interroger les différents services de nommage, quelle qu'en soit leur source. Par exemple, la correspondance nom de machine et adresse IP peut être trouvée dans le fichier texte `/etc/hosts` sur le système Linux, mais également en procédant à une interrogation DNS. De même, la définition des comptes utilisateur peut être locale dans les fichiers du répertoire `/etc`, mais il est possible de trouver la correspondance en interrogeant un service de gestion centralisé tel que NIS. Le fichier `/etc/nsswitch.conf`, dont un extrait est présenté ci-après, contient la description des sources à consulter pour obtenir l'information sur un item particulier.

Extrait du fichier /etc/nsswitch.conf

```
passwd:     files nis
shadow:     files nis
group:      files nis
hosts:      files dns
```

Network Information Service - NIS

NIS est l'acronyme de Network Information Service. Développé par la société SUN Microsystems, NIS est une extension du mécanisme d'authentification d'un système d'exploitation Unix. NIS permet notamment la redistribution des informations contenues dans les fichiers `passwd`, `group` et `shadow` d'une machine de référence (le serveur maître NIS), à l'ensemble des machines du réseau. Le protocole d'authentification par comparaison de mot de passe chiffré reste identique à celui présenté précédemment. Seule la source des informations utilisées pour la base de données des utilisateurs change. Elle n'est plus locale, mais consultée à travers le réseau. L'administration des comptes à partir d'un point unique est donc possible. C'est dans ce contexte que NIS a retenu l'attention des administrateurs de Tamalo.

RÉFÉRENCE NIS pour Linux

Le groupe Linux NIS, comme son nom l'indique, est à l'origine du portage du système NIS sur la plateforme Linux. Il s'agit aujourd'hui d'un logiciel mature utilisable en milieu système hétérogène et permettant l'interaction avec d'autres composants NIS de sources différentes.

- http://www.linux-nis.org

HISTOIRE Network Information Service

Initialement connu sous le nom de Yellow Page puis renommé Network Information Service à cause de la marque déposée de la société United Kingdom of British Telecom pour son annuaire téléphonique papier, NIS a été développé par la société SUN Microsystems. Prévu pour redistribuer aux machines d'une même communauté (le domaine NIS) des informations de configuration diverses contenues par exemple dans les fichiers `passwd`, `group`, `services`, `networks` ou encore `hosts` du répertoire `/etc`, NIS a été rendu très populaire par sa simplicité d'utilisation. Ce système permet aujourd'hui à de nombreux sites d'administrer l'ensemble des comptes utilisateur à partir d'un point unique.

- http://fr.wikipedia.org/wiki/NIS

Fonctionnement

Dans le modèle de fonctionnement proposé par NIS, un serveur maître détient la copie originelle de la base de données de référence des comptes utilisateur. Cette base de données, modifiable, est tout simplement contenue dans les fichiers locaux `passwd`, `group` et `shadow` comme pour un poste de travail ou un serveur autonome. Une conversion du contenu de ces fichiers dans un format « NIS » (les cartes ou les « maps » NIS en anglais), en général un format permettant l'indexation, est ensuite propagée à des serveurs esclaves de la même communauté NIS (le domaine NIS) et redistribuée sur le réseau aux machines clientes de ce domaine nécessitant ces informations d'authentification (voir figure 10-4).

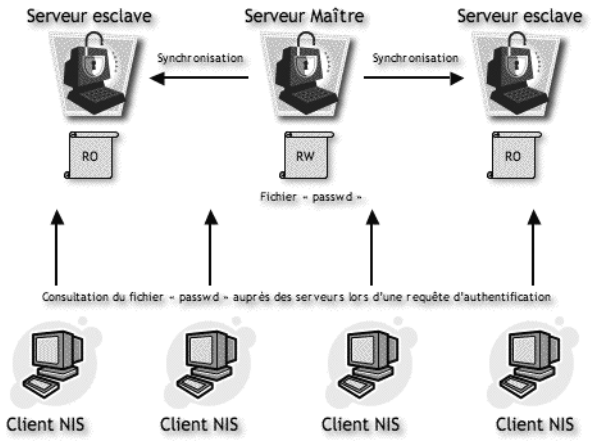


Figure 10–4
Architecture d'un domaine NIS

Les serveurs esclaves ne disposent que d'une version en lecture seule (RO ou *read-only* en anglais), de la base de données. Les modifications ne sont réalisées que sur le serveur maître avec les commandes standards présentées à la section « Gestion des comptes utilisateur ». Après modification du contenu

des fichiers de référence `passwd`, `shadow` et `group`, il convient de propager la nouvelle version aux serveurs esclaves comme cela est présenté dans l'exemple qui suit.

- Ajout de l'utilisateur « bill » avec les commandes standards du système :

```
# useradd -u 1000 -g whouse bill
# passwd bill
Changing password for user bill
New password:
Retype new password:
passwd:
all authentication tokens updated successfully
```

- Conversion du contenu des fichiers plats en « maps » NIS :

```
# cd /var/yp
# make
```

Affichage des informations contenues dans les maps NIS

Même s'il y a conversion de format de la base de données des comptes utilisateur avec NIS, la présentation des données change peu pour l'utilisateur final.

La commande `ypcat` affiche le contenu d'une carte NIS toute entière. Dans l'exemple suivant, il serait probablement difficile de faire la différence avec la visualisation du contenu du fichier `/etc/group` :

```
# ypcat group
whouse:x:1000:
hexagone:x:1001:
```

La commande `ypmatch` recherche dans une « map » NIS indexée. Les « maps » `passwd` et `group`, sont indexées par noms et par identificateurs numériques :

```
# ypmatch bill passwd.byname
bill:aaTHidUEku7:1000:1000:Bill:/nfs/home/bill:/bin/bash
```

Toutes les informations relatives à un domaine NIS sont contenues dans le sous-répertoire portant le nom du domaine NIS du répertoire `/var/yp`.

Le domaine NIS

Le domaine NIS correspond à une communauté de machines partageant les mêmes informations propagées par un serveur maître et ses esclaves. N'importe quelle machine peut rejoindre le domaine si elle dispose du nom de ce dernier.

Répartition de charge et disponibilité

En plus d'offrir une gestion de la base de données utilisateur à partir d'un point central, il a été prévu dans NIS des mécanismes de répartition de charge et de redondance. En effet, l'utilisation d'un service réseau pour un système aussi critique que celui de l'authentification nécessite une disponibi-

lité maximale. Cette architecture à plusieurs serveurs permet à la fois de répartir la charge induite par les requêtes des machines clientes, mais également de remédier à la disparition d'une machine qui officierait comme serveur NIS.

Rejoindre un domaine NIS et trouver son serveur

Pour rejoindre un domaine NIS, son nom doit être connu par le client (démon `ypbind`) qui s'exécute sur le système d'exploitation. Deux méthodes servent ensuite à sélectionner le serveur à interroger :

- La première consiste en un mécanisme d'interrogation par *broadcast*. Une requête est envoyée sur le réseau pour déterminer la liste des serveurs disponibles pour le domaine choisi. Le premier serveur qui répond à cette requête, en général le moins chargé, est alors choisi comme référent par le client NIS. Cette opération est appelée association ou *binding* en anglais. Si d'aventure ce serveur venait à disparaître, il suffirait de réémettre une requête de recherche pour trouver un autre serveur.
- À ce mode de configuration très souple existe une alternative plus statique. En effet, une liste de serveurs connus peut être décrite dans le fichier de configuration `/etc/yp.conf` du client NIS.

Adresse de diffusion ou broadcast

L'adresse de diffusion ou adresse de *broadcast*, est l'adresse IP d'un réseau qui désigne l'ensemble des machines de ce réseau. Une requête envoyée à l'adresse de diffusion sera reçue par l'ensemble des machines du réseau. Il y aura donc autant de réponses à la requête que de machines connectées au réseau (sur lesquelles le service auquel s'adresse la requête est actif). Cette adresse a la particularité d'avoir tous les bits de la partie adresse des machines dans le réseau positionnés à 1 (valeur décimale 255 si le codage est effectué sur 1 octet).

Limites du système NIS

Simplicité ne rime pas toujours avec sécurité. NIS, conçu à une époque où la sécurité n'était pas la préoccupation principale des développeurs, souffre de quelques défauts majeurs dont il faut être conscient avant de procéder à son déploiement. Suivant le niveau de sécurité souhaité, NIS sera ou non le bon candidat.

Tout d'abord, les informations sont redistribuées sur le réseau sans chiffrement. Un pirate capturant le trafic réseau pourra récupérer les chaînes cryptées des mots de passe et utiliser un outil de « crackage » de type « brute force » pour découvrir quelques mots de passe dont la robustesse ne serait pas assez éprouvée.

Pire encore, n'importe quelle machine configurée en client NIS peut rejoindre un domaine si elle en connaît le nom et donc récupérer les informations distribuées par un serveur. La seule restriction possible, trop souvent oubliée, consiste à définir sur le serveur dans les fichiers de configuration `/etc/ypserv.conf` et `/var/yp/securenets`, l'intervalle d'adressage réseau à partir duquel les requêtes sont possibles. Cette configuration sera décrite à l'annexe B.

Le manque de mécanisme d'authentification dans la recherche d'un serveur permet à une personne malintentionnée de configurer une machine en ser-

Attaque « brute force »

L'attaque par force brute est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Il s'agit de tester, une à une, toutes les combinaisons possibles. Cette méthode de recherche exhaustive ne marche que dans les cas où le mot de passe cherché est constitué de peu de caractères.
Pour contrer cette attaque, il suffit simplement de choisir des mots de passe d'une longueur conséquente avec une complexité importante ou des clés suffisamment grandes.
Cette méthode est souvent combinée avec l'attaque par dictionnaire pour obtenir de meilleurs résultats. De la puissance actuelle des ordinateurs dépend la durée nécessaire pour trouver un mot de passe, et donc la capacité des pirates à compromettre les comptes utilisateur.

veur NIS avec un nom de domaine existant. Un client configuré en mode *broadcast* pour s'associer à un serveur, peut en pratique décider de choisir la machine du pirate. De fausses informations d'authentification permettraient alors à celui qui les diffuserait de se connecter sur la machine client avec le compte et le mot de passe de son choix.

Lightweight Directory Access Protocol - LDAP

LDAP est l'abréviation de l'anglais Lightweight Directory Access Protocol. LDAP est aussi et surtout un protocole standard de communication pour les services d'annuaires. Ces derniers sont en général utilisés pour stocker des informations sur les comptes utilisateur, mais peuvent également servir pour stocker tout autre type d'information. Les annuaires LDAP sont très appréciés car simples à mettre en œuvre et optimisés et performants pour des requêtes de lecture.

À l'instar de NIS, LDAP est utilisé dans le système d'authentification pour permettre une gestion centralisée et un partage des comptes utilisateur. Sur le système Linux, OpenLDAP, implémentation libre d'un serveur LDAP, est utilisé pour répondre aux besoins de partage des informations de comptes entre machines. Son schéma de base de données peut être étendu pour ajouter des types d'informations supplémentaires à la gestion traditionnelle des comptes Unix, comme des adresses électroniques, des numéros de téléphone, etc.

OUTIL OpenLDAP

OpenLDAP est une implémentation Open Source d'un serveur d'annuaire offrant un protocole d'accès conforme à la normalisation LDAP. Très populaire dans l'environnement Linux, il est un des logiciels les plus utilisés dans la gestion et le partage des informations ayant trait aux comptes utilisateur.

- http://www.openldap.org

RÉFÉRENCE LDAP et normalisation

Le protocole LDAP et ses extensions ont fait l'objet d'une description détaillée dans de nombreuses RFC (Requests for Comment), disponibles pour la plupart sur le site :

- http://www.mozilla.org/directory/standards.html

Fonctionnement

Dans le modèle proposé par OpenLDAP, un serveur principal détient la base de données originelle des comptes utilisateur. La description de ces comptes dans cette base de données satisfait aux RFC 2307 (*An approach for Using LDAP as a Network Information Service*) et 2829 (*Authentication methods for LDAP*). La description des comptes utilisateur et des groupes répond alors à la norme Posix.

Plus simplement, les informations contenues dans les fichiers `passwd`, `group` et `shadow` retrouvent leur équivalent dans la structure de base de données LDAP, ce qu'on appelle communément le schéma. Des outils sont proposés pour convertir les fichiers plats dans leur équivalant LDAP.

L'interrogation d'un serveur LDAP et la modification des données qu'il gère sont possibles depuis n'importe quelle machine du réseau ayant satisfait aux règles d'accès imposées par le serveur.

En utilisant une authentification classique par mot de passe, une fois encore, l'utilisation de LDAP, tout comme NIS, est transparente aux applications du

système. Le module PAM associé interrogera le serveur adéquat lors des phases d'authentification afin de récupérer les caractéristiques de chacun des comptes utilisateur.

Contrairement à NIS, la base de données LDAP peut être dissociée du contenu des fichiers plats du répertoire `/etc`, et donc être gérée de manière indépendante.

LDAP et la sécurité

LDAP est un protocole réseau disposant d'une implémentation sur la couche sécurisée SSL (voir chapitre 4 « Chiffrement des communications avec SSH et SSL »). Les possibilités de chiffrer systématiquement tout le trafic réseau entre serveur et client, ainsi que d'authentifier le serveur, limitent la sensibilité à l'écoute frauduleuse et le risque d'utiliser les services d'un serveur pirate.

Des mécanismes évolués de gestion de listes de contrôle d'accès (Access Control List ou ACL, Access Control Item ou ACI), permettent de gérer finement les opérations de lecture, d'écriture et de modification des informations de la base LDAP. L'accès est notamment contrôlé au niveau de la machine ou de l'ensemble de machines, mais également au niveau de l'utilisateur. Dans le cas de NIS par exemple, tous les utilisateurs ont accès en lecture à l'ensemble de la base de données, y compris les mots de passe cryptés. Avec LDAP, cette possibilité est mieux contrôlée. Des listes de contrôle d'accès correctement positionnées ne permettent aux utilisateurs de voir que ce qu'ils ont le droit de voir.

Exemple d'ACL OpenLDAP

```
access to attr=userPassword
        by self write
        by * auth
access to * by * read
```

Répartition de charge et disponibilité

Par des mécanismes de réplication sur différents serveurs, les informations d'authentification, c'est-à-dire la base de données des utilisateurs, peuvent exister en plusieurs exemplaires. Plusieurs serveurs sont ainsi disponibles pour offrir une répartition de la charge et un service tolérant à la panne.

Limitation du système LDAP

Dans le modèle proposé par LDAP, les serveurs réplicas ne détiennent que des copies de la base de données originale. Même si ces copies sont ne pas en

lecture seule, les modifications ne doivent être faites que sur le serveur principal et propagées ensuite aux réplicas.

LDAP n'est certainement pas aussi intégré que NIS au système Linux ou Unix plus généralement. Même si des outils de conversion du contenu des fichiers plats `passwd` et `group` accompagnent la distribution OpenLDAP, ils ne permettent que des opérations massives utilisées lors d'une migration d'un système d'authentification traditionnel à LDAP et non pour une gestion quotidienne des comptes utilisateur. Il est alors nécessaire de développer des outils propres pour la gestion des comptes et manipuler le contenu de la base de données LDAP.

Même s'il est non sécurisé, le mécanisme de découverte des serveurs NIS par *broadcast* est très séduisant. Aucune configuration spécifique n'est nécessaire sur les clients. Avec LDAP, une configuration statique est obligatoire. Le nom des serveurs doit apparaître dans le fichier de configuration `/etc/ldap.conf`.

Kerberos

Kerberos est un protocole d'authentification réseau basé sur l'utilisation de clés partagées. Il est construit sur le postulat que le réseau est non sécurisé et dangereux, notamment parce qu'il est le lieu d'écoute frauduleuse et d'usurpation d'identité. Dans le modèle Kerberos, le contrôleur de domaine, KDC ou Kerberos Domain Controler en anglais, est l'autorité de certification ultime. Ce serveur maître d'authentification peut être secondé par des serveurs disposant d'une copie de la base de données d'authentification ; on parle alors d'esclaves.

Plusieurs versions du protocole Kerberos ont vu le jour. La version 5 est très certainement la plus mature. Deux implémentations de cette version diffusées dans le monde Open Source sont celle du Massachusetts Institute of Technology et celle du projet Heimdal de l'Institut Royal de Technologie de Stockholm.

OUTILS Distributions Kerberos 5

Deux distributions de Kerberos 5 proposées respectivement par le Massachusetts Institute of Technology (MIT) et l'Institut Royal de Technologie de Stockholm (KTH) sont disponibles en Open Source. Elles sont compatibles avec la majorité des systèmes Unix et même, pour celle du MIT, avec le système Microsoft Windows.

- http://web.mit.edu/kerberos/www/
- http://www.pdc.kth.se/heimdal/

Fonctionnement

Le serveur Kerberos gère une base de données de couples composés d'un « principal » et d'une clé privée associée. Ce « principal » représente en général un compte utilisateur ou un serveur applicatif. La base de données Kerberos n'est donc pas une base de données de comptes utilisateur telle que nous la connaissons avec NIS ou encore avec LDAP. Les informations qu'elle contient sont réduites et propres au fonctionnement du système d'authentification. Il n'y a pas de relation entre système d'authentification et

TERMINOLOGIE « realm » et « principal »

Le royaume, traduit de l'anglais « realm », est plus communément appelé un domaine. Dans la terminologie Kerberos, il représente la communauté des systèmes utilisant le même serveur ou le même ensemble de serveurs Kerberos partageant la même base de données d'authentification. Ce « realm » est en général un nom de domaine DNS en majuscules, par exemple TAMALO.COM.
Le sujet, « principal » en anglais, correspond à une entité unique que le serveur Kerberos peut authentifier. Plus concrètement, un « principal » peut être un compte utilisateur ou encore un serveur applicatif.

identification locale sur un système d'exploitation donné. Il est possible d'être authentifié auprès du serveur Kerberos sans pour autant disposer d'une définition locale sur le système, un UID par exemple. Les deux fonctionnalités authentification et identification sont indépendantes.

Kerberos et la sécurité

Kerberos utilise des techniques de cryptographie à tous les niveaux. Les communications par exemple sont chiffrées et évitent ainsi l'écoute frauduleuse. De plus, à aucun moment dans les phases d'authentification le mot de passe n'est transmis sur le réseau. Il n'est en effet pas nécessaire pour authentifier un utilisateur que celui ci fasse parvenir son mot de passe jusqu'au système d'authentification comme cela est le cas pour des systèmes plus classiques. Dans la mesure où le mot de passe est partagé par les deux parties, il existe des mécanismes permettant de vérifier la détention du secret par l'autre partie sans que ce secret soit transmis. La figure 10-5 présente une manière assez simpliste, les mécanismes Kerberos étant nettement plus évolués, d'authentifier un utilisateur sans lui demander son mot de passe.

Le serveur d'authentification envoie au client un challenge, dans cet exemple un nombre tiré aléatoirement, que le client incrémentera après l'avoir déchiffré. Le résultat sera ré-encrypté avec la clé partagée avant d'être retourné au serveur d'authentification. Une fois décrypté par le serveur d'authentification, le résultat sera comparé et le client authentifié ou non.

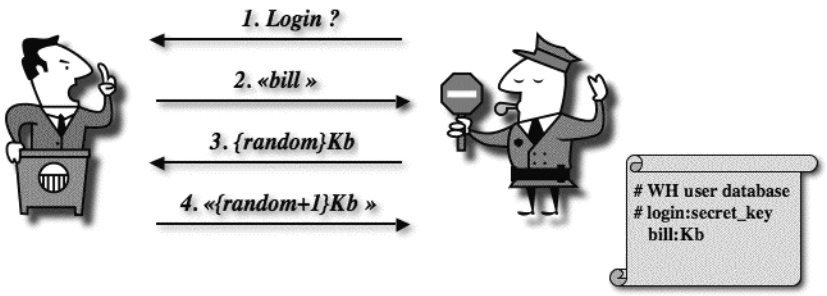


Figure 10–5
Authentification sans transmission du mot de passe

Authentification unique ou « Single Sign On »

Avec Kerberos, après une authentification réussie, l'utilisateur se voit délivrer une accréditation valable pendant une durée limitée : le ticket. Ce ticket peut être utilisé pendant toute sa durée de validité pour authentifier le porteur auprès d'un service « kerberisé », c'est-à-dire utilisant les mécanismes d'authentification Kerberos. Cette technique permet d'utiliser un mot de

passe (le secret partagé) une première fois, puis d'utiliser le ticket Kerberos obtenu pour les phases d'authentification suivantes auprès d'autres services. Un mécanisme utilisant une seule authentification par mot de passe pour l'ensemble des services du système d'information, est appelée en anglais « Single Sign On ».

La commande `klist` permet d'afficher le ticket, cette accréditation temporaire Kerberos, obtenu lors de la phase d'authentification par mot de passe :

```
# klist
Ticket cache: FILE:/tmp/krb5cc-0
Default principal: admin/admin@TAMALO.COM

Valid starting     Expires            Service principal
05/09/05 10:44:31  05/09/05 20:44:31  krbtgt/TAMALO.COM@TAMALO.COM
        renew until 05/20/05 10:44:31
```

Limites du système Kerberos

La mise en œuvre de Kerberos se heurte au besoin de devoir déployer un système de gestion de comptes utilisateur externe pour l'identification de ces derniers sur le système Linux. En effet, Kerberos se limite à fournir des protocoles et des mécanismes d'authentification très sécurisés. Il n'assure pas la gestion des informations à caractère « administratif » du point de vue système, permettant par exemple la correspondance UID – nom de compte et GID – nom de groupe. Des logiciels tels que NIS et LDAP doivent être utilisés en complément pour distribuer ces informations à caractère administratif. Il est alors nécessaire de mettre en œuvre des mécanismes garantissant l'homogénéité des différentes bases de données, celle de Kerberos et celle de l'annuaire annexe, afin d'éviter toute incohérence. Cette gestion du système d'authentification peut être considérée comme lourde et n'est pas envisageable dans toutes les situations.

B.A.-BA GINA

GINA (Graphical Identification aNd Authentication) est la bibliothèque d'authentification fournie par Microsoft pour ses systèmes NT/2000/XP. Sous sa forme native, GINA n'a aucune possibilité d'être interfacée avec d'autres moyens d'authentification que ceux prévus par l'éditeur de logiciels. Néanmoins, il a été prévu que cette bibliothèque dynamique puisse être remplacée.

RÉFÉRENCE pGINA

Le projet pGINA propose une bibliothèque de remplacement à GINA chargeant dynamiquement les modules d'authentification les plus courants comme LDAP et NIS.

- http://www.pgina.org

Interopérabilité

Même si l'environnement des machines de Tamalo est homogène, les choix des administrateurs système restent dictés par une possible ouverture. Les outils choisis pour le système d'authentification doivent pouvoir être utilisés avec un autre système d'exploitation.

Pour NIS, LDAP et Kerberos, l'intégration est réalisée sans difficulté sur la quasi majorité des systèmes Unix. Dans le cas des systèmes Microsoft Windows, l'intégration peut être plus ardue.

Une distribution de Kerberos est proposée par le MIT pour les différentes versions du système Windows. Pour NIS et LDAP, un développement Open Source du nom de pGINA, offre la possibilité d'interfacer ces systèmes dans le schéma d'authentification d'un système Windows. pGINA est une alternative à l'utilisation de GINA, la bibliothèque d'authentification utilisée sur les systèmes Windows. Son fonctionnement est semblable à celui de PAM sous Linux. pGINA utilise le module que l'administrateur système aura configuré pour l'authentification de l'utilisateur sur le poste de travail.

En résumé…

Il n'existe pas un service d'authentification et de gestion des comptes utilisateur universel, mais un grand nombre de solutions plus ou moins adaptées à des besoins différents. Il convient, comme pour tout déploiement de service réseau, de bien maîtriser l'outil et d'en évaluer les limites.

Si NIS semble être le système le plus homogène et le plus intégré au système Unix, il pèche néanmoins par d'importantes lacunes liées à la sécurité dont l'étendue peut être heureusement limitée au domaine du réseau local.

Kerberos est sans contexte le plus sûr des trois systèmes étudiés, mais cette sécurité en fait un outil parfois lourd à gérer. La nécessité de lui adjoindre une base de données annexe pour l'identification des comptes utilisateur sur le système d'exploitation représente un niveau de complexité supplémentaire.

LDAP semble être un bon candidat à plus de sécurité et de souplesse dans la gestion d'une base unique des comptes utilisateur. Son défaut principal réside certainement dans la nécessité de développer des outils de gestion de comptes utilisateur propres à sa structure de données.

Infrastructure à gestion de clés : création de l'autorité de certification de Tamalo.com

A

Référence Infrastructure à gestion de clés du CNRS

http://www.urec.fr
rubrique *IGC Infrastructure à gestion de clés*.

Qu'est-ce qu'une bi-clé ?

Les techniques de chiffrement asymétrique décrites au chapitre 4 reposent sur l'utilisation d'algorithmes mettant en jeu des couples de clés composés d'une clé publique et de sa clé privée associée. Ce couple est communément appelé « bi-clé ».

Les techniques de chiffrement décrites au chapitre 4 permettent d'assurer la confidentialité des échanges électroniques. L'utilisation de bi-clés (clé publique/clé privée) pose le problème de la gestion des clés publiques. En effet, comment être sûr que la clé publique présentée correspond bien à la personne ou au service voulu ?

Cette problématique est celle que résout une Infrastructure à gestion de clés (IGC), Public Key Infrastructure (PKI) en anglais. Les aspects organisationnels de la création d'une IGC peuvent être très complexes, souvent largement plus que les aspects techniques traités dans ce qui suit. En effet, les protocoles utilisés pour autoriser la délivrance d'un certificat électronique à une personne ou à un service vont influer directement sur la confiance que l'on pourra accorder aux certificats ainsi délivrés.

Cette annexe donne la démarche pratique pour la création d'une Autorité de certification (AC), Certification Authority (CA) en anglais, pour la société Tamalo.com. Cette AC permettra de signer des certificats (voir au chapitre 4) pour des utilisateurs ou des services. On donnera un exemple pratique d'utilisation des certificats ainsi concus, pour authentifier un utilisateur voulant signer un message électronique, ainsi que pour authentifier un serveur HTTPS.

RÉFÉRENCE

Des outils pour déployer une IGC

- http://www.openssl.org
  le site officiel de OpenSSL

Si vous êtes – un peu - allergique à la ligne de commande, ces références sont pour vous :

- http://www.cryptonit.org/
  pour les logiciels clients.
- http://idx-pki.idealx.org
  pour le côté serveur.

OpenSSL et les IGC

OpenSSL est une implémentation libre et gratuite de SSL (Secure Sockets Layer) et TLS (Transport Layer Security). Présente sur la majorité des systèmes Linux et portée sur un grand nombre d'autres systèmes, cette boîte à outils propose un jeu de commandes complet pour la création et la manipulation des certificats utilisés dans le cadre d'une IGC. Dans cette annexe, vous trouverez les commandes OpenSSL nécessaires pour créer une IGC et délivrer des certificats personnels ou de services.

Création des certificats X.509

Certificat électronique

Un certificat contient, tel un passeport ou une carte d'identité, un ensemble d'informations administratives sur son propriétaire (nom, prénom, adresse électronique, etc.), sur sa validité, et sur l'organisme d'émission. Il contient également la clé publique de l'utilisateur et un sceau (la signature électronique de l'autorité de certification) nécessaire à la vérification de son authenticité. Il permet de garantir l'identité du possesseur de la clé privée associée. Voir au chapitre 4.

La commande `openssl` implémente un ensemble de sous-commandes permettant d'effectuer toutes les opérations nécessaires à la fabrication et à la gestion des certificats électroniques.

Pour obtenir de l'aide sur une sous-commande d'`openssl` (ici `genrsa`), invoquez : `openssl genrsa -`.

La création des certificats X.509 comprend les étapes suivantes :

- création d'une bi-clé de l'autorité de certification ;
- signature du certificat de l'AC ;
- création des bi-clés des utilisateurs et des services ;
- création des demandes de certificats des utilisateurs et des services ;
- signature des certificats des utilisateurs et des services par l'AC.

Bi-clés RSA

Les bi-clés RSA pour une autorité de certification, un utilisateur ou une machine, sont créés avec la commande `openssl genrsa`, comme indiqué sur la figure A-1.

Par défaut, les clés produites ont une longueur de 512 bits. Elles sont encodées avec l'algorithme 3DES. Ces paramètres sont aisément modifiables. Les trois bi-clés calculées sont enregistrées dans les fichiers `CA-Tamalo.key` pour l'autorité de certification, `Boutherin.key` et `Delaunay.key` respectivement pour les utilisateurs Boutherin et Delaunay.

Il est possible de visualiser les couples de clés en format texte avec la commande :

```
# openssl rsa –text –in <nom de fichier>
```

```
root@rh71: /root/PKI
File  Edit  Settings  Help
[root@rh71 PKI]# openssl genrsa -des3 -out CA-Tamalo.key
warning, not much extra random data, consider using the -rand option
Generating RSA private key, 512 bit long modulus
...................................++++++++++++
...................................++++++++++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
[root@rh71 PKI]# openssl genrsa -des3 -out Boutherin.key
warning, not much extra random data, consider using the -rand option
Generating RSA private key, 512 bit long modulus
...............++++++++++++
.....................++++++++++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
[root@rh71 PKI]# openssl genrsa -des3 -out Delaunay.key
warning, not much extra random data, consider using the -rand option
Generating RSA private key, 512 bit long modulus
...........++++++++++++
.....................++++++++++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
[root@rh71 PKI]# ls
Boutherin.key  CA-Tamalo.key  Delaunay.key
[root@rh71 PKI]#
```

Figure A–1 Création des bi-clés RSA pour l'autorité de certification et pour les utilisateurs Boutherin et Delaunay

Certificat X.509 auto-signé de l'autorité de certification

L'autorité de certification racine (*root CA*) est l'instance suprême de la chaîne de certification. Elle signe elle-même son certificat.

La commande utilisée pour calculer un certificat auto-signé est la suivante :

```
# openssl req -new -key CA-Tamalo.key -x509 -days <nombre de
jours> -out CA-Tamalo.crt
```

L'option `days` permet de fixer la durée de vie du certificat. Pour l'AC, cette durée doit être importante, car les certificats signés expireront au maximum en même temps que l'AC. Dans l'exemple de la figure A-2, la durée de vie du certificat de l'AC est de 10 ans.

Pour créer le certificat électronique, il faut fournir les informations sur l'identité de l'AC, voir figure A-2. Ces informations correspondent au champ DN (Distinguished Name) d'un enregistrement LDAP. Elles seront stockées dans le certificat. La valeur par défaut de ces champs peut être modifiée dans le fichier `/usr/share/ssl/openssl.cnf`

Ce certificat est auto-signé : le nom de l'AC signataire du certificat (voir le champ `Issuer` dans la figure A-3) est le même que le nom du titulaire du certificat (voir le champ `Subject` dans la figure A-3).

```
root@rh71: /root/PKI
File  Edit  Settings  Help
[root@rh71 PKI]# ls
Boutherin.key  CA-Tamalo.key  Delaunay.key
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]# openssl req -new -key CA-Tamalo.key -x509 -days 3650 -out CA-Tamalo.crt
Using configuration from /usr/share/ssl/openssl.cnf
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]:Isere
Locality Name (eg, city) []:Grenoble
Organization Name (eg, company) [Internet Widgits Pty Ltd]:tamalo.com
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:TAMALO
Email Address []:cert@tamalo.com
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]#
```

Figure A–2
Création du certificat auto-signé de l'autorité de certification

```
root@rh71: /root/PKI
File  Edit  Settings  Help
[root@rh71 PKI]# openssl x509 -in  CA-Tamalo.crt -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=FR, ST=Isere, L=Grenoble, O=tamalo.com, CN=TAMALO/Email=cert@tamalo.com
        Validity
            Not Before: Feb 20 14:43:25 2004 GMT
            Not After : Feb 17 14:43:25 2014 GMT
        Subject: C=FR, ST=Isere, L=Grenoble, O=tamalo.com, CN=TAMALO/Email=cert@tamalo.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:f2:8d:ba:5c:37:68:06:2f:2f:58:59:33:9b:3c:
                    26:90:ec:64:d9:fd:95:fa:4b:ea:44:e4:32:ef:25:
                    f7:88:a6:bb:27:11:c3:90:14:47:a7:e8:4f:06:cf:
                    68:bf:51:80:e9:94:29:96:6c:2b:dd:32:5e:74:f1:
                    91:8d:15:8b:61
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                CA:F7:48:5F:AE:DF:C9:3B:E3:AE:A7:28:FF:A5:D7:2E:A5:71:D6:FE
            X509v3 Authority Key Identifier:
                keyid:CA:F7:48:5F:AE:DF:C9:3B:E3:AE:A7:28:FF:A5:D7:2E:A5:71:D6:FE
                DirName:/C=FR/ST=Isere/L=Grenoble/O=tamalo.com/CN=TAMALO/Email=cert@tamalo.com
                serial:00

            X509v3 Basic Constraints:
                CA:TRUE
```

Figure A–3
Visualisation du certificat de l'autorité de certification

Demande de certificats utilisateur

La commande pour créer une demande de certificat est la suivante :

```
# openssl req -new -key <fichier_contenant_la_bi-clé> -out
<demande_de_certificat>
```

La demande de certificat permet de générer un fichier qui sera transmis à la CA correspondante pour être signée par celle-ci.

Pour la demande de certificat utilisateur, il faut fournir les informations concernant le DN de l'utilisateur. Le champ `Email Address` (voir figure A-4) est très important. En effet si l'utilisateur veut se servir de son certificat pour signer des messages électroniques, ce champ doit correspondre exactement à son adresse de messagerie.

```
root@rh71: /root/PKI
File  Edit  Settings  Help
[root@rh71 PKI]# openssl req -new -key Boutherin.key -out Boutherin.csr
Using configuration from /usr/share/ssl/openssl.cnf
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]:Isere
Locality Name (eg, city) []:Grenoble
Organization Name (eg, company) [Internet Widgits Pty Ltd]:tamalo.com
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:Bernard Boutherin
Email Address []:boutherin@tamalo.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
[root@rh71 PKI]#
[root@rh71 PKI]#
```

Figure A–4
Demande de certificat utilisateur

Signature des certificats par l'autorité de certification

L'AC signe les certificats demandés avec la commande `openssl` donnée ci-dessous. C'est elle qui fixe la durée de validité des certificats signés, 365 jours dans l'exemple de la figure A-5.

```
# openssl x509 -CA <certificat_de_l'AC> -CAkey <bi-
cles_de_l'AC> -req -in <demande_de_certificat> -out
<certificat_signe> -days 365 -Cacreateserial
```

Comme le montre la figure A-5, pour signer la demande de certificat, il faut bien sûr fournir la *passphrase* permettant d'accéder à la clé privée de l'AC !

```
root@rh71: /root/PKI
File  Edit  Settings  Help
[root@rh71 PKI]# openssl x509 -CA CA-Tamalo.crt -CAkey CA-Tamalo.key -req -in Boutherin.csr -out Bouth
erin.crt -days 365 -CAcreateserial
Signature ok
subject=/C=FR/ST=Isere/L=Grenoble/O=tamalo.com/CN=Bernard Boutherin/Email=boutherin@tamalo.com
Getting CA Private Key
Enter PEM pass phrase:
[root@rh71 PKI]# openssl x509 -text -in Boutherin.crt
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 2 (0x2)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=FR, ST=Isere, L=Grenoble, O=tamalo.com, CN=TAMALO/Email=cert@tamalo.com
        Validity
            Not Before: Feb 20 14:58:07 2004 GMT
            Not After : Feb 19 14:58:07 2005 GMT
        Subject: C=FR, ST=Isere, L=Grenoble, O=tamalo.com, CN=Bernard Boutherin/Email=boutherin@tamalo
.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (512 bit)
                Modulus (512 bit):
                    00:ce:83:bf:c8:c7:09:8e:cf:2c:59:ad:ae:e8:99:
                    d3:33:d3:b5:c9:99:54:d5:ef:f6:ff:de:14:70:01:
```

Figure A–5
Signature d'un certificat par l'Autorité de certification

Création d'un fichier contenant la clé privée et le certificat au format PKCS12

Ce fichier est destiné au transport de la clé privée et du certificat, ainsi qu'à leur importation dans un navigateur Web (figure A-6). Il est au format PKCS12. Le fichier résultant d'une transformation au format PKCS12 est chiffré, donc protégé des regards indiscrets.

```
# openssl pkcs12 –export –in <certificat> –inkey <bi-cles> –out
<fichier_PKCS12>
```

```
root@rh71: /root/PKI
File  Edit  Settings  Help
R7ek3Q/cYBOCT/CKvudDpU76yOymoTDtaZ12EJPkcyejSIlIDxr5ivmxvZIRJGJw
Q2P/n2DBvlDkVCE=
-----END CERTIFICATE-----
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]#
[root@rh71 PKI]# openssl pkcs12 -export -in Boutherin.crt -inkey Boutherin.key -out Boutherin.p12
Enter PEM pass phrase:
Enter Export Password:
Verifying password - Enter Export Password:
[root@rh71 PKI]#
```

Figure A–6
Création d'un fichier PKCS12

Mise en œuvre d'un serveur Web sécurisé HTTPS

Création du certificat du serveur www.tamalo.com

La création d'un certificat pour le serveur www.tamalo.com s'effectue de la même façon que pour un utilisateur : calcul d'une bi-clé pour le serveur, puis demande de certificat et signature de la demande par l'AC. Comme on le voit sur la figure A-7, le champ important est ici le CN (Common Name) du serveur, qui doit correspondre exactement au nom enregistré dans le DNS pour ce serveur.

```
root@rh71: /root/PKI
File  Edit  Settings  Help
[root@rh71 PKI]# openssl genrsa -des3 -out www.tamalo.com.key
warning, not much extra random data, consider using the -rand option
Generating RSA private key, 512 bit long modulus
...................++++++++++++
............++++++++++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
[root@rh71 PKI]# openssl req -new -key www.tamalo.com.key -out www.tamalo.com.csr
Using configuration from /usr/share/ssl/openssl.cnf
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]:Isere
Locality Name (eg, city) []:Grenoble
Organization Name (eg, company) [Internet Widgits Pty Ltd]:tamalo.com
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:www.tamalo.com
Email Address []:webmaster@tamalo.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
[root@rh71 PKI]# openssl x509 -CA CA-Tamalo.crt -CAkey CA-Tamalo.key -req -in www.tamalo.com.csr -out
www.tamalo.com.crt -days 365 -CAcreateserial
Signature ok
subject=/C=FR/ST=Isere/L=Grenoble/O=tamalo.com/CN=www.tamalo.com/Email=webmaster@tamalo.com
Getting CA Private Key
Enter PEM pass phrase:
[root@rh71 PKI]#
```

Figure A–7
Création d'un certificat serveur

Pour utiliser le certificat serveur ainsi établi, le serveur Apache a besoin de sa clé privée (fichier `www.tamalo.com.key`), de son certificat (fichier `www.tamalo.com.crt`) et du certificat de l'AC signataire (fichier `CA-Tamalo.crt`) permettant de vérifier la chaîne de certification. Il suffit de placer ces fichiers sur le serveur dans le sous-répertoire correspondant de `/etc/httpd/conf/` et de mettre à jour le fichier de configuration d'Apache `/etc/httpd/conf/httpd.conf` comme suit :

```
SSLCertificateFile /etc/httpd/conf/ssl.crt/www.tamalo.com.crt
SSLCertificateKeyFile /etc/httpd/conf/ssl.key/
www.tamalo.com.key
SSLCertificateChainFile /etc/httpd/conf/ssl.crt/CA-Tamalo.crt
```

Attention, le fichier `www.tamalo.com.key` contient la clé privée de la machine. Cette clé ne doit pas être protégée par un mot de passe si on souhaite que le serveur httpd puisse redémarrer sans intervention en cas de *reboot*. Pour déprotéger cette clé, utilisez la commande :

```
openssl rsa -in www.tamalo.com.key -out www.tamalo.com.key
```

Installation de la chaîne de certification sur le client

Les navigateurs des postes clients qui devront accéder au site https://www.tamalo.com devront faire confiance à l'autorité de certification de Tamalo.com. Dans les navigateurs à notre disposition, il existe déjà un certain nombre d'autorités de certification (commerciales) qui sont pré-enregistrées. Comme le certificat de l'AC de Tamalo.com n'est pas signé par une de ces autorités, il est nécessaire de l'ajouter dans les navigateurs.

Pour cela, nous allons créer un script CGI qui permettra de charger le certificat de Tamalo.com dans un navigateur à partir du site Web http://www.tamalo.com.

Certains navigateurs ne reconnaissent pas le type MIME `application/x-x509-cert-ca` utilisé pour les certificats électroniques. Il faut donc définir ce type dans notre script.

Dans le répertoire `cgi-bin` de www.tamalo.com, on crée le fichier suivant :

```
/var/www/cgi-bin/loadca
#!/bin/bash
## Chargement du certificat de l'AC racine
##
echo -e "Content-type: application/x-x509-ca-cert\n"
cat /root/PKI/CA-Tamalo.crt
```

Comme le montre la figure A-8, il suffit alors d'ouvrir la page http://www.tamalo.com/cgi-bin/loadca pour charger le certificat de l'AC dans un navigateur.

Il est possible de visualiser le certificat de l'AC tamalo.com comme le montre la figure A-9.

Cette opération doit être réitérée depuis les clients web qui doivent faire confiance à l'autorité de certification de tamalo.com.

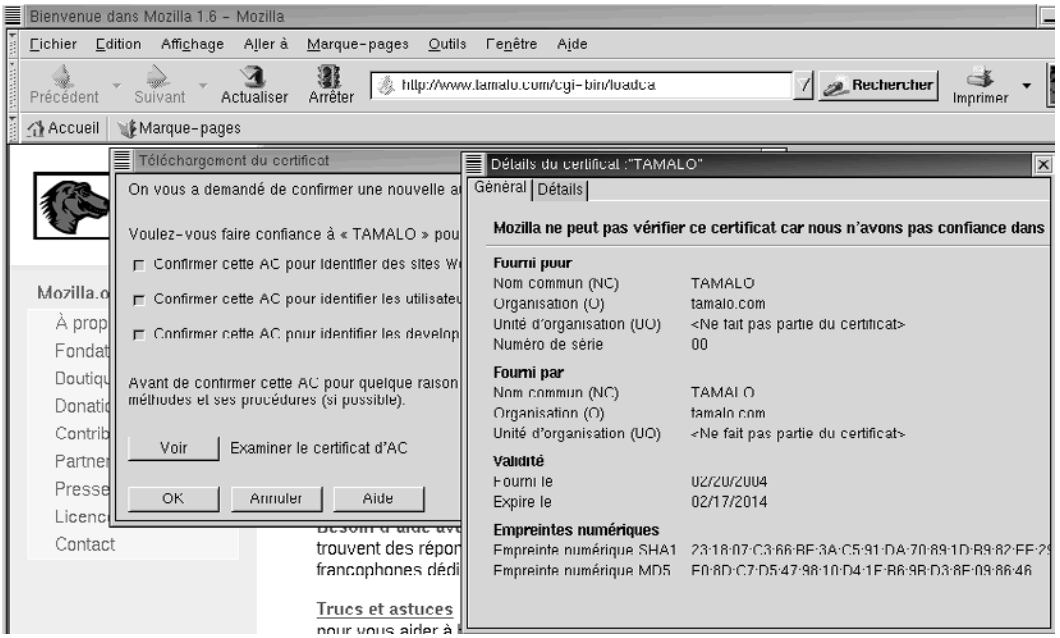


Figure A–8 Chargement du certificat de l'AC dans un navigateur

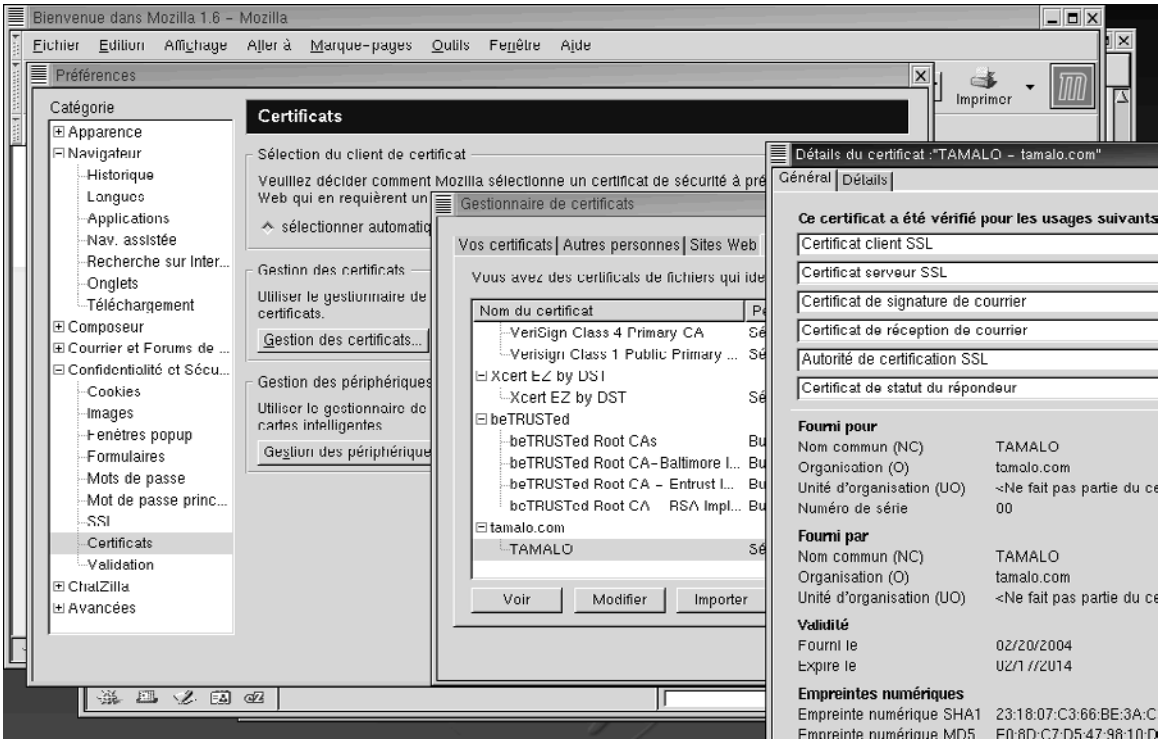


Figure A–9 Visualisation des certificats des CA de confiance avec Mozilla

Installation d'un certificat personnel dans le navigateur

Le certificat et la clé privée de l'utilisateur peuvent-être importés dans le navigateur à partir du format PKCS12.

La figure A-10 montre l'importation d'un certificat personnel dans Mozilla.

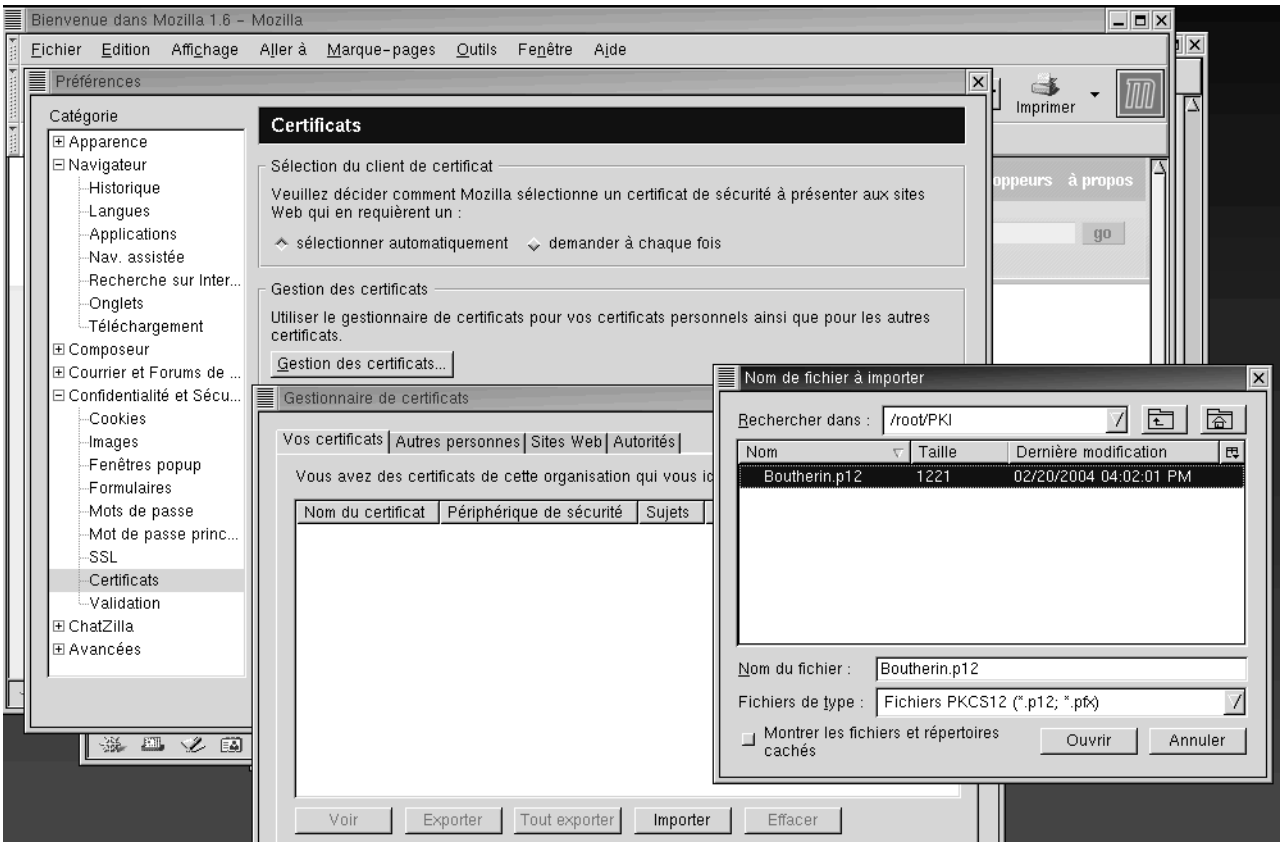


Figure A–10 Importation d'un certificat personnel dans Mozilla

Après avoir importé un certificat personnel, il est possible de le vérifier. Cette vérification s'effectue sur la chaîne complète de certification, jusqu'à trouver un certificat auto-signé d'une CA acceptée par le navigateur, voir figure A-11.

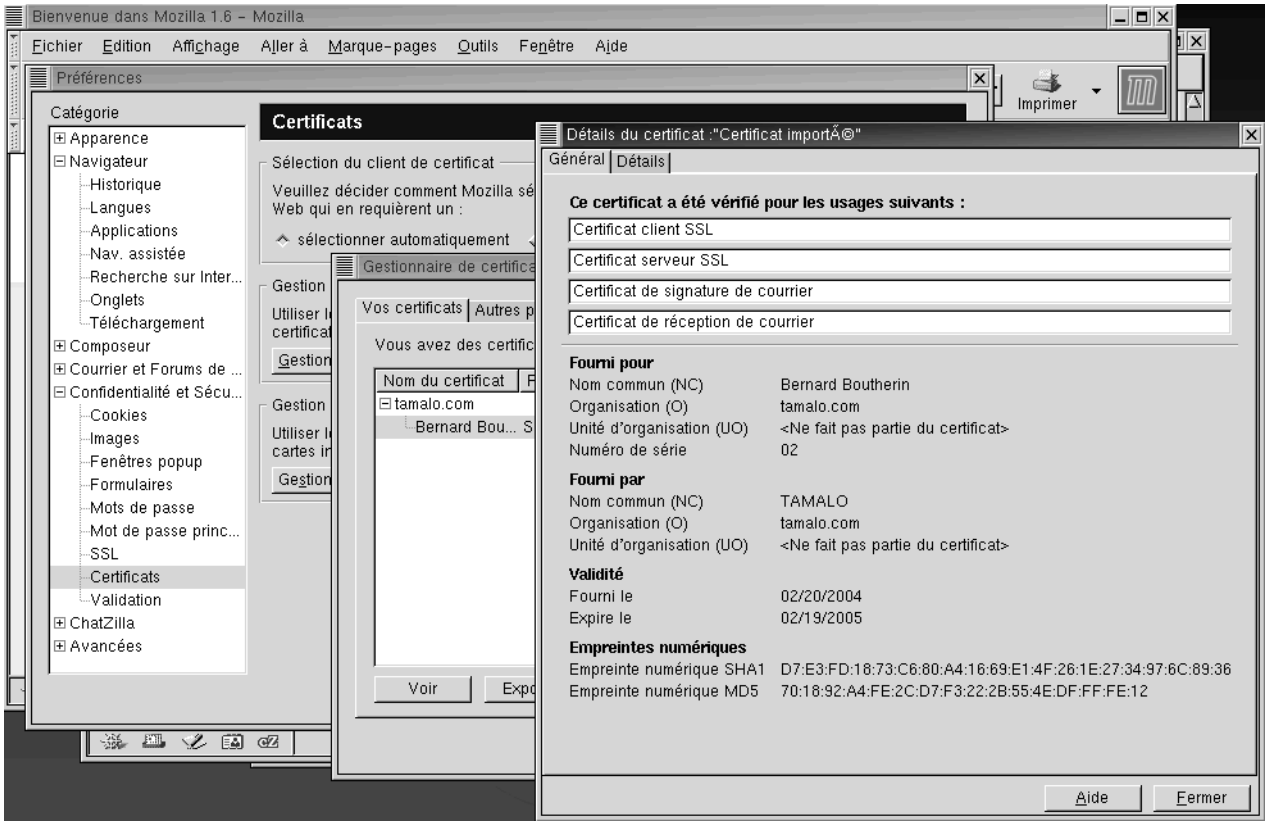


Figure A–11 Vérification de la chaîne de certification avec Mozilla

Utilisation des certificats pour signer et/ou chiffrer les courriers électroniques

Une fois installés votre certificat personnel et celui de l'autorité de certification dans votre navigateur, il est possible d'utiliser ce certificat pour signer des messages électroniques.

Il faut auparavant configurer le client de messagerie pour lui indiquer le certificat à utiliser pour signer ou chiffrer des messages (voir figure A-12).

Il est alors possible d'envoyer un message électronique signé comme indiqué sur la figure A-13.

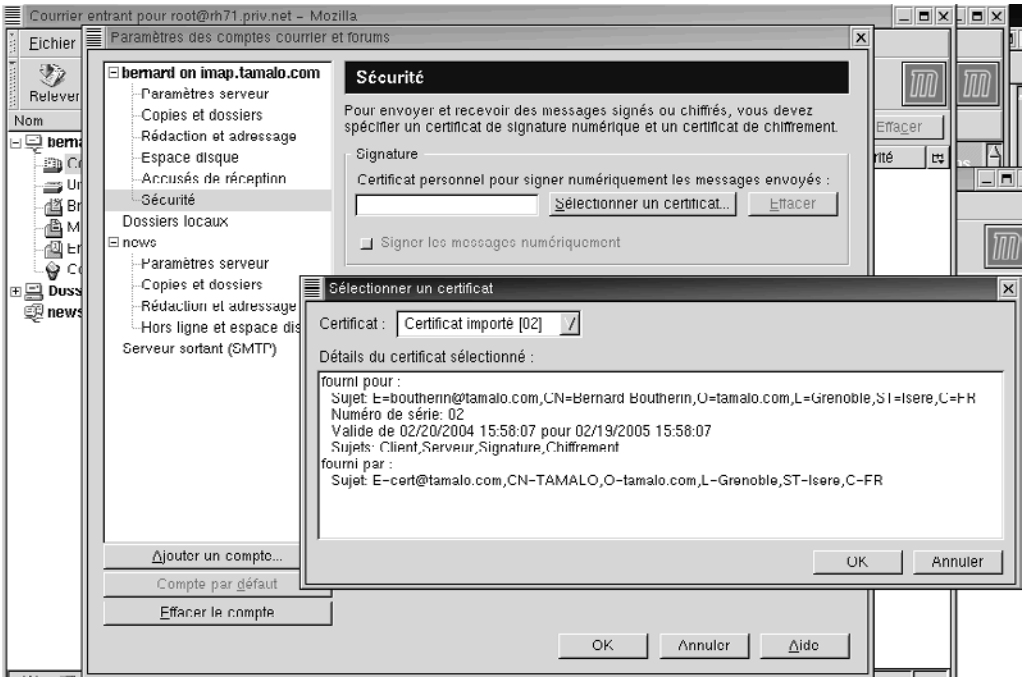


Figure A–12
Sélection d'un certificat pour signer ou chiffrer des messages avec Mozilla

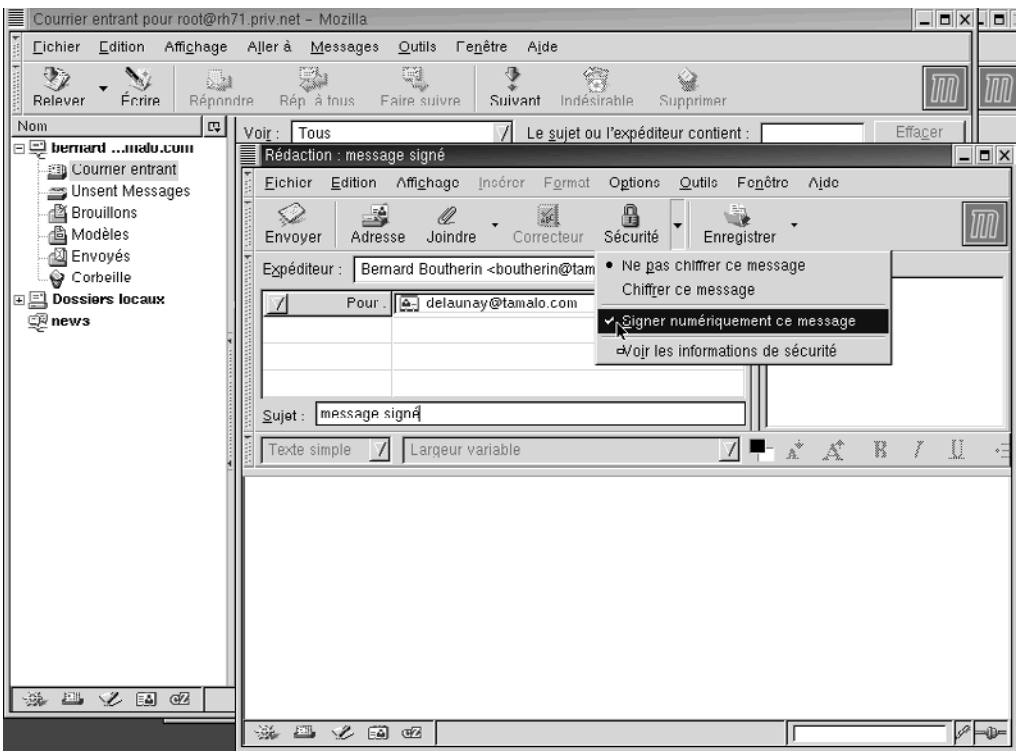


Figure A–13
Envoi d'un message signé avec Mozilla

En conclusion

La certification et les IGC répondent à un besoin grandissant de confidentialité et de non-répudiation. Ils nous permettent aujourd'hui de signer et de chiffrer les courriers électroniques, d'effectuer des achats en ligne, de déclarer nos revenus et bientôt de voter électroniquement. Les certificats pourraient devenir dans le futur le moyen unique d'authentification et d'accès aux services électroniques.

Authentification, mise en œuvre de NIS, LDAP et Kerberos

B

L'administration de quelques dizaines de postes de travail dans un environnement distribué ne peut pas être réalisée de la même manière que celle d'un poste autonome. La mise en place d'outils de gestion centralisée des comptes utilisateur est nécessaire. Nous avons présenté au chapitre 10 les trois principaux logiciels pouvant répondre à la problématique rencontrée par la société Tamalo pour la gestion de sa base de données de comptes et pour son système d'authentification réseau. Voici maintenant la mise en œuvre de NIS, LDAP et Kerberos.

Mise en œuvre de NIS

Développé par la société SUN Microsystems, NIS (Network Information Service) est une extension réseau du service d'authentification d'un système d'exploitation Unix. NIS permet en particulier de redistribuer la base de données des comptes utilisateur contenue généralement dans le fichier `/etc/passwd` d'une machine, à un ensemble de machines connectées au réseau. Cette application est disponible pour une grande majorité de systèmes Unix propriétaires ou Open Source.

Installation du système NIS

La procédure qui suit décrit l'installation d'un serveur maître NIS sur une machine dotée du système d'exploitation Linux RedHat 9 de l'entreprise Tamalo. Cette description inclut également la configuration d'un système client NIS ainsi qu'un exemple de génération de la base de données des comptes.

L'installation et la configuration d'un serveur esclave ne seront pas traitées ici, mais un tel serveur serait nécessaire pour assurer disponibilité du service et répartition de charge dans un environnement de production.

Installation des paquetages NIS

Tableau B–1 Paquetages NIS pour Linux Red Hat 9

Paquetage	Description
`ypserv-2.6-2.rpm`	Serveur NIS, programmes et fichiers de configuration associés (`ypserv`, `yppasswdd`…)
`yp-tools-2.7-5.rpm`	Commandes clientes NIS (`ypcat`, `ypwhich`, `ypmatch`…)
`ypbind-1.11-4.rpm`	Client NIS (`ypbind`) et fichier de configuration associé (`/etc/yp.conf`)

Tous les paquetages du tableau B-1 sont installés sur le serveur NIS. Seuls les paquetages `yp-tools` et `ypbind` sont requis pour une machine utilisant le service d'authentification réseau.

```
# rpm –Uvh <nom_du_paquetage>
```

Configuration du serveur maître NIS

Le fichier /etc/ypserv.conf

Le `fichier /etc/ypserv.conf` est fourni par le paquetage `ypserv`. Il contient les options de configuration des modules serveurs. La version par défaut proposée dans le paquetage convient pour l'utilisation faite à Tamalo.

Exemple de fichier /etc/ypserv.conf

```
#  In this file you can set certain options for the NIS server,
#  and you can deny or restrict access to certain maps based
#  on the originating host.
#
#  See ypserv.conf(5) for a description of the syntax.
#
# Some options for ypserv. This things are all not needed, if
# you have a Linux net.
#
# Should we do DNS lookups for hosts not found in the hosts
# table ?
```

```
# This option is ignored in the moment.
dns: no

# How many map file handles should be cached ?
files: 30

# Should we register ypserv with SLP ?
slp: no
# After how many seconds we should re-register ypserv with SLP ?
slp_timeout: 3600

# xfr requests are only allowed from ports < 1024
xfr_check_port: yes

# The following, when uncommented,  will give you shadow like
# passwords.
# Note that it will not work if you have slave NIS servers in your
# network that do not run the same server as you.

# Host                  : Domain  : Map              : Security
#
# *                     : *       : passwd.byname    : port
# *                     : *       : passwd.byuid     : port

# Not everybody should see the shadow passwords, not secure, since
# under MSDOG everbody is root and can access ports < 1024 !!!
*                       : *       : shadow.byname    : port
*                       : *       : passwd.adjunct.byname : port
# If you comment out the next rule, ypserv and rpc.ypxfrd will
# look for YP_SECURE and YP_AUTHDES in the maps. This will make

# the security check a little bit slower, but you only have to
# change the keys on the master server, not the
# configuration files on each NIS server.
# If you have maps with YP_SECURE or YP_AUTHDES, you should
create
# a rule for them above, that's much faster.
# *                        : *       : *                : none
```

Le fichier /var/yp/securenets

Le fichier `/var/yp/securenets` contient les règles d'accès au serveur NIS. Il comporte notamment la liste des réseaux ou des machines pour lesquels l'accès à l'information distribuée par le serveur NIS est autorisé. La procédure de création de ce fichier est la suivante :

```
# cp /usr/share/doc/ypserv-2.6/securenets /var/yp
# chown root:root /var/yp/securenets
# chmod 644 /var/yp/securenets
```

Exemple de fichier /var/yp/securenets

```
# securenets    This file defines the access rights to
#               your NIS server for NIS clients.
#
#               This file contains netmask/network pairs.
#               A clients IP address needs to match with
#               at least one of those.
#
#               One can use the word "host" instead of a
#               netmask of 255.255.255.255.
#
#               Only IP addresses are allowed in this
#               file, not hostnames.
#
# Always allow access for localhost
255.0.0.0       127.0.0.0

# Tamalo's subnets
193.48.97.64
255.255.255.0192.168.154.0
255.255.255.0192.168.155.0
```

Configuration du nom de domaine NIS

La commande `nisdomainname` permet de configurer le nom de domaine NIS manuellement sur une machine appartenant au domaine, qu'elle soit serveur ou non.

```
# nisdomainname tamalonis
```

Pour que le nom de domaine soit configuré à chaque redémarrage du système d'exploitation, sa définition doit apparaître dans le fichier `/etc/sysconfig/network`.

```
NISDOMAIN=tamalonis
```

Lancement du serveur NIS

Le protocole réseau NIS repose sur l'utilisation des RPC (Remote Procedure Call en anglais) et dépend donc de la présence du service de détermination de ports *Portmap*. Les lancements manuels du démon `portmap` et du serveur NIS `ypserv` s'effectuent sous le compte administrateur « root ».

```
# service portmap start

# service ypserv start
```

La commande `chkconfig` rend le lancement de ces services automatique à chaque redémarrage du système :

```
# chkconfig --level 345 portmap on
# chkconfig --level 345 ypserv on
```

L'ensemble des informations du domaine « tamalonis» doit être généré après le lancement du serveur. Le répertoire `/var/yp/tamalonis` est créé lors de cette phase de configuration. Il contient les bases de données du domaine NIS (comptes utilisateur, etc…).

```
# cd /var/yp
# make
```

La configuration du serveur maître NIS est maintenant achevée.

Configuration d'un client NIS

Le fichier de configuration /etc/yp.conf

Le fichier de configuration `/etc/yp.conf` est utilisé par le démon client NIS `ypbind`. Si le nom de domaine NIS est déjà défini dans le fichier `/etc/sysconfig/network`, la directive `broadcast` doit au minimum apparaître dans `/etc/yp.conf`.

Exemple de fichier /etc/yp.conf

```
# /etc/yp.conf - ypbind configuration file
# Valid entries are
#
# domain NISDOMAIN server HOSTNAME
#       Use server HOSTNAME for the domain NISDOMAIN.
#
# domain NISDOMAIN broadcast
#       Use  broadcast  on  the local net for domain NISDOMAIN
#
# domain NISDOMAIN slp
#      Query local SLP server for ypserver supporting NISDOMAIN
#
# ypserver HOSTNAME
#       Use server HOSTNAME for the  local  domain.  The
#       IP-address of server must be listed in /etc/hosts.
#
# broadcast
#       If no server for the default domain is specified or
#       none of them is rechable, try a broadcast call to
#       find a server.
#
broadcast
```

Attention, l'utilisation du *broadcast* ne peut se faire que dans le même sous-réseau. Dans la mesure où trois sous-réseaux sont utilisés au sein de l'entreprise Tamalo, il convient soit de définir la liste des serveurs de manière statique dans le fichier `yp.conf`, soit de configurer des serveurs esclaves dans chacun des sous-réseaux où des clients NIS sont configurés.

Lancement du client NIS

Préalablement au lancement su client NIS `ypbind`, il est nécessaire de s'assurer que le service *Portmap* est démarré et que le domaine a été configuré.

Le lancement manuel du client NIS s'effectue par la commande suivante :

```
# service ypbind start
```

Le lancement automatique du client NIS à chaque redémarrage du système est configuré avec la commande `chkconfig` :

```
# chkconfig –level 345 ypbind on
```

Il est possible d'afficher le serveur NIS référant que le démon `ypbind` a sélectionné, en exécutant la commande `ypwhich` :

```
# ypwhich
nis1.tamalo.com
```

Configuration de l'identification et de l'authentification

Par défaut, le système d'exploitation cherche à faire la correspondance entre identificateurs (UID et GID) et noms associés d'utilisateurs ou de groupes avec les informations locales, c'est-à-dire celles contenues dans les fichiers `passwd` et `group`.

Lors de l'utilisation de NIS, ou de tout autre système centralisé de gestion de comptes, il devient nécessaire de renseigner le système sur la source des informations qu'il devra utiliser. Ceci est réalisé en modifiant le contenu du fichier `/etc/nsswitch.conf` de la manière suivante :

```
passwd:     files nis
shadow:     files nis
group:      files nis
```

Ces trois directives indiquent au système de chercher les informations de comptes d'abord localement dans les fichiers plats, puis en cas d'échec, d'interroger le serveur NIS référent.

Il en est de même pour l'authentification. Même si l'utilisation d'un système d'authentification externe est transparente pour les applicatifs, la configuration du service PAM doit refléter cette particularité.

Dans le fichier `/etc/pam.d/system-auth`, remplacez la ligne suivante :

```
password    sufficient    /lib/security/$ISA/pam_unix.so nullok
use_authtok md5 shadow
```

par :

```
password    sufficient    /lib/security/$ISA/pam_unix.so nullok
use_authtok md5 shadow nis
```

Création de comptes utilisateur

Modification du fichier /var/yp/Makefile

Le fichier `/var/yp/Makefile` décrit les règles de génération des cartes (*maps*) NIS. Il est utilisé par la commande `make` lorsque des modifications ont été faites sur les fichiers plats. Dans le cas présent, seule la génération des *maps* associées aux fichiers `passwd` et `group` est utilisée. Il convient de modifier le fichier `Makefile` afin de remplacer la ligne suivante :

```
all:  passwd group hosts rpc services netid protocols mail \
```

par :

```
all:  passwd group
```

Création d'un groupe et d'un compte utilisateur

L'enchaînement de commandes qui suit définit un groupe puis un compte utilisateur, localement sur le serveur maître NIS. Cette définition est ensuite propagée dans les *maps* NIS afin que toutes les machines du réseau disposent de ces nouvelles informations.

Attention, les outils de génération des *maps* NIS ne propagent que les définitions de groupes et de comptes ayant respectivement des GID et des UID supérieurs à une valeur minimale qui par défaut est 500.

Création du groupe « direction »

```
# groupadd -g 1000 direction
```

Création du compte « bernard »

```
# useradd -u 1000 -g direction bernard
```

Initialisation du mot de passe du compte « bernard »

```
# passwd bernard
```

Génération des *maps* NIS

```
# cd /var/yp
# make
```

Consultation des maps NIS

Les commandes en ligne `ypcat` et `ypmatch` montrent le contenu des cartes NIS. La première en affiche le contenu tout en entier, la seconde effectue une recherche dans une *map* indexée.

Exemple d'utilisation de la commande ypcat

```
# ypcat group
direction:x:1000:
sysadmin:x:1001:
```

Exemple d'utilisation de la commande ypmatch

```
# ypmatch bernard passwd.byname
bernard:aaTHidUEku7:1000:1000:Bernard Boutherin:/nfs/home/
bernard:/bin/bash
```

Mise en œuvre de OpenLDAP

OpenLDAP est un service d'annuaire appliquant le protocole de communication LDAP. Comme pour NIS, il est utilisé dans le système d'authentification pour héberger les informations de comptes utilisateur. OpenLDAP, optimisé pour les accès en lecture, est devenu très populaire dans le monde du logiciel libre.

Introduction

La procédure suivante décrit l'installation d'un serveur OpenLDAP sur une machine dotée du système d'exploitation Linux RedHat 9 de l'entreprise Tamalo. Cette description inclut également la configuration d'un système client capable d'interroger la base de données des comptes utilisateur et de l'utiliser comme système d'authentification.

L'installation et la configuration d'un réplica ne seront pas traitées. Toutefois, un second serveur serait nécessaire pour assurer disponibilité du service et répartition de charge dans un environnement de production.

Installation des paquetages OpenLDAP

Tableau B–2 Paquetages OpenLDAP pour Linux Red Hat 9

Paquetage	Description
`nss_ldap-202-5.rpm`	Modules PAM et NSS intégrant LDAP dans le mécanisme d'authentification du système.
`openldap-2.0.27-8.rpm`	Fichiers de configuration et bibliothèques partagées.
`openldap-servers-2.0.27-8.rpm`	Serveur OpenLDAP, programmes et fichiers de configuration associés.
`openldap-clients-2.0.27-8.rpm`	Commandes clientes LDAP (`ldapadd`, `ldapdelete`, `ldapmodify`, etc.)

Tous les paquetages du tableau B-2 sont installés sur le serveur LDAP. Seuls les paquetages `nss_ldap` et `openldap` sont requis pour une machine qui doit utiliser le service d'authentification LDAP, mais qui ne se comporte pas comme un serveur.

```
# rpm –Uvh <nom_du_paquetage>
```

Redirection des messages de logs

Le serveur OpenLDAP envoie par défaut les traces de son activité au système Syslog avec le label *local4*. Il est important d'avoir accès à ces traces lors de la phase d'installation du logiciel, mais également pendant sa phase de production afin de corriger d'éventuels problèmes. Pour cela, il est nécessaire d'éditer le fichier de configuration du Syslog et de lui ajouter la directive suivante :

```
# Save ldap messages to ldap.log
local4.*                        /var/log/ldap.log
```

Le redémarrage du Syslog est nécessaire pour la prise en compte de cette modification :

```
# touch /var/log/ldap.log
# service syslog restart
```

Configuration du serveur OpenLDAP

Le fichier `/etc/openldap/slapd.conf` contient la configuration du serveur OpenLDAP. Une version de ce fichier est fournie par le paquetage `openldap-servers`. Le contenu du fichier de configuration utilisé dans le cadre du travail d'évaluation mené par Tamalo est présenté ci-après .

```
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include    /etc/openldap/schema/core.schema
include    /etc/openldap/schema/cosine.schema
include    /etc/openldap/schema/inetorgperson.schema
include    /etc/openldap/schema/nis.schema
include    /etc/openldap/schema/redhat/rfc822-MailMember.schema
include    /etc/openldap/schema/redhat/autofs.schema
include    /etc/openldap/schema/redhat/kerberosobject.schema
#
TLSCertificateFile /usr/share/ssl/certs/slapd.pem
TLSCertificateKeyFile /usr/share/ssl/certs/slapd.pem
TLSCACertificateFile /usr/share/ssl/certs/ca-bundle.crt
#

access to attr=userPassword
        by self write
        by anonymous auth
        by * none
#
access to * by self write
        by users read
        by peername.regex="IP=127\.0\.0\.1" read
        by domain=".*\.tamalo\.com" read
        by * none
# ldbm database definitions
database        ldbm
suffix          "dc=tamalo,dc=com"
rootdn          "cn=admin,dc=tamalo,dc=com"
rootpw          {SSHA}SXf8jSt9A8YjdJwCKIzs4aGUEFp4PkYI
# The database directory MUST exist prior to running slapd AND # ldb UEFp4U
rooT* [(re*doT* 5ude*do5ud5ude5ud5ud55ud5ud551A5ud551A51A551ud551A51A551ud5
```

Comment le mot de passe du rootdn a-t-il été généré ?

C'est la commande `slappasswd` qui génère la chaîne cryptée du mot de passe utilisé par l'administrateur de service LDAP. La chaîne de caractères ainsi obtenue est copiée dans le fichier de configuration.

Quelles sont les restrictions d'accès ?

Les permissions décrites dans le fichier de configuration (mot-clé `access`) autorisent les accès en lecture uniquement à partir du serveur lui-même et des machines configurées dans le domaine DNS *tamalo.com*. Des autorisations plus sélectives permettent aux utilisateurs authentifiés de modifier uniquement les attributs de leur compte.

Lancement du serveur OpenLDAP

Le lancement manuel du serveur OpenLDAP s'effectue par la commande suivante :

```
# service ldap start
```

La commande `chkconfig` configure le lancement automatique à chaque redémarrage du système :

```
# chkconfig –level 345 ldap on
```

Après le lancement, il peut être instructif de consulter le contenu du fichier `/var/log/ldap.log`.

Configuration des commandes client

Le fichier `/etc/openldap/ldap.conf` contient les informations nécessaires aux commandes client pour contacter le serveur LDAP. Ces commandes serviront à l'administrateur de la base pour en manipuler le contenu (ajout, suppression, modification d'enregistrements…).

Fichier /etc/openldap/ldap.conf

```
HOST 127.0.0.1
BASE dc=tamalo,dc=com
```

L'adresse du serveur LDAP est celle de *loopback*. Sur une machine qui ne serait pas le serveur, cette adresse devrait être celle de la carte réseau de ce dernier.

Le serveur est lancé, les commandes pour peupler la base de données sont prêtes à être utilisées.

Création du schéma de la base de données

Aucune définition n'a été faite sur la structure du contenu de la base de données LDAP. Cette opération est réalisée avec la commande `ldapadd` et un fichier contenant ces informations au format LDIF (format d'importation).

Le fichier `ldap_base.ldif` est créé par les administrateurs de Tamalo. Il contient les caractéristiques de la base de données LDAP utilisée pour la gestion des comptes.

Fichier ldap_base.ldif

```
dn: dc=tamalo,dc=com
dc: tamalo
objectClass: top
objectClass: domain

dn: ou=People,dc=tamalo,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit
dn: ou=Group,dc=tamalo,dc=com
ou: Group
objectClass: top
objectClass: organizationalUnit
```

Ces définitions sont importées sous le compte administrateur *rootdn* du serveur LDAP avec la commande suivante :

```
# ldapadd -D "cn=admin,dc=tamalo,dc=com" -W -x -f
ldap_base.ldif
Enter LDAP Password:
adding new entry "dc=tamalo,dc=com"
adding new entry "ou=People,dc=tamalo,dc=com"
adding new entry "ou=Group,dc=tamalo,dc=com"
```

La base de données est maintenant prête pour être peuplée avec les définitions des groupes et des comptes utilisateur.

Création d'un groupe

De même que pour la structure de la base de données, la définition d'un groupe se fait en manipulant des données au format LDIF.

Définition du groupe « direction » dans le fichier ldap_group.ldif

```
dn: cn=direction,ou=Group,dc=tamalo,dc=com
objectClass: posixGroup
objectClass: top
cn: direction
userPassword: {crypt}x
gidNumber: 1000
```

Importation de la définition du groupe

```
ldapadd -D "cn=admin,dc=tamalo,dc=com" -W -x -f ldap_group.ldif
Enter LDAP Password:
adding new entry "cn=direction,ou=Group,dc=tamalo,dc=com"
```

Création d'un compte utilisateur

Voici le contenu du fichier `ldap_user.ldif` contenant la définition du compte utilisateur « bernard » :

```
dn: uid=bernard,ou=People,dc=tamalo,dc=com
uid: bernard
cn: bernard
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: {crypt}$1$OybE4vbhDJjeukb$ZxITTVjS4kyTS1
shadowLastChange: 13276
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
homeDirectory: /nfs/home/bernard
```

Importation de la définition de l'utilisateur « bernard »

```
# ldapadd -D "cn=admin,dc=tamalo,dc=com" -W -x -f
ldap_user.ldif
Enter LDAP Password:
adding new entry "uid=bernard,ou=People,dc=tamalo,dc=com"
```

Des outils permettant la migration en masse du contenu des fichiers `/etc/passwd` et `/etc/group` dans la base de données LDAP sont distribués avec le paquetage `openldap-servers`. Ils sont contenus dans le répertoire `/usr/share/openldap/migration`.

Affichage d'un enregistrement

```
# ldapsearch -D "cn=admin,dc=tamalo,dc=com" -W -x "cn=bernard"
Enter LDAP Password:
version: 2

#
# filter: cn=bernard
# requesting: ALL
#

# bernard, People, tamalo, com
dn: uid=bernard,ou=People,dc=tamalo,dc=com
uid: bernard
cn: bernard
objectClass: account
objectClass: posixAccount
objectClass: top
```

```
objectClass: shadowAccount
userPassword:: e2NyeXB0fSQxJE95YkU0dmJoREpqZXVrYiRaeEl
shadowLastChange: 13276
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
homeDirectory: /nfs/home/bernard
```

Configuration de l'identification et de l'authentification

Avant de configurer le système pour qu'il consulte l'annuaire LDAP lors de la phase d'authentification des utilisateurs, il est nécessaire de renseigner dans le fichier de configuration `/etc/ldap.conf` le nom du serveur et de la base à utiliser.

Extrait du fichier /etc/ldap.conf

```
# Your LDAP server. Must be resolvable without using LDAP.
host 127.0.0.1

# The distinguished name of the search base.
base dc=tamalo,dc=com
```

Si la configuration est effectuée sur une autre machine que le serveur d'annuaire, l'adresse IP qui apparaît avec la directive `host` doit être celle de l'interface réseau du serveur.

Par défaut, le système d'exploitation cherche à faire la correspondance entre identificateurs (UID et GID) et noms associés d'utilisateurs ou de groupes avec les informations locales, c'est-à-dire avec celles contenues dans les fichiers `passwd` et `group`.

Lors de l'utilisation de LDAP, il est nécessaire de renseigner le système sur la source des informations qu'il devra utiliser. Ceci est réalisé en modifiant le contenu du fichier `/etc/nsswitch.conf` de la manière suivante :

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap
```

Ces trois directives indiquent au système de chercher les informations de comptes d'abord localement dans les fichiers plats, puis, en cas d'échec, d'interroger le serveur LDAP.

Il en est de même pour l'authentification. Même si l'utilisation d'un système d'authentification externe est transparente pour les applicatifs, la configuration du service PAM doit refléter cette particularité.

Voici le fichier `/etc/pam.d/system-auth` modifié afin de permettre l'interrogation à l'annuaire LDAP :

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.

auth  required      /lib/security/$ISA/pam_env.so
auth  sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth  sufficient    /lib/security/$ISA/pam_ldap.so use_first_pass
auth  required      /lib/security/$ISA/pam_deny.so

account  required      /lib/security/$ISA/pam_unix.so
account  [default=bad success=ok user_unknown=ignore
         ➥ service_err=ignore system_err=ignore]
         ➥ /lib/security/$ISA/pam_ldap.so

password  required      /lib/security/$ISA/pam_cracklib.so retry=3 type=
password  sufficient    /lib/security/$ISA/pam_unix.so nullok use_authtok md5 shadow
password  sufficient    /lib/security/$ISA/pam_ldap.so use_authtok
password  required      /lib/security/$ISA/pam_deny.so

session   required      /lib/security/$ISA/pam_limits.so
session   required      /lib/security/$ISA/pam_unix.so
session   optional      /lib/security/$ISA/pam_ldap.so
```

Ces opérations peuvent être réalisées de manière plus confortable et transparente en utilisant l'interface de configuration `authconfig`.

Mise en œuvre de Kerberos

Kerberos est un protocole d'authentification réseau utilisant des techniques de cryptographie. Il a été développé dans le but de proposer des mécanismes d'authentification forte pour des applications basées sur le modèle client/serveur. Plusieurs implémentations commerciales et Open Source sont disponibles pour la plupart des systèmes d'exploitation utilisés de nos jours (Unix, Microsoft Windows, etc.). Les plus connues dans le monde Open Source sont celle du Massachusetts Institute of Technology (http://web.mit.edu/kerberos/www/) et celle du projet Heimdal de l'Institut Royal de Technologie de Stockholm (http://www.pdc.kth.se/heimdal/).

Installation d'un serveur Kerberos 5

La procédure décrit les étapes nécessaires à l'installation et à la configuration d'un serveur Kerberos 5 issu de la distribution du MIT. Elle présente également la configuration du système d'authentification PAM KRB5 du système Linux Red Hat 9.

MISES À JOUR **Kerberos et la sécurité**

Comme tout logiciel, Kerberos a eu son lot de failles de sécurité. Pour un système d'authentification des utilisateurs, il est plus important encore d'appliquer les mises à jour de sécurité.

- http://web.mit.edu/kerberos/www/advisories/

Cette description n'inclut pas l'installation et la configuration d'un serveur secondaire Kerberos. Celui-ci serait cependant nécessaire pour assurer disponibilité de service et répartition de charge dans un environnement de production.

Installation des paquetages Kerberos 5

Tableau B–3 Paquetages RPM Kerberos 5 MIT pour Linux Red Hat 9

Paquetage	Description
`krb5-workstation-1.2.7-38-3.rpm`	Commandes clientes Kerberos 5 (`kinit`, `klist`, `kdestroy`, `kpasswd`)
`krb5-libs-1.2.7-38.3.rpm`	Bibliothèques partagées Kerberos 5 et fichier de configuration `/etc/krb5.conf`.
`krb5-server-1.2.7-38.3.rpm`	Serveur de Kerberos 5 (`krb5kdc`, `kadmind`)
`pam_krb5-1.60-1.rpm`	Bibliothèque PAM `/lib/security/pam_krb5.so`

Tous les paquetages présentés dans le tableau B-3 sont installés sur le serveur Kerberos. Seuls les paquetages `krb5-workstation` et `krb5-libs` sont installés sur les machines utilisant le système d'authentification Kerberos mais non configurées comme serveur.

```
# rpm -Uvh <nom_du_paquetage>
```

Configuration du serveur Kerberos 5

Le fichier /etc/krb5.conf

Ce fichier est fourni par le paquetage `krb5-libs`. Il est utilisé par les différentes commandes clientes et par le module d'authentification PAM Kerberos 5. Il contient la description du domaine Kerberos, le *realm*, auprès duquel l'authentification doit être réalisée.

Par rapport au fichier de configuration par défaut contenu dans le paquetage, seuls les noms du *realm* et du serveur Kerberos sont remplacés.

Pour cette présentation, le *realm* est « TAMALO.COM » et le seul serveur utilisé se nomme « cerbere.tamalo.com ».

Exemple de fichier /etc/krb5.conf

```
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
```

```
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
default_realm = TAMALO.COM
dns_lookup_realm = false
dns_lookup_kdc = false

[realms]
TAMALO.COM = {
kdc = cerbere.tamalo.com:88
admin_server = cerbere.tamalo.com:749
default_domain = tamalo.com
}

[domain_realm]
.tamalo.com = TAMALO.COM
tamalo.com = TAMALO.COM
[kdc]
profile = /var/kerberos/krb5kdc/kdc.conf
[appdefaults]
pam = {
debug = false
ticket_lifetime = 36000
renew_lifetime = 36000
forwardable = true
krb4_convert = false
}
```

Le fichier /var/kerberos/krb5kdc/kdc.conf

`kdc.conf` est le fichier de configuration du serveur de domaine Kerberos, ou Kerberos Domain Controler (KDC). Il contient les informations nécessaires au fonctionnement du serveur Kerberos, comme le nom de domaine, le chemin d'accès ou encore des directives de logs.

Exemple de fichier /var/kerberos/krb5kdc/kdc.conf

```
[kdcdefaults]
acl_file = /var/kerberos/krb5kdc/kadm5.acl
dict_file = /usr/share/dict/words
admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
v4_mode = nopreauth

[realms]
TAMALO.COM = {
master_key_type = des-cbc-crc
supported_enctypes = des3-hmac-sha1:normal arcfour-hmac:normal \
des-hmac-sha1:normal des-cbc-md5:normal   \
des-cbc-crc:normal des-cbc-crc:v4 des-cbc-crc:afs3
}
```

Le fichier /var/kerberos/krb5kdc/kadm5.acl

Ce fichier contient la définition des droits particuliers de chacun des comptes. Dans la terminologie Kerberos, un compte est un *principal*.

L'exemple suivant donne une configuration minimale de ces droits.

Exemple de fichier /var/kerberos/krb5kdc/kadm5.acl

```
admin/admin@TAMALO.COM *
```

Le principal « admin/admin » possède tous les droits sur la base de données Kerberos. C'est l'administrateur du système d'authentification.

Création de la base de données Kerberos 5

Sous le compte administrateur local « root », exécuter la commande :

```
# kdb5_util create -r TAMALO.COM -s
```

La base de données Kerberos vient d'être créée dans le répertoire `/var/kerberos/krb5kdc`.

Ajout d'un compte administrateur Kerberos

À cette étape, la configuration du serveur Kerberos (ou KDC) est achevée. Toutes les opérations ont été réalisées localement sous le compte administrateur « root » du serveur Kerberos.

Pour administrer à distance la base de données Kerberos, c'est-à-dire à partir de n'importe quelle machine utilisant le système d'authentification, un compte doit être créé (dans cet exemple, il s'agit de celui qui a été défini dans le fichier `kadm5.acl` lors de l'étape précédente). Ceci est réalisé en exécutant la commande `kadmin.local` sous le compte privilégié « root ».

```
# kadmin.local
kadmin.local: addprinc admin/admin@TAMALO.COM
```

Création du fichier /var/kerberos/krb5kdc/kadm5.keytab

Sous le compte privilégié « root », exécuter la commande suivante :

```
# kadmin.local
kadmin.local: ktadd -k /var/kerberos/krb5kdc/kadm5.keytab \
kadmin/admin kadmin/changepw
```

Le fichier `/var/kerberos/krb5kdc/kadm5.keytab` est créé. Il contient la clé d'authentification qui permettra au serveur Kerberos d'authentifier le service de gestion du registre déporté `kadmind`.

Lancement des instances Kerberos sur le serveur KDC

Les deux services `krb5kdc` et `kadmin` sont nécessaires. Le premier est le serveur d'authentification KRB, le second est le service d'administration déportée de la base de données, également appelée registre Kerberos des comptes utilisateur.

Le lancement manuel des services `krb5kdc` et `kadmind` est réalisé sous le compte privilégié « root ».

```
# service krb5kdc start
# service kadmin start
```

Le lancement automatique de Kerberos au redémarrage du système est configuré par :

```
# chkconfig --level 345 krb5kdc on
# chkconfig --level 345 kadmin on
```

Configuration de l'authentification Kerberos

La configuration de l'authentification nécessite celle du fichier `/etc/krb5.conf` comme présenté précédemment, ainsi que la modification du fichier `/etc/pam.d/system-auth`. Ces modifications peuvent être réalisées manuellement ou automatiquement avec l'utilitaire en ligne de commande `authconfig`.

Extrait du fichier de configuration /etc/pam.d/system-auth modifié pour utiliser Kerberos

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth  required      /lib/security/$ISA/pam_env.so
auth  sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth  sufficient    /lib/security/$ISA/pam_krb5.so use_first_pass
auth  required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so
account     [default=bad success=ok user_unknown=ignore service_err=ignore
system_err=ignore] /lib/security/$ISA/pam_krb5.so

password    required      /lib/security/$ISA/pam_cracklib.so retry=3 type=
password    sufficient    /lib/security/$ISA/pam_unix.so nullok use_authtok md5 shadow
password    sufficient    /lib/security/$ISA/pam_krb5.so use_authtok
password    required      /lib/security/$ISA/pam_deny.so

session     required      /lib/security/$ISA/pam_limits.so
session     required      /lib/security/$ISA/pam_unix.so
session     optional      /lib/security/$ISA/pam_krb5.so
```

Création des comptes Kerberos

Les comptes Kerberos sont créés avec l'utilitaire *kadmin*. Ce dernier est lancé avec l'identifiant de l'administrateur Kerberos sous lequel les opérations de création vont être réalisées. Cette commande peut être exécutée sous n'importe quel compte Unix sur n'importe quelle machine permettant l'authentification Kerberos, dans la mesure où l'identifiant de l'administrateur Kerberos est passé en argument.

Ajout du principal « bernard »

```
# kadmin admin/admin@TAMALO.COM
kadmin: addprinc bernard
```

Affichage des attributs du principal « bernard »

```
# kadmin admin/admin@TAMALO.COM
kadmin: getprinc bernard
Principal: bernard@TAMALO.COM
Expiration date: [never]
Last password change: Fri May 1 00:01:55 DFT 2006
Password expiration date: [none]
Maximum ticket life: 0 days 10:00:00
Maximum renewable life: 7 days 00:00:00
Last modified: Fri May 1 00:01:55 DFT 2006 (admin/
admin@TAMALO.COM)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 2
Key: vno 1, Triple DES cbc mode with HMAC/sha1, no salt
Key: vno 1, DES cbc mode with CRC-32, no salt
Attributes:
```

Définition des utilisateurs

Kerberos est un système d'authentification. Il ne propose pas une base de données de gestion des comptes utilisateur comprenant des attributs de type Unix (UID, GID, home, etc.) ou encore à caractère administratif (adresse électronique, numéro de téléphone etc.) comme NIS ou LDAP. La redistribution de ces informations doit être prise en charge par un autre service conjointement à l'utilisation de Kerberos.

Dans le cas de systèmes Unix, une définition locale dans le fichier `/etc/passwd` peut suffire. Dans un environnement comportant plusieurs dizaines de machines, des systèmes comme NIS ou encore LDAP deviennent plus adaptés.

Index

Symboles

/etc/cron.allow 84
/etc/cron.deny 84
/etc/group 212
/etc/gshadow 214
/etc/hosts.allow 83
/etc/hosts.deny 83
/etc/httpd/conf 126
/etc/inittab 77
/etc/krb5.conf 256
/etc/ldap.conf 223, 254
/etc/mail 110
/etc/mail/access 111, 113
/etc/mail/local-host-names 114
/etc/mail/mailertable 113
/etc/mail/sendmail.cf 112
/etc/mail/sendmail.mc 110–113
/etc/nsswitch.conf 217
/etc/pam.d/system-auth 216
/etc/passwd 212
/etc/resolv.conf 104
/etc/shadow 213
/etc/sudoers 81
/etc/sysconfig/named 99
/etc/sysctl.conf 85, 87
/etc/syslog.conf 84
/etc/yp.conf 220, 245
/etc/ypserv.conf 220, 242
/sbin/nologin 83
/usr/share/ssl/certs 124
/var/kerberos/krb5kdc/kadm5.acl 258
/var/kerberos/krb5kdc/kadm5.keytab 258
/var/kerberos/krb5kdc/kdc.conf 257
/var/yp/securenets 220, 243

Numériques

127.0.0.1 111
3DES 49, 53
802.1X 184

A

abuse 41, 42
ack 138, 142
adresse MAC 36
ADSL (Asymmetric Digital Subscriber Line) 3
AES (Advanced Encryption Standard) 49
anaconda-ks.cfg 70
analyse
  à chaud 28
  à froid 29
  disque piraté 29
antispam
  milter-greylist 121
antivirus 116
  base de signatures 118
Apache 12, 126
API 115
APT 74
arpwatch 161
attaque 40
  phishing 128
attaque MiM 128
audit 197, 206
Authentification 209, 210
authentification 18, 46, 47, 52–57, 60–61, 125, 128
  client 52, 53
  fonctionnement 211
  Linux 212
  mot de passe 215
  par certificat 132
  serveur 52, 53
  système 211
AuthorizedKeysFile 61
autorité de certification 52, 53
auto-signé 229
avis de sécurité 42

B

backdoor 28, 30, 33, 38
bibliothèque dynamique 28
bi-clé 227
BIND (Berkeley Internet Name Domain) 12, 16, 97, 98
blacklist 107
blowfish 49
boîte à outils 33
broadcast 158, 220, 223, 246
brute force 220
buffer overflow 26, 145

C

CA (Autorité de Certification) 52
capture 61

CERT (Computer Emergency Resource Team) 42
Renater 2
certificat 52, 53, 126, 227
challenge 46
Challenge-response 211
chiffrement 47, 51, 54, 55, 125
algorithme 47, 50, 54, 57
asymétrique 49, 50
clé publique 49
clé secrète 48
symétrique 48, 53, 54, 57
chiffrer 46, 237
chkconfig 58, 59, 77, 78, 103, 126
chroot 94, 97
ClamAV 117
installation 118
clé
de session 50, 54, 57
privée 49, 54, 62
publique 49, 50, 52–53, 54, 61
secrète 54
cloisonnement 15, 154
CNIL (Commission Nationale de l'Informatique et des Libertés) 192
compression 56
compromission 2, 27, 154
confidentialité 47, 48, 49, 50, 52, 54
configuration minimale 75
connection tracking 138, 148, 171
connexion 139
interactive 59
sécurisée 54, 57
types de 3
Numéris 3
RTC (Réseau téléphonique commuté) 3
consommation 193
copyrights 50
core 40
correctif 26, 73
cracker 25
crond 84
cryptanalystes 46
cryptographie 51
CTIME 30

D

DARPA (Defense Advanced Research Projects Agency) 2, 13
DDOS (Distributed Deny Of Service) 25
débordement de mémoire 26
déchiffrer 46
décrypter 46
défi 46
démon 96
déni de service 88
distribué 25
DES (Data Encryption Standard) 49, 53, 214
détection
intrusion 201
Diffie-Hellman 57
DISPLAY 64
DMZ (DeMilitarized Zone) 154, 155, 158, 186
DNAT 165
DNS (Domain Name System) 12, 95–104
attaque 25
chroot 98
commande host 96, 104
IN 101
MX 101, 107
PTR 102
récursif 96
SOA 101
zone 96, 101
drapeau 141
droit 79, 80
DSA (Digital Signature Algorithm) 50, 53, 57, 61
dsniff 33
dynamic port 139

E

EAP 185
MD5 185
TLS 185
TTLS 185
écoute 15, 46, 55
empreinte 192
established 142, 144, 150
Ethereal 33, 141, 151
écoute d'une session FTP 34
Ethernet 36
exploit 26, 39

F

faille système 26
filtrage 138, 151, 171, 186
avec états 143
FTP 147, 150
limites 142
principes de base 138
règles de base 140
sans état 139
tout autorisé sauf 144
tout interdit sauf 145
find 80, 81
FIRST (Forum of Incident Response and Security Team) 42
flux réseau 18, 155
FreeRADIUS 184
FTP (File Transfer Protocol) 12, 56, 146
mode actif 147
mode passif 149, 150
ftpaccess (/etc/ftpaccess) 151

G

gestion centralisée des logs 191
GINA (Graphical Identification aNd Authentication 225
GNU 5
gourou des systèmes 25
greylist 107

H

H323 144
hacker 25
HostbasedAuthentication 61
HTTP (Hyper Text Transfer Protocol) 16
HTTPS 16, 54, 125, 126

I

IANA (Internet Assigned Numbers Authority) 139
ICMP (Internet Control Message Protocol) 85, 138, 201
Echo request 87

Ignore Bogus Response 87
Redirect 85, 86, 87, 140
IDEA (International Data Encryption Algorithm) 49
identification 209, 210
identité 50
IETF (Internet Engineering Task Force) 51
IGC (Infrastructure à gestion de clés) 53, 54, 227
IgnoreRhosts 61
IMAP (Internet Message Access Protocol) 13, 54, 105, 113, 124
IMAPS 16, 54, 105, 124
indicateurs 204
inetd 83
inetd.conf 30
infrastructure à gestion de clés 227
init.d 30
inittab 30
intégrité 50, 193
Internet 2
intrusion
détection 201
informatique 24
IP (Internet Protocol) 2, 12, 13
spoofing 88, 140
IPchains 171
IPtables 16, 89, 154, 171, 186
chaîne 171
destination NAT 174
filtrage 174
mascarade 174
NAT 174
règle 173
source NAT 174
table 171
ISDN (Integrated Service Digital Network) 3

J

journal 190
journalisation 173

K

KDC (Kerberos Domain Controler) 223
Kerberos 209, 210, 223, 255
configuration 256
fonctionnement 223
limites 225
KickStart 70, 71, 89
kiddies 24
ks.cfg 73

L

LDAP (Lightweight Directory Access Protocol) 209, 210, 221, 248
limites 223
sécurité 222
licence GPL 117
Linux 171
distributions Linux sécurisées 5
histoire 4
noyau 5
système 5
liste
blanche 107
grise 121
noire 107
localhost 111
log 85
Logcheck 190
Logsurfer 190
Logwatch 190
LPRng 75
lprng 40

M

MAC (Media Access Control) 36, 161
man in the middle 128
martien 88
mascarade 168
masque de sous-réseau 158
MD5 214
messagerie
accès nomade 127
architecture 108
électronique 104
MX 104
passerelle 107
profil 108
relais 113
vulnérabilités 105
métrologie 190, 206
MIB (Management Information Base) 193
Milter 115
configuration 116
filtre 116
milter-greylist 121
installation 121
MiM (Man in the Middle) 86
mise à jour 70, 73
modèle OSI 36, 161
modification
bibliothèques dynamiques 28
commandes 28
modules du noyau 28
module du noyau 28
mot de passe 61
BIOS 74
boot 74
démarrage 74
MRTG (Multi Router Traffic Grapher) 193, 206
configuration 196
installation 195, 196
lancement 196
visualisation 196
MTA (Message Transfer Agent) 110
MTIME 30
MX (Mail eXchanger) 96

N

named 97
named.cache 103
named.conf 99
NAT (Network Address Translation) 154, 163, 165, 171, 186
destination 165
dynamique 165, 168, 170
source 165
statique 165, 166, 167
Nessus 16, 197
configuration 198
installation 199
rapport d'audit 200
netcat 148
Netfilter 16, 171, 186
netmask 158

netstat 78
Network Information Service 209
NFS (Network File System) 13
NIDS (Network Intrusion Detection System) 201
NIS (Network Information Service) 209, 210, 217, 218, 241
  client 245
  configuration 242
  domaine 219, 244
  fonctionnement 218
  installation 241
  limites 220
  maps 219
  serveur 244
niveaux d'exécution de Linux 77
NMAP 16, 26, 197
normaliser 51
NSS (Name Service Switch) 209, 217
nullclient 109, 110

O

open relay 106
openbsd 56
OpenLDAP 210, 221, 248
  configuration 249
  fonctionnement 221
  lancement 251
openssh 56, 57, 58, 60
  configuration 58
  openssh-clients 57
  openssh-server 57
  protocole 56
OpenSSL 228
openssl 54
openssl.cnf 229
OSI (Open System Interconnection) 35
outil d'analyse de réseau 34

P

PAM (Pluggable Authentication Modules) 209, 216, 247, 254, 255
paquet forgé 142
pare-feu 16, 138, 151, 154, 157, 173
passphrase 62
PasswdAuthentication 61
PAT (Port Address Translation) 169
patch 26
permission 79
  /tmp 80
PermitEmptypasswords 61
pGINA 225
pirate informatique 24
PKCS12 232
PKI (Public Key Infrastructure) 53, 227
pops 54
port
  dynamic 139
  monitoring 201
  réseau 78
  well known 139
porte dérobée 28, 33, 38
poste de travail 16
PostgreSQL 12
principal 223
processus 76, 78
profil 70
  poste de travail 79
promiscuous 29, 35
protocole
  IP (Internet Protocol) 2
proxy 148, 154, 163, 171
  ARP 167, 181
  rôle 163
ps 78
PubkeyAuthentication 61
Public Key Infrastructure 227

R

RADIUS 184
rc.d/ 30
RC2, RC4, RC5 49, 53
r-commandes 56
rcp 56
realm 223
rebond 25
Red Hat
  faille lprng 40
Red Hat Network 74
registered port 139, 151
règle du firewall
  SNMP 195
  syslog 192
relais 54
  ouvert 106
  X11 54, 64
Renater 42
répartition de charge 222
réplica 222
réseau
  Arpanet 2
  Internet 2
  Milnet 2
  privé virtuel 160
RHN (Red Hat Network) 74
RhostsAuthentication 61
RhostsRSAAuthentication 61
rlogin 56
RNIS (Réseau Numérique à Intégration de Service) 3
root
  /etc/securetty 82
  PATH 83
  umask 83
rootkit 33, 37, 38
  t0rn 38
routage par la source 87
route 158
routeur 154
rpm 57, 73
RSA 50, 53, 57, 61
RSAAuthentication 61
rsh 56
rst 138

S

sans fil 183
sauvegarde du système 29
scan 2, 26
  détection 201
  horizontal 26, 39
  vertical 26
scanner 26, 30, 197
scp 56, 58, 60
secret partagé 54
sécurité 5
  défaillance des systèmes 4
  enjeu 2, 14
  menace informatique 2

objectif 2
segmentation 153
sendmail 16, 105, 106, 109, 110
activation 109
m4 109
sendmail-cf 109
serveur
interne 16
service 12, 59, 75, 84
(/etc/services) 139
actif 77
désactivation 70, 78
état 79
réseau 77, 78, 96
sftp 56, 58, 60
sgid 79, 80, 106
danger 80
find 81
signature 47, 50, 52, 193, 201
signer 237
Single Sign On 224
smmsp 110
SMTP (Simple Mail Transfer Protocol) 13, 104
SNAT 165
sniff 46
sniffer 29, 30, 33
SNMP (Simple Network Management Protocol) 193, 194
configuration 195
Snort 16
snort 201
configuration 201
détection de scan 201
sonde 201
SOCKS 164
source routing 87, 88, 140
sous-réseau 156
spam 47, 106, 108
ssh 56, 77
/etc/ssh 58
authorized_keys 62
compression 60
config 58
configuration 60
id_dsa 62
id_dsa.pub 62
id_rsa 62
id_rsa.pub 62
relais 60
secure shell 54
ssh_config 58
ssh-keygen 61
SSH (Secure Shell) 16, 54–65, 75, 76
sshd 57, 58, 61, 64
sshd_config 58
SSL (Secure Socket Layer) 51–54, 132, 228
stateful 143
stateless 142
static 28
stéganographie 50
sticky 80
stunnel 127
configuration 127
configuration serveur 129
subnet 156, 160
sudo 81
sudoers 81
suid 79, 80, 106
/etc/fstab 82
alternative 81
danger 80
find 80
nosuid 82
suivi de connexion 138
surveillance 190, 192, 206
proportionnalité 192
réglementation 192
transparence 192
Swatch 190
syn 138, 142
SYN flooding 88
sysinit 30
syslog 84, 190, 206

T

tableau de bord 204
Tamalo.com 10
TCP (Transmit Control Protocol) 13, 139
drapeau 140
flag 141, 147
tcpdump 33
TCPWrapper 16, 83
telnet 54
TLS 51, 228
topologie 16, 154, 186
trace 190
traduction d'adresses 165
IPtables 174
transfert 60
Tripwire 16, 192, 193, 206
tunnel chiffré 56

U

UDP (User Datagram Protocol) 13, 139, 193
Unix 4

V

virus 106, 108
I love you 106
visioconférence 144
visudo 81
VLAN (Virtual Local Area Network) 160
limites 163
par adresse MAC 161
par port 160
vulnérabilité 15, 55, 140

W

warez 2, 25
web 125
protection 125
vulnérabilité 125
Webmail 127
well known port 139, 151
WEP 184
whitelist 107
whois 41
APNIC 41
ARIN 41
RIPE 41
Wi-Fi 183
WU-FTPD 151
wu-ftpd 12

X

X.509 52, 228
X11 64
xfs 79

xinetd 124
xinetd.conf 30
xinetd.d 30

Z

zone démilitarisée 186