

de tout inconnu le Sage se méfie.

JEAN DE LA FONTAINE

Préambule

J'utilise les différents systèmes dérivés d'Unix depuis 1985. J'administre ces systèmes depuis 1988. J'ai ainsi administré des ordinateurs utilisant Aix, HP-UX, Ultrix, Unix, SunOS, Solaris, NetBSD, FreeBSD, OpenBSD et plusieurs versions de GNU/Linux : Raspbian, RedHat, Slackware, Debian...

J'enseigne les télécommunications, les réseaux avec GNU/Linux, et les logiciels libres au département *Réseaux et Télécommunications* de l'IUT d'Aix-Marseille (Aix-Marseille Université). J'administre pour cela un petit réseau constitué de 10 serveurs, 70 stations, 15 segments réseaux et quelques centaines de réseaux virtuels (un par étudiant).

Je fais ma recherche sur la sécurisation de l'Internet des objets. Cette recherche est appliquée et pluridisciplinaire. Elle est appliquée car j'étudie comment sécuriser un système complet. Cela fait donc intervenir (un peu) la partie humaine, les logiciels, les systèmes d'exploitations, les méthodes de distribution des logiciels, le chiffrement, les transmissions, le matériel et le firmware caché... Elle regroupe donc de nombreuses disciplines.

Actuellement, beaucoup d'informations dans ces domaines se trouvent sur Internet. En fait, tout y est. Et même, plus encore. La question n'est pas de trouver de l'information, mais de trouver l'information *pertinente*. Dans le monde de l'informatique, certaines informations sont faciles à vérifier. Il est aisé de tester une commande et de regarder si le résultat est conforme.

Par contre, quand il faut choisir un logiciel à installer ou appliquer une configuration spécifique, il n'est pas prudent d'installer tout, n'importe comment dans son réseau. Il faut donc *faire confiance* à ce site. Qui a rédigé ce site? Il y a combien de temps? L'auteur est-il un étudiant qui rédige son projet sur internet ou un spécialiste du domaine? L'auteur est-il de bonne foi ou essaie-t-il de nous leurrer? Est-il conscient des conséquences en terme de sécurité, de sûreté de fonctionnement, de pérennité du logiciel?

Il est donc important de bien évaluer les conséquences des actions faites en suivant les conseils glanés sur Internet. D'ailleurs, ceci était déjà vrai *avant* Internet. Les indications techniques, comme installer un serveur web, devront donc être réalisées dans un environnement de test. Les conseils avisés, comme

le choix d'un serveur web, imposent de faire confiance à l'auteur du conseil. Il faut donc savoir *qui* est l'auteur et *pourquoi* son conseil est pertinent.

L'autre faiblesse d'Internet, qui est aussi plus générale, c'est de fournir de nombreuses informations sur *un* logiciel ou *un* aspect d'un élément. Il existe peu d'informations sur une approche holistique d'un réseau d'entreprise. Il ne s'agit pas de gérer un serveur web, mais de gérer celui-ci en lui ajoutant un langage de programmation, un serveur de bases de données. Ensuite, il faudra ajouter quelques applications web. Puis il faudra fournir des accès aux gestionnaires et contrôler leurs autorisations d'accès. Le tout doit être sauvegardé régulièrement pour récupérer une version récente après une défaillance matérielle. Il faudra aussi surveiller régulièrement tous les éléments pour détecter, voire prévenir les interruptions de service. Il faut gérer le réseau dans sa globalité.

J'ai voulu écrire ce livre pour combler un vide. Il a pour but de montrer comment mettre en place un réseau élémentaire mais complet. Ce livre ne demande pas de prérequis. La connaissance de la langue (écrite) anglaise est importante. La maîtrise des ordinateurs n'est pas nécessaire. Par contre, de nombreux ouvrages permettront d'affiner les différents éléments.

Cette méthode est issue de mes enseignements. J'enseigne à des étudiants dont certains n'ont presque jamais utilisé d'ordinateur avant leur arrivée en France. Après un temps d'adaptation, ils sont aussi efficaces que les étudiants habitués aux ordinateurs depuis leur enfance.

Le logiciel libre, c'est quoi ?

Commençons par définir les logiciels libres. La notion de *logiciel libre* a été formalisée par Richard M. Stallman[16]. Il définit quatre propriétés (parfois numérotées de 0 à 3) :

1. la liberté d'exécuter le programme, pour tous les usages ;
2. la liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins ;
3. la liberté de redistribuer des copies du programme ;
4. la liberté d'améliorer le programme et de distribuer ces améliorations.

La plupart des humains n'imaginent pas pouvoir modifier un logiciel. Pour-tant, une modification importante que tous peuvent faire, c'est d'adapter un logiciel en le traduisant ou en corrigeant l'orthographe. La suite bureautique *LibreOffice* est disponible dans plus de cent langues différentes, dont le breton et l'espéranto. En comparaison, la suite privatrice concurrente ne propose que de trente langues.

La notion de *logiciel libre* s'oppose à deux autres notions : les logiciels à source ouverte (*open source*) et les logiciels privés.

Le mouvement de l'*open source*[5] est né après le *logiciel libre*. La formalisation du logiciel libre s'est faite sur des critères humains et philosophiques de partage de la connaissance. L'*open source* se définit sur des critères techniques. C'est la principale différence entre les deux définitions. Un logiciel libre est *open source* et réciproquement.

L'autre opposition concerne les logiciels privés. Un logiciel libre respecte les libertés de l'utilisateur. Un logiciel privé interdit certaines actions (utilisation dans un cadre professionnel, copie, étude du fonctionnement, modification...). La propagande de certains vendeurs abuse du qualificatif de *propriétaire*. Il est certes plus vendeur de proposer un logiciel *propriétaire* que *privé*. Cependant les logiciels libres sont des logiciels propriétaires et restent la propriété de leurs auteurs. C'est *l'usage* qui est libéré.

Les logiciels libres ne sont pas *libres de droits*. Certains logiciels, sous licence GPL, imposent une restriction aux utilisateurs. Cette restriction consiste à interdire à un utilisateur de verrouiller le logiciel et de le redistribuer avec une licence privée.

Beaucoup de gens croient que les logiciels libres sont gratuits. Ceci est renforcé en anglais par le double sens du mot *free* : libre ou gratuit. D'où l'expression : « *free, as in free speech, not free beer* » (Libre, comme liberté de parole, pas comme bière gratuite). Les logiciels libres sont payants ! Pour développer une nouvelle fonctionnalité, il faut que quelqu'un la paie. Soit en payant un développeur, soit en payant de son temps. Le travail est payé, la copie est gratuite.

La sécurité des logiciels libres

Un des points importants, concernant le logiciel libre, c'est la sécurité. Certes, il ne suffit pas de publier son code sous licence libre pour qu'il devienne magiquement sécurisé. Néanmoins, l'histoire fournit des arguments implacables. Voici quelques dates clés :

- 1883** la cryptographie militaire[7]
- 1985** publication du *livre orange*[12]
- 1988** premier virus sur Internet : le Ver Morris ;
- 1997** publication des *Halloween Documents* ;
- 2013** révélations d'Edward Snowden[15].

Le *livre orange* est un document fondateur pour la sécurité des systèmes informatiques. Il montre, en particulier, que pour pouvoir envisager d'être

sécurisé, un système informatique doit publier son code source. En cryptographie, la publication de l'algorithme est un des principes nécessaires, définis par Auguste Kerckhoffs en 1883[7].

Le ver Morris est le premier qui s'est diffusé sur Internet. Il a infecté 10% des ordinateurs connectés et a bloqué le réseau. Il a permis de prendre conscience que les systèmes d'exploitation ne devaient pas exécuter n'importe quel code venu de l'extérieur. Cette leçon fait que les systèmes GNU/Linux, bien gérés, ne sont pas victimes de virus. La leçon n'a pas été apprise par tous et beaucoup d'*objets connectés* restent vulnérables à ce genre d'attaque.

Les *Halloween Documents*[20], censés rester secrets, sont une présentation des dangers que présente GNU/Linux et les logiciels libres pour Microsoft. Eric S. Raymond remercie d'ailleurs le travail de l'entreprise pour « *authoring such remarkable and effective testimonials to the excellence of Linux and open-source software in general* ». Ces documents montrent, en particulier, la supériorité des logiciels libres pour la sécurité. Quelques années plus tard, Bernard Ourghanlian (directeur Technique et Sécurité à Microsoft France) reconnaît la supériorité des logiciels *open source* sur France Culture[4].

Enfin, les documents libérés (ou volés) par Edward Snowden révèlent l'ampleur de la collaboration entre les multinationales de l'information et la NSA. Deux questions se posent alors. Pourquoi les autres pays n'ont pas interdit les logiciels, services et équipements qui fournissent des renseignements à une puissance étrangère ? Les agences de renseignement protègent-elles les citoyens contre les menaces ou les multinationales et les gouvernements contre les citoyens ?

La question subsidiaire qui concerne plus l'administrateur réseau, c'est de savoir si les agences de renseignement introduisent des portes dérobées qui pourraient être utilisées par d'autres malfaisants, comme les attaques récentes *WannaCry* ou *NotPetya*[22] ?

Enfin, une question pragmatique. Pourquoi une entreprise investirait pour la sécurisation de ses anciens équipements ? Les utilisateurs ne sont pas très regardants, ce n'est donc pas un critère d'achat. La licence des logiciels privés indique qu'il n'y a *aucune garantie* (comme pour les logiciels libres). De plus, il est souvent interdit d'essayer de comprendre ce que fait le programme et encore plus de le corriger. Enfin, en cas de dysfonctionnement, l'acheteur est invité à acheter le nouveau produit (comme pour les machines à laver).

Le logiciel libre n'est pas magiquement sécurisé. Il ne suffit pas de publier un code pour qu'il devienne sûr. De plus, en particulier dans le domaine de l'embarqué ou de l'Internet des objets, il n'y a pas de mécanisme de mise à jour efficace. Ce n'est pas Linux qui est sécurisé, c'est Debian et les autres distributions sérieuses.

Tous les logiciels un peu complexes ont des failles. Pour un logiciel privé, la faille est détectée quand un malfaisant l'utilise. Pour le logiciel libre, au moins pour les plus utilisés, certains chercheurs scannent le code source à la recherche d'erreurs de programmation. Il est fréquent de détecter des vulnérabilités *avant* qu'elles ne soient exploitées. Le chercheur indique alors l'erreur au gestionnaire du programme qui corrige et propose une mise à jour.

Les équipes de sécurité des distributions GNU/Linux majeures collaborent pour fournir simultanément les correctifs. La sécurité est très importante pour les distributions GNU/Linux. Elle est prise en compte dans son ensemble. c'est-à-dire pas uniquement pour l'utilisation d'un protocole, d'un algorithme de cryptographie ou d'un logiciel isolé. L'ensemble du système : matériel, logiciels et leurs interactions sont surveillés. C'est pourquoi les informations de sécurité sont présentées tout au long de cet ouvrage. Il n'y a pas de partie isolée pour la sécurité, même si la dernière partie présente des outils de surveillance et l'utilisation du chiffrement.

Quelques détails techniques

Un reproche souvent fait aux logiciels libres, c'est que « c'est moins facile ! ». Cet argument ne tient pas à l'examen des faits. Une démonstration simple consiste à demander à deux experts (un libre, un privé) d'installer un ordinateur avec une suite bureautique, un logiciel de dessin vectoriel, un logiciel de retouche photo, un serveur web, les compilateurs Java et C. Au début de l'expérience il leur est fourni un ordinateur vierge et un accès internet. Le libriste finit environ deux heures plus tard, le privé mettra quelques jours s'il n'y a pas de problème technique. De plus, un des éléments passé sous silence, c'est la gestion des achats et la gestion des licences privées. L'utilisateur et l'administrateur devraient lire et accepter *chaque* licence et gérer les droits des logiciels, en particulier la copie, la transmission et l'expiration.

L'autre élément effrayant, c'est l'utilisation de la ligne de commande. La ligne de commande n'est pas obligatoire pour utiliser une station graphique ou de nombreux équipements utilisant les logiciels libres. Donc, cet argument n'est pas non plus très honnête. Si *moi*, j'utilise la ligne de commande, je n'interdis pas à d'autres de s'en passer. Par contre, l'utilisation de la ligne de commande, c'est comme le langage articulé : cela simplifie grandement les interactions. Essayez de rentrer dans une boulangerie pâtisserie pour commander deux baguettes pas trop cuites, trois tartelettes aux fraises (pas celle qui est abîmée), quatre croissants au beurre, un pain au chocolat et de commander une charlotte aux fraise pour dimanche prochain sans parler et avec un seul doigt !

Enfin, une erreur consiste à essayer de plaquer sur le logiciel libre les usages issus des limitations des logiciels propriétaires. Ainsi, pour faciliter la réinstallation d'un ordinateur, il est possible de prendre une image du disque dur. Pour les systèmes propriétaires, c'est plus facile que de tout installer de nouveau. Un des inconvénients, c'est que l'image est figée, la procédure est relativement lourde. Pour les systèmes libres, il est plus efficace de refaire l'installation et d'appliquer les modifications des fichiers de configuration. C'est plus efficace et permet de bénéficier des mises à jour de sécurité.

Beaucoup de logiciels libres sont très efficaces. Depuis l'origine, une des philosophies des développeurs Unix consiste à faire un programme qui ne fait qu'une seule chose, mais qui la fait bien. Pour les logiciels vendus, il est fréquent que les nouvelles versions ajoutent de nouveaux éléments pour justifier l'achat. Le logiciel devient alors plus complexe et donc moins efficace.

De plus, les logiciels libres sont développés par des utilisateurs. Ce n'est pas une équipe de direction qui impose une direction. Si le logiciel souffre d'un handicap, il y aura des développeurs qui vont corriger le problème. Mon expérience, c'est que les logiciels que j'utilise (dont le système d'exploitation) sont de plus en plus efficaces et faciles à utiliser. Mais comme il y a de plus en plus de fonctionnalités, un ordinateur plus puissant est le bienvenu.

Enfin, j'ai choisi la distribution Debian GNU/Linux pour sa stabilité. Debian publie irrégulièrement de nouvelles versions. La version actuelle (été 2020) est nommée *Buster*. Elle est aussi labellisée *stable*. Il y a en permanence une et une seule version stable. La précédente (*Stretch*) prend le label *oldstable* (ancienne stable). La prochaine (*Bullseye*) est la version *testing*.

Les logiciels de la version stable sont figés. Les erreurs sont corrigées, y compris quelques mois ou années après son déclassement en *oldstable*. Le comportement ne peut être modifié que par une mise à jour de sécurité. Il est donc possible d'installer un serveur et de le mettre à jour quotidiennement sans modifier son comportement. Mon expérience sur vingt ans montre que cette stabilité est réelle. L'inconvénient, c'est de ne pas avoir accès aux dernières versions des logiciels, ce qui est rarement un défaut pour les serveurs en production.

Liberté de choix

Parmi les libertés offertes par les logiciels libres, il faut choisir le logiciel à utiliser, voire la version à utiliser. Comme plusieurs logiciels fournissent le même service, les adeptes de l'un s'opposent aux autres dans un *troll* d'autant plus beau qu'il est inutile. Pour un débutant, il est difficile de choisir le *meilleur* logiciel. Un tel logiciel n'existe pas. C'est pourquoi dans cet ouvrage, je guide

le lecteur en proposant directement le ou les logiciels à utiliser. En général, passer d'un logiciel à un autre reste relativement facile. Dans de nombreux cas, la configuration des deux logiciels est très similaire.

Le choix du bon logiciel peut dépendre de l'expérience de l'utilisateur. Ainsi, pour modifier un fichier, il faut utiliser un éditeur de textes. Il existe des éditeurs graphiques et d'autres non graphiques. Pour un administrateur qui doit pouvoir administrer un réseau dans des conditions parfois délicates, il est important de maîtriser un éditeur non graphique. Le débutant commencera avec *nano*, très facile à prendre en main mais très limité. Avec l'expérience, il se tournera vers *emacs* ou *vi*, les meilleurs.

Dans d'autres cas, le choix dépendra des critères locaux. Ainsi, on choisira le logiciel déjà connu, le plus performant, le plus configurable, celui recommandé par un collègue plus compétent. . . . Voire ne pas choisir et en utiliser plusieurs pour des usages distincts. Il n'y a pas de raisons d'essayer de rentabiliser un logiciel déjà acheté en l'utilisant pour une tâche qu'il ferait médiocrement.

Enfin, certains logiciels peuvent devenir obsolètes. Ainsi, depuis le début du siècle, la commande `ifconfig` (créée en 1983) est remplacée par les commandes de la suite *iproute2*. Néanmoins, elle est encore disponible aujourd'hui. Ces logiciels commencent par montrer des faiblesses, une équipe propose une nouvelle version, l'ancienne est considérée comme obsolète. Pour l'utilisateur, il est averti qu'il faudra changer un jour ou l'autre, il changera quand il sera prêt ou parce qu'il aura besoin des nouvelles fonctionnalités. Une autre obsolescence est liée à la *propriété intellectuelle*. Les logiciels portent souvent un nom. La loi est formelle, ce nom ne peut pas être libre ! Quand Linus Torwald a nommé son programme *Linux*, il n'a pas acheté le mot. Plusieurs demandes ont été effectuées en 1994 et 1995 pour déposer ce mot par d'autres acteurs. Finalement, Linus Torwald a obtenu la propriété du mot et l'a transféré à la Fondation *Linux Mark Institute*. Dans d'autres cas, le propriétaire du nom d'un logiciel fait des choix qui n'ont pas convenu aux utilisateurs. Le logiciel a donc changé de nom. Ainsi *Ethereal* est devenu *wireshark* ; *XFree86*, *Xorg* ; *OpenOffice*, *LibreOffice* ; *Nagios*, *Icinga*. . . Le code est libre, pas le nom.

Plan de l'ouvrage

Cet ouvrage est divisé en quatre parties :

1. une brève théorie des réseaux ;
2. installation du réseau local ;
3. le Web 4.0 ;
4. surveillance et cryptographie.

Nous commençons par présenter les concepts et les outils utiles pour la gestion, le diagnostic et la supervision du réseau d'entreprise. Cette partie est relativement théorique et présente les concepts minimums à maîtriser par l'administrateur système et réseaux. Nous commençons par montrer les protocoles du modèles TCP/IP, puis les commandes fondamentales pour diagnostiquer son réseau et enfin le fonctionnement de Linux comme routeur et firewall.

La deuxième partie présente la mise en place du réseau. Nous détaillons la procédure d'installation du système d'exploitation pour une station de travail ou un serveur. Nous mettons ces équipements en réseau pour partager les fichiers et permettre la connexion de chaque utilisateur sur toutes les stations. Enfin, nous présentons une procédure efficace de sauvegarde.

Les serveurs applicatifs sont de plus en plus intégrés dans un serveur web. Ainsi, ils sont utilisables par un simple navigateur. Il n'ont plus besoin d'un logiciel de contrôle à installer sur le poste client avec les limitations souvent constatées sur ce genre de logiciels. Donc nous montrons comment mettre en place une infrastructure pour ces applications et proposons deux exemples : un wiki et un nuage (*cloud*) personnel.

Enfin, la dernière partie présente la surveillance et l'utilisation de cryptographie. Nous commençons par deux logiciels de supervision. Ensuite, nous montrerons comment établir des connexions sécurisées à travers un réseau incertain, puis comment chiffrer le trafic web et les disques durs.

Conventions

Voici quelques conventions utilisées dans cet ouvrage, au-delà du respect du code typographique.


- Les logiciels sont mentionnés en *italique*, les noms de paquets, de fichiers et les commandes en style **machine à écrire**.
- Les commandes sont parfois écrites sur plusieurs lignes. Dans ce cas, il est d'usage d'ajouter un caractère *barre inversée* (`\`) en fin de ligne pour indiquer qu'elle continue sur la ligne d'après.
- Le contenu des fichiers et l'affichage produit par une commande sont souvent volontairement tronqués. Les informations non essentielles sont effacées et de nombreuses lignes oubliées. Parfois, les parties effacées sont remplacées par des points de suspension.
-  Les informations importantes pour la sécurité sont affichées avec un logo spécifique pour attirer l'attention.
- Nous définissons les termes peu courants dans le glossaire.

Table des matières

I	Une brève théorie des réseaux	15
1	Le modèle TCP/IP	19
1.1	Les modèles théoriques	20
1.2	La couche liaison de données	21
1.2.1	Le protocole local : Ethernet	23
1.2.2	La liaison sans fil, dont le Wi-Fi	24
1.2.3	Communication longue distance	25
1.2.4	Établissement de la liaison numérique	26
1.3	La couche réseau	27
1.3.1	Relation avec la couche liaison	27
1.3.2	Le routage	29
1.3.3	Les routeurs	30
1.4	La couche transport	30
1.5	La couche application	32
1.6	Les autres réseaux	33
2	Les utilitaires réseau	35
2.1	La couche liaison de données	35
2.1.1	Valider les câbles	36
2.1.2	Valider les transmissions hertziennes	37
2.1.3	Valider la couche liaison de données	40
2.2	La couche réseau	47
2.2.1	Configurer une interface	48
2.2.2	Configurer les routes	51
2.2.3	Résolution de nom	52
2.2.4	Outils de mise au point	52
2.3	La couche transport	57
2.4	La couche application	57
2.4.1	Le service de noms	58
2.4.2	fournir les paramètres réseaux : DHCP	59

2.4.3	Transfert de fichier sans contrôle : TFTP	61
2.4.4	Synchronisation des horloges : NTP	62
3	Arrêter les ennemis par le firewall	63
3.1	Le firewall Linux : netfilter	63
3.2	La sécurité apportée par un firewall	65
3.2.1	Protection d'un serveur Internet	65
3.2.2	Protection de l'intranet	66
3.2.3	Séparation des domaines	66
3.2.4	Protection contre les objets internes	66
3.3	Comment le firewall Linux fonctionne	67
3.4	Quelques exemples	69
3.4.1	Sélection d'un paquet	70
3.4.2	Définir des actions	70
3.4.3	Les règles pour le firewall	71
II	Mettre les ordinateurs professionnels en réseau	77
4	Un système d'exploitation libre	81
4.1	Installation d'une Debian stable : Buster	83
4.1.1	Le démarrage d'un ordinateur	83
4.1.2	La procédure d'installation	85
4.2	Imprimantes	93
5	Le partage de fichiers	95
5.1	Network File System	96
5.1.1	Premier montage	96
5.1.2	Les commandes pour NFS	99
5.1.3	Les fichiers	101
5.2	Système de fichiers en espace utilisateur	102
5.2.1	Accéder aux fichiers à travers ssh	102
6	La gestion des utilisateurs	105
6.1	Les comptes locaux	105
6.2	Serveur de compte : LDAP	111
6.2.1	Installation du serveur LDAP	111
6.2.2	Gérer les enregistrements dans l'annuaire	112
6.2.3	Ajouter des clients LDAP	117
6.2.4	Sauvegarde et restauration	119

<i>TABLE DES MATIÈRES</i>	13
7 La sauvegarde des données	121
7.1 Politique de sauvegarde	122
7.1.1 Choisir les données à sauvegarder	122
7.2 Borgbackup : sauvegarder est utile	124
7.3 Restaurez les données	126
III Le Web 4.0	129
8 L'infrastructure pour le Web	133
8.1 L'infrastructure LAMP	133
8.1.1 Installer le serveur Apache	135
8.1.2 Support de la programmation web : PHP	147
8.1.3 Le gestionnaire de bases de données	149
8.1.4 Installation d'un service Web, l'exemple de MediaWiki	152
9 Le nuage dont vous êtes le héros	159
9.1 Le cœur de votre prochain nuage	160
9.2 Mise à jour	165
9.3 Un client pour le bureau	165
IV Quelques bribes de sécurité	169
10 Superviser	173
10.1 Munin	174
10.2 Superviser les services avec Icinga	177
10.2.1 Installer l'ordonnanceur Icinga	178
10.2.2 Installer l'interface IcingaWeb	179
11 Éléments de cryptographie	191
11.1 Le chiffrement asymétrique	192
11.2 La boîte à outils : openssl	194
11.3 Connexion chiffrée par ssh	194
11.3.1 Connexion sécurisée vers un serveur	195
11.3.2 Connexion sans mot de passe	196
11.3.3 Utilisation d'une phrase de passe	198
11.3.4 Délocalisation de port	199
11.4 Le Réseau Privé Virtuel : OpenVPN	202
11.4.1 Architecture d'OpenVPN	203
11.4.2 L'infrastructure de gestion de clefs	204
11.5 Chiffrer le disque dur	213

11.5.1	Chiffrement initial	214
11.5.2	Récupérer un disque chiffré	216
11.5.3	Chiffrer un disque après installation	217
11.5.4	Gérer les phrases de passe	218
11.6	Cachez ce port que je ne saurais voir	219
11.6.1	knockd	220
Glossaire		227
Bibliographie		231
Index		233