OpenSSH

Présentation pour le groupe SUR (Sécurité Unix et Réseaux)

08/03/2005

Saâd Kadhi <saad.kadhi@hapsis.fr>

Agenda

- Un peu d'histoire et plus encore
- Fonctionnalités principales
- Mécanismes de sécurité
- Conclusion, Références, Remerciements

Un peu d'histoire et plus encore

La vie avant

- Plusieurs outils pour administrer les systèmes Unix/Linux
- telnet, ftp, commandes R, X11
- Flux en clair
- Authentification faible, voire inexistante
- Ces outils ne sont pas adaptés à nos environnements actuels

SSH fait son entrée

- Tatu Ylönen développe alors SSH
  - Le protocole et l'implémentation
- Protocole permettant une communication TCP chiffrée entre deux hôtes
- Charge utile TCP chiffrée

SSH, l'entreprise

- Succès de ce nouveau protocole et de l'implémentation associée
- Tatu décide de rendre son produit payant
- SSH Communications Security voit le jour en 1995

Liberté

- La dernière version de *ssh* développée par Tatu considérée comme libre fût la 1.2.12
- Les versions suivantes fûrent de moins en moins libres jusqu'à devenir entièrement propriétaires
- Björn Gronvall et OSSH entrent en scène en 1999

OSSH et OpenSSH

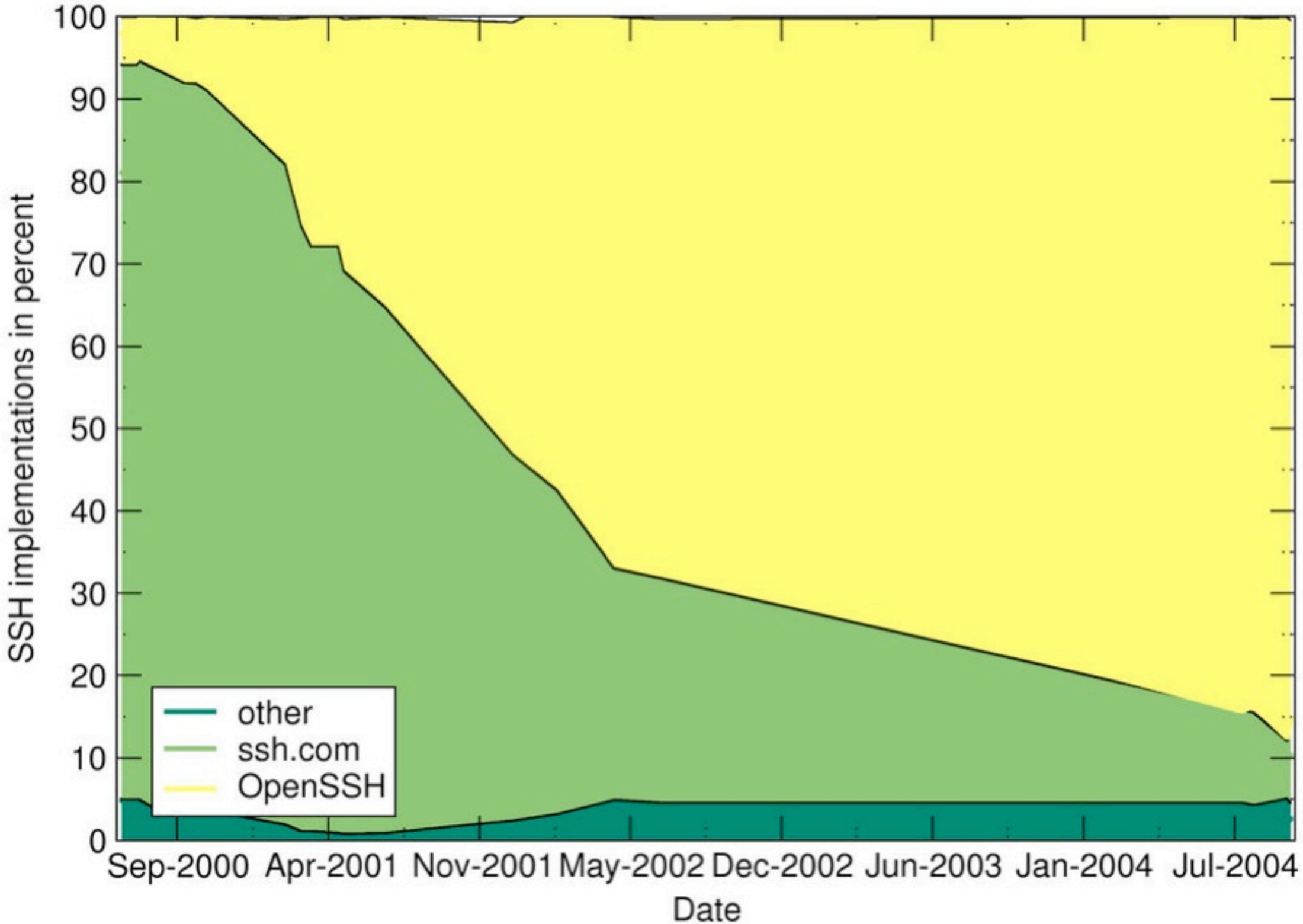
- OSSH est basé sur *ssh* 1.2.12 et ne supportait que la version 1.3 du protocole SSH
- OpenSSH fût créé à partir de OSSH en octobre 1999
  - Par l'équipe OpenBSD
- La première version de OpenSSH fût développée très rapidement

Pourquoi OpenSSH ?

- Avoir une implémentation libre du protocole SSH
  - La majorité du code est sous licence BSD
- Supporter les différentes versions du protocole SSH
  - 1.3, 1.5, 2.0
- Fournir une sécurité accrue
  - Audit permanent du code

Hier et aujourd'hui

- OpenBSD 2.6 fût le 1er OS à intégrer OpenSSH
- Le succès fût immédiat et le portage vers d'autres OS commence rapidement
- Aujourd'hui, OpenSSH est très probablement l'implémentation SSH la plus utilisée


100
90
80
70
60
50
40
30
20
10
0
SSH implementations in percent
other
ssh.com
OpenSSH
Sep-2000
Apr-2001
Nov-2001
May-2002
Dec-2002
Jun-2003
Jan-2004
Jul-2004
Date

Deux formes

- OpenSSH existe sous deux formes
- OpenSSH pour OpenBSD (openssh-3.9.tgz)
- Portable OpenSSH pour les autres OS (openssh-3.9p1.tgz)
  - Linux, Mac OS X, cygwin, FreeBSD, NetBSD, Solaris ...
  - couche de portabilité ajoutée à la version OpenBSD

Sponsors ?

- Comme le projet OpenBSD, le “chantier” OpenSSH vit de la vente de T-shirts et de donations
- Aucun sponsor qui pourrait restreindre la liberté de OpenSSH

Documentation

- Les pages du manuel sont d'excellente facture
- Le site web est entièrement traduit et à jour
  - Améliorations en cours
- Ouvrage O'Reilly entièrement consacré à SSH
- De multiples articles existent sur le sujet

Pré-requis

- OpenSSH s'appuie fortement sur OpenSSL
- Zlib pour la compression des flux
- Perl lors de l'installation mais pas pour le fonctionnement
- Et n'oublions pas une bonne source d'entropie !
  - /dev/random, PRNGD

Fonctionnalités principales

Une implémentation riche

- Composants
- Algorithmes
- Authentification
- Filtrage d'accès
- Tunnels TCP

Composants

- Client/Serveur : ssh et sshd
- SFTP : sftp et sftp-server
- Outils de “confort” : ssh-agent, ssh-add, scp, ssh-keyscan
- Génération de biclefs d’authentification : ssh-keygen
- Autres : ssh-keysign

Algorithmes

- Algorithmes de chiffrement à clef publique
  - RSA, DSA (SSH 2.0)
- Algorithmes symétriques
  - Arcfour, Blowfish, 3DES, AES
- Algorithmes MAC (SSH 2.0)
  - hmac-md5, hmac-sha1, hmac-ripemd160

Authentification

- Mot de passe Unix
- Biclefs RSA/DSA
- S/Key
- Kerberos V
- SmartCard via OpenSC et Sectok
- PAM

Filtrage d'accès

- Utilisateur (AllowUsers)
- Groupe (AllowGroups)
- Clef (authorized_keys)
- Pour chaque clef
  - adresse IP (from), commande (command), allocation de pty, création de tunnels (permitopen & co.) ...

Tunnels TCP - 1

- OpenSSH permet la création de tunnels chiffrés TCP
  - Applications TCP à port “fixe” : IMAP, POP, ...
  - X11 Forwarding
- Trois types de tunnels
  - Local, Remote, Dynamic (SOCKS)

Tunnels TCP - 2

- Permettent de sécuriser de façon très simple des échanges en clair
- Attention, ils peuvent servir de "covert channels"
  - Peuvent être encapsulés dans du trafic HTTP/HTTPS à l'aide d'outils comme corkscrew ou httptunnel

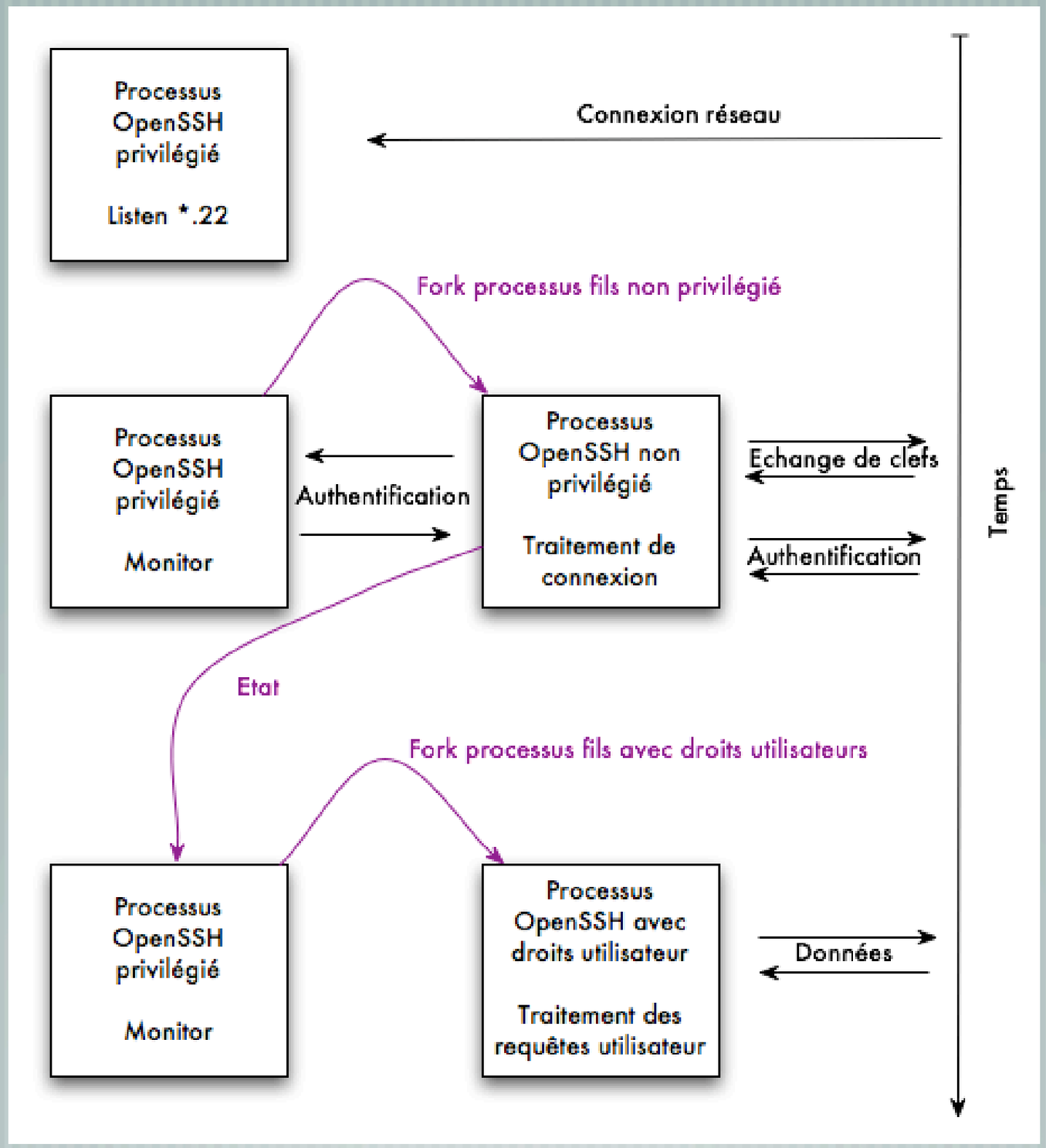
Nouveautés version 3.9(p)

- IdentitiesOnly
- MaxAuthTries
- Vérification des permissions ssh_config
- Améliorations SFTP
- Multiplexage de sessions : ControlPath, ControlMaster
- Suppression de tunnels distants

Mécanismes de sécurité

PrivSep

- Distinguer les opérations privilégiées, réduites au strict minimum, des autres opérations
  - Traitements effectués par des processus différents
- Introduit par OpenSSH 3.4(p)
- Utilisé par d'autres logiciels
  - OpenNTPD, OpenBGPD, postfix, vsftpd ...


Processus
OpenSSH
privilégié
Listen *.22
Connexion réseau
Fork processus fils non privilégié
Processus
OpenSSH
privilégié
Monitor
Authentification
Processus
OpenSSH non
privilégié
Traitement de
connexion
Echange de clefs
Authentification
Temps
Etat
Fork processus fils avec droits utilisateurs
Processus
OpenSSH
privilégié
Monitor
Processus
OpenSSH avec
droits utilisateur
Traitement des
requêtes utilisateur
Données

ProPolice/SSP

- La version OpenBSD est compilée par défaut en utilisant ProPolice/SSP (Stack-Smashing Protector)
  - Comme tout programme OpenBSD
- Protection contre les dépassements de tampon
  - Notion de canari
- Support en cours par Gentoo Hardened

Vérification de clef - 1

- Le client ssh peut refuser de se connecter à un serveur
  - si la clef publique RSA/DSA n'est pas connue
  - si elle a changé
  - directive StrictHostKeyChecking dans ssh_config
- Possibilité de collecter les clefs publiques des serveurs à l'aide de ssh-keyscan

Vérification de clef - 2

- Le client ssh peut vérifier la clef publique RSA/DSA d'un serveur SSH à l'aide du DNS
  - directive VerifyHostKeyDNS dans ssh_config
  - Resource Records de type SSHFP (BIND ≥ 9.3.0) ou TYPE 44
  - ssh-keygen -r <hostname> -f ssh_host_[rd]sa_key.pub

Autres mécanismes

- OpenSSH pour OpenBSD bénéficie des mécanismes sécurité mis en place par cet OS
  - ld.so randomization, W^X, ...
- Portable OpenSSH s'interface avec PAM et grsecurity
- Il est conseillé de durcir la configuration de sshd
  - Utiliser la profusion de directives dans sshd_config

Conclusion, Références,

Remerciements

Conclusion

- OpenSSH est une suite d'outils fonctionnellement très riche
  - Interopérable avec un grand nombre de clients/serveurs libres et propriétaires
  - Peut-on s'en passer aujourd'hui ?
- Outil à double tranchant
  - Améliore la sécurité mais permet de la contourner aussi

Références

- http://www.openssh.com/fr/
- http://www.gentoo.org/proj/en/keychain/index.xml
- http://www.oreilly.com/catalog/sshtdg/index.html
- http://www.citi.umich.edu/u/provos/ssh/privsep.html
- http://www.research.ibm.com/trl/projects/security/ssp/

Remerciements

- HAPSIS, mon employeur
  - http://www.hapsis.fr/
- Dave McMurray et Doc Powell
  - Pour la musique ;-)
- Et vous tous
  - Pour m'avoir écouté !