Guide de formation avec ateliers pratiques

Linux Administration

Jean-François Bouchaudy

Les Guides de formation Tsoft
Plus de 40 000 personnes formées à Unix et Linux

Tome 3

Sécuriser un serveur Linux

2e édition

Tsoft EDITEUR

EYROLLES

Linux Administration

Les Guides de formation Tsoft
Rédigés par des professionnels de la formation, *les Guides de formation Tsoft* ont été adoptés par de nombreuses entreprises comme supports de cours ou manuels d'autoformation. Chaque manuel est découpé en modules thématiques présentés sous forme de fiches descriptives illustrées de nombreux exemples de commandes et de scripts. Chaque module se termine par une série de travaux dirigés minutés pour mettre immédiatement en pratique les notions acquises.

Après des études de doctorat en biophysique, **Jean-François Bouchaudy** s'est orienté vers l'informatique en se spécialisant dans le développement d'applications en C et d'applications TCP/IP sous Unix. Il anime des formations sur les thèmes suivants : administration Unix et Linux, programmation système Unix, langages C, C++ et Perl, TCP/IP, Samba, sécurité réseau, etc.

Code éditeur : G13462
ISBN : 978-2-212-13462-9

Sécuriser un serveur Linux

Après deux premiers tomes dédiés aux bases et aux aspects avancés de l'administration système, ce troisième tome de la série *Linux Administration* traite de toutes les facettes de la sécurité Linux : sécurité de connexion, sécurité réseau, pare-feu, audit de sécurité, politiques de sécurité, etc.

Très pragmatique dans son approche, l'auteur va à l'essentiel avec des fiches de cours synthétiques, accompagnées d'ateliers qui forment un recueil de recettes pratiques pour la gestion au quotidien de la sécurité d'un serveur Linux.

Cette seconde édition prend en compte l'évolution des outils et des usages, et se fait l'écho de la montée en puissance d'outils comme SELinux, OpenVPN, OpenVAS ou Metasploit.

L'auteur a choisi de mettre l'accent sur le mode commande, plutôt que sur les outils graphiques fournis avec les distributions. La connaissance des fichiers et des commandes qui se cachent derrière ces outils est en effet indispensable aux administrateurs opérant dans un contexte professionnel, et offre l'avantage d'une certaine indépendance vis-à-vis des distributions. Les quelques variantes sont indiquées dans des sections intitulées *Les particularités des distributions*, qui couvrent Red Hat/Fedora/CentOS et Debian/Ubuntu.

Au sommaire

Tsoft EDITEUR

www.editions-eyrolles.com

Linux
Administration

CHEZ LE MÊME ÉDITEUR

Dans la collection *Les guides de formation Tsoft*

J.-F. Bouchaudy. – **Linux Administration. Tome 1 : les bases de l'administration système**.
N°12624, 2e édition, 2009, 270 pages.

J.-F. Bouchaudy. – **Linux Administration. Tome 2 : administration système avancée**.
N°12882, 2e édition, 2011, 504 pages.

J.-F. Bouchaudy. – **Linux Administration. Tome 4 : les services applicatifs Internet (Web, email, FTP).**
N°12248, 2009, 400 pages.

R. Bizoï. – **Oracle 11g Administration**.
N°12898, 2011, 880 pages.

R. Bizoï. – **Oracle 11g Sauvegarde et restauration**.
N°12899, 2011, 420 pages.

F. Joucla, J.-F. Rouquié. – **Lotus Domino 8.5 Administration – Tome 2. Gestion et optimisation.**
N°12791, 2010, 380 pages.

F. Joucla, J.-F. Rouquié. – **Lotus Domino 8.5 Administration – Tome 1. Installation et configuration.**
N°12790, 2010, 420 pages.

Autres ouvrages

R. Stallman, S. Williams, C. Masutti. – **Richard Stallman et la révolution du logiciel libre.**
N°12609, 2010, 324 pages (Collection Accès libre).

J. Gabès. – **Nagios 3 pour la supervision et la métrologie.** *Déploiement, configuration et optimisation*.
N°12473, 2009, 482 pages (Cahiers de l'Admin).

P. Hansteen, adapté par M. Derche. – **Le livre de Packet Filter (PF).**
N°12516, 2009, 194 pages (Cahiers de l'Admin).

R. Hertzog, R. Mas. – **Debian Squeeze.**
N°13248, 2011, 476 pages avec DVD-Rom (Cahiers de l'Admin).

L. Dricot. – **Ubuntu efficace.** *Ubuntu 9.04 « Jaunty Jackalope »*
N°12362, 3e édition, 2009, 344 pages avec CD-Rom (Collection Accès libre).

C. Blaess. – **Shells Linux et Unix par la pratique.**
N°12273, 2008, 262 pages (Collection Blanche).

C. Blaess. – **Développement système sous Linux.**
N°12881, 3e édition, 2011, 1004 pages (Collection Blanche).

P. Ficheux. – **Linux embarqué.**
N°12452, 3e édition, 2011, 378 pages (Collection Blanche).

I. Hurbain. – **Mémento Unix/Linux**
N°13306, 2e édition, 2011, 14 pages.

Linux Administration

Tome 3

Sécuriser un serveur Linux

Jean-François Bouchaudy

Tsoft EDITEUR

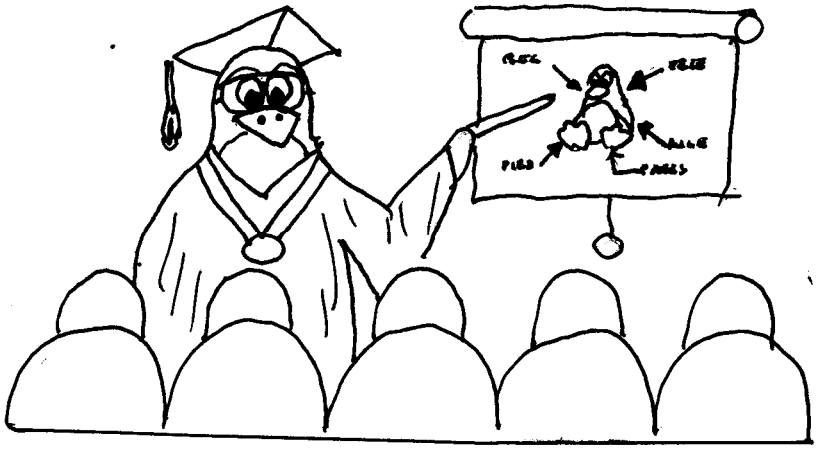
EYROLLES

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

TSOFT
10, rue du Colisée
75008 Paris
www.tsoft.fr

À ma fille Alice

Avant-propos



Présentation de l'ouvrage

Aujourd'hui, il n'est plus besoin de présenter Linux, même les non-informaticiens le connaissent, et certains l'utilisent à titre personnel. Dans les entreprises et les administrations, il est encore peu présent sur le poste de travail mais il envahit de plus en plus de serveurs… Et les serveurs doivent être protégés.

Ce livre traite de leur sécurisation. Il fait suite aux deux tomes précédents, qui abordent respectivement les tâches élémentaires et avancées de l'administration système Linux.

L'objectif principal de ce troisième tome est la sécurisation d'un serveur Linux. Ce sujet précis fait d'ailleurs l'objet de deux modules spécifiques (sécurisation des applications et sécurisation du serveur) mais il reste sous-jacent à l'ensemble du texte. L'ensemble des techniques de sécurisation est abordé : les protocoles réseaux qui sécurisent les transferts (SSH, SSL, Kerberos, IPSec, OpenVPN…), les techniques pare-feu (Iptables, Wrappers, Squid …), l'utilisation de techniques d'audit (HIDS, NIDS…). La sécurité locale et la sécurité de connexion ne sont pas oubliées (PAM, SELinux…).

Du fait qu'un vaste panel de méthodes est étudié ainsi que des concepts fondamentaux (cryptologie, pare-feu, techniques d'attaques et de sécurisation), l'ouvrage dispense aussi la culture générale autour de la sécurité que doit posséder tout administrateur système Linux ou tout administrateur d'application fonctionnant sous Linux. Cette culture lui permet de dialoguer sereinement avec le responsable de la sécurité globale de l'entreprise. Elle l'autorise également à assurer cette fonction de responsable dans une petite structure.

Il existe de nombreux ouvrages sur la sécurisation de Linux, en quoi ce livre est-il original ?

D'abord, il se veut manuel de formation. À ce titre, chaque module est divisé en deux parties : une partie « cours » et une partie « ateliers ». La partie « cours » se divise elle-même en théorie et savoir pratique (commandes, fichiers…). Les « ateliers » ne sont pas une accumulation d'exercices mais plutôt une séquence cohérente d'actions que le lecteur doit effectuer. Non seulement ils illustrent le cours, mais ils représentent un savoir concret en ce sens que la plupart des ateliers peuvent être considérés comme des « recettes pratiques d'administration ». Les ateliers sont regroupés en « tâches ». Le lecteur n'est pas obligé de les réaliser toutes. Il doit privilégier celles qui correspondent à des concepts fondateurs ou à des sujets qui répondent à un besoin immédiat.

Volontairement, ce livre privilégie le mode commande. Le système Windows a habitué l'utilisateur et l'administrateur à tout résoudre par des clics dans un environnement graphique. Ce mode existe sous Linux, mais n'est pas celui utilisé par l'expert. Le mode commande en mode texte est plébiscité par l'ensemble des administrateurs Linux. Pourquoi ? Tout simplement parce qu'il est plus puissant, intemporel et même, à l'usage, plus simple. Ce mode permet l'administration complète d'un système Linux à distance avec une liaison

inférieure à 9 600 bauds (débit pris en charge par un téléphone portable) ! Le mode commande est primordial dans l'approche automatisée de l'administration grâce à l'écriture de scripts shell. Il permet également une administration indépendante des distributions.

Ce livre ne se limite pas à une distribution particulière. Certes, pour les ateliers, il a bien fallu en choisir une. Nous avons opté pour CentOS, qui est un clone de la distribution RedHat, la plus utilisée dans les entreprises. Elle est binaire compatible avec elle. Dans les parties « cours », la rubrique « *Les particularités des distributions* » indique les commandes, les fichiers ou les aspects propres à une distribution particulière. Il a fallu faire un choix : seules les distributions RedHat, Debian et Ubuntu sont mentionnées.

Ce livre se veut le plus intemporel possible. Si de nouvelles commandes apparaissent avant une prochaine édition de cet ouvrage, le lecteur trouvera sur le site www.tsoft.fr (cf. plus loin) de nouvelles rubriques sur ces sujets.

Public

Le public visé par ce livre est d'abord les administrateurs de serveurs Linux. Du fait que les ateliers forment une sorte de recueil de « recettes pratiques d'administration et de sécurisation », il peut être lu avec profit par tout administrateur, développeur ou utilisateur d'un système Linux.

Support de formation

Ce support convient à des formations sur la sécurisation d'un système Linux d'une durée comprise entre deux et six jours. L'idéal est de cinq jours. La durée peut être écourtée ou allongée en fonction des modules et ateliers traités ainsi qu'en fonction du niveau des participants.

L'éditeur Tsoft (www.tsoft.fr) peut fournir aux organismes de formation et aux formateurs des « diapositives instructeur » complémentaires destinés à aider le personnel enseignant.

Guide d'autoformation

Ce livre peut être également utilisé en tant que support d'autoformation. L'élève doit disposer d'un ordinateur qui sera dédié à Linux (on le reformate complètement). Plusieurs modules, dont Kerberos, nécessitent l'utilisation de deux serveurs. Il est possible d'utiliser des serveurs virtuels sous Windows ou Linux.

Certifications

La certification LPI (Linux Professional Institute), indépendante des distributions, est prise en charge, parmi d'autres, par SuSe et IBM. L'ouvrage est une bonne préparation aux deux premiers niveaux du programme LPIC. Nous invitons les lecteurs à se renseigner auprès du LPI : http://www.lpi.org.

Un livre dynamique grâce à Internet

Le noyau Linux, les distributions Linux vont peut-être évoluer plus rapidement que cet ouvrage. Les sites www.tsoft.fr et www.editions-eyrolles.com proposeront sur la page de présentation du présent ouvrage des liens dédiés à des compléments sur ces évolutions.

- sur le site www.tsoft.fr, dans la zone <Recherche> saisissez TS0101 et validez par <Entrée>, puis cliquez sur le lien vers la page de l'ouvrage ;
- sur le site www.editions-eyrolles.com, dans la zone <Recherche> saisissez `G13462` et validez par <Entrée>.

Table des matières

Quelques conventions de notation

Dans les parties théoriques

Dans les pages de ce livre, les commandes du système et les fichiers de paramètres sont expliqués en détail avec le contexte de leur utilisation.

Les noms des commandes sont imprimés en police Courier, par exemple :

`configfile` Charge un fichier de configuration.

Les noms des fichiers sont imprimés en police Arial, par exemple :

/boot/grub/grub.conf Le fichier de configuration de GRUB.

Dans les ateliers

Ce livre fait une très large part aux ateliers pratiques à réaliser par le lecteur. Les commandes à passer et les résultats obtenus sont imprimés en police Courier sur un arrière-plan grisé. Les commandes mises en gras sont celles que vous devez taper au clavier, les autres informations sont celles qui sont affichées par le système.

```
[root@linux1 ~]# mount -t tmpfs tmpfs /mnt/disk
[root@linux1 ~]# free
             total       used       free     shared    buffers     cached
Mem:        449152     200580     248572          0      25832     144988
-/+ buffers/cache:      29760     419392
Swap:       522104          0     522104
[root@linux1 ~]# df -Th /mnt/disk
Filesystem    Type    Size  Used Avail Use% Mounted on
tmpfs         tmpfs    220M     0  220M   0% /mnt/disk
[root@linux1 ~]# ls -ld /mnt/disk
drwxrwxrwt  2 root root 40 Jan  5 21:45 /mnt/disk
```

Dans certains exercices, des lignes sont mises en italique, ce sont des lignes qu'il vous aura été demandé d'ajouter ou de modifier dans certains fichiers au cours des ateliers.

```
 [root@linux1 grub]# vi grub.conf
default=1
#timeout=5
title CentOS-4 i386 (2.6.9-42.EL)
        root (hd0,0)
        kernel /vmlinuz-2.6.9-42.EL ro root=LABEL=/1 rhgb quiet
        initrd /initrd-2.6.9-42.EL.img
title New Kernel (2.6.17)
        root (hd0,0)
        kernel /linux-2.6.17 ro root=LABEL=/1 rhgb
        initrd /initrd-2.6.17.img
title CentOS-4 i386 single
        root (hd0,0)
        kernel /vmlinuz-2.6.9-42.EL ro root=LABEL=/1 rhgb single
        initrd /initrd-2.6.9-42.EL.img
[root@linux1 grub]# cd
[root@linux1 ~]#
```

Indications bibliographiques

Les livres sont mentionnés par leur titre et le nom d'un auteur. Le lecteur pourra trouver aisément le nom de l'éditeur et la date de parution sur le site www.amazon.fr.

Progression pédagogique

Introduction

Ce chapitre présente la sécurité informatique, les principales attaques ainsi que les logiciels malveillants. Il donne des conseils pour établir une politique de sécurité et pour réagir en cas d'incident.

La cryptologie

Ce chapitre présente la cryptologie. Cette science a pour but de dissimuler la signification des messages échangés entre deux entités. Les concepts et le vocabulaire de la cryptologie sont étudiés. À la fin du chapitre le lecteur sait utiliser des logiciels cryptographiques pour protéger ses fichiers ou ses systèmes de fichiers ainsi que ses e-mails.

La sécurité locale

Ce chapitre présente la sécurité locale. Ses concepts de base sont rappelés : la sécurité d'un système multi-utilisateur, les droits, la sécurité de connexion. On présente également les capacités (« capabilities » en anglais) du noyau et la commande `sudo`.

PAM

Ce chapitre présente PAM ou la sécurité de connexion en plug-in. La plupart des logiciels d'authentification utilisent cette technologie. On présente les concepts de PAM, la syntaxe de sa configuration et les principaux modules PAM.

SELinux

Le système SELinux, créé par la NSA apporte une nouvelle dimension à la sécurité Linux. Ce chapitre présente cette approche ainsi que sa mise en œuvre et le dépannage.

SSH

Ce chapitre étudie la technologie SSH. Notamment le fonctionnement du protocole cryptographique de même nom. De manière plus concrète, le lecteur apprend à utiliser les commandes SSH de connexion distante, d'exécution de commandes à distance et de transfert de fichiers. Enfin sont étudiés des éléments complémentaires comme l'utilisation d'agent.

PKI et SSL

Ce chapitre présente l'approche PKI. Concrètement le lecteur apprend à créer des certificats et à instaurer une PKI interne. Il apprend également à dépanner des liaisons SSL ou des échanges S/MIME. Enfin, il étudie le logiciel Stunnel qui permet de créer des tunnels SSL.

Kerberos

Ce chapitre présente la sécurité Kerberos. Elle apporte la sécurité des transactions réseaux et l'authentification unique (SSO). Après avoir étudié le protocole, le lecteur apprend concrètement comment mettre en œuvre Kerberos. Il apprend d'abord son installation, son administration et son utilisation. Les deux versions MIT et Heimdal sont étudiées.

Les pare-feu

Ce chapitre présente les pare-feu. Il commence par étudier les différents types de pare-feu et les concepts associés. Concrètement, le lecteur apprend à configurer un pare-feu local avec Iptables. Il étudie aussi les logiciels pare-feu Squid et Tcpd.

VPN

Ce chapitre présente les VPN, c'est-à-dire les réseaux virtuels privés. Il commence par présenter les différentes techniques et ensuite le lecteur apprend concrètement à mettre en place un VPN avec le logiciel OpenVPN puis un VPN IPSec.

Sécurisation des applications

Après avoir donné des conseils généraux sur la sécurisation des applications, on présente les particularités du Web, du DNS, des bases de données et de l'e-mail. Le logiciel Apache est traité plus en détail.

Audit

Ce chapitre présente les logiciels d'audit. Le lecteur, après avoir pris connaissance de l'ensemble de ces outils, étudie les plus connus : Tcpdump, Wireshark, Nmap, Nessus, Aide, Tripwire et Snort. Ce dernier, pourtant abondamment décrit dans ce module, est surtout présenté dans un contexte local. L'apprentissage du NIDS Snort ferait aisément l'objet d'un ouvrage complet.

Sécuriser un serveur

Ce chapitre présente comment sécuriser un serveur Linux. Après avoir fait une revue de détail des techniques, le lecteur apprend à sécuriser un serveur RedHat dès son installation. La gestion des journaux de bord est également traitée.

- *CIA*
- *Exploit, DOS*
- *Virus, Trojan*
- *SANS*
- *CERT*

1

Introduction

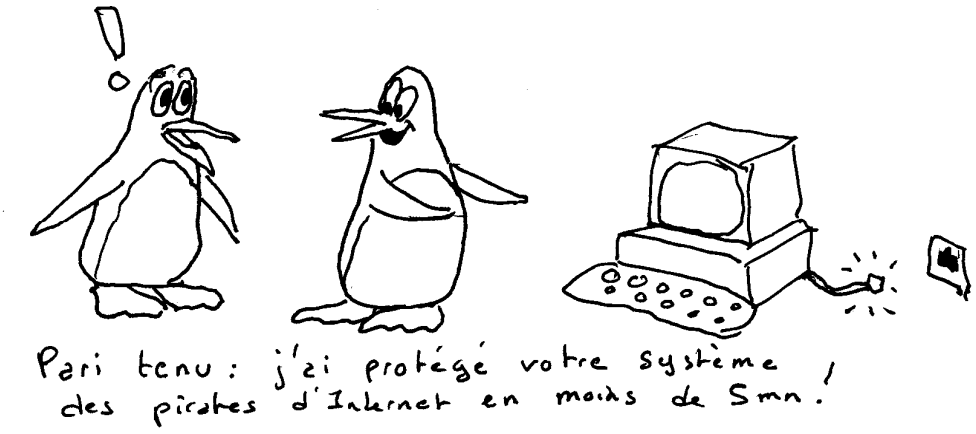
Objectifs

Ce chapitre présente la sécurité informatique, les principales attaques ainsi que les logiciels malveillants. Il donne des… conseils pour établir une politique de sécurité et pour réagir en cas d'incident.

Contenu

La sécurité informatique.
Les politiques de sécurité.

Attaques et logiciels malveillants.
Conduite à tenir en cas d'incident.



La sécurité informatique

La théorie

Les objectifs de la sécurité informatique

Concrètement, je ne veux pas que l'on détruise mes fichiers informatiques, par exemple, mes fichiers comptables. Je ne souhaite pas que mes concurrents puissent lire la liste de mes clients. Il me serait désagréable d'avoir mon site Web inaccessible ou mes ordinateurs fréquemment inopérants. Enfin l'image de marque de ma société serait entachée si mon système informatique servait de base d'attaque d'autres sites.

En résumez, je désire :

- La confidentialité de mes données.
- L'intégrité de mes données.
- La continuité de service.

En anglais, le terme CIA (Confidentiality, Integrity and Availability) en est le sigle.

Les principes de la sécurité

La sécurité, qu'elle s'applique à l'espionnage ou à l'informatique, repose sur les mêmes principes :

- Le secret. Un espion ne révèle à personne, y compris à sa famille, la nature de ses activités.
- La compartimentation ou l'isolation. Les espions sont répartis en réseaux étanches qui s'ignorent mutuellement. Si un espion est pris, il ne peut livrer que les membres de son propre réseau.
- L'authentification. Des espions qui se rencontrent utilisent des mots de passe pour se faire connaître de leur contact. Des détails anatomiques (des cicatrices…) peuvent également les identifier.
- L'accréditation et la classification des documents. Des documents sont classés « Top Secret », « Secret » ou « Confidential ». Seules les personnes accréditées auront accès à ces documents.
- Plusieurs lignes de défense. Dans une base militaire ultra secrète, il y a plusieurs enceintes et au passage de chacune d'elles, des gardes vérifient votre identité et vos accréditations d'accès.
- La minimisation. Moins il y a de personnes qui connaissent les membres d'un groupe d'espions, meilleure en est sa sécurité. Moins il y a de portes d'entrée dans une base secrète, plus elle est facile à garder.
- La surveillance. Des patrouilles circulent en permanence dans une base secrète. Des enquêtes de routine vérifient la qualité des mesures de sécurité et la présence éventuelle de « taupes ».
- La désinformation. Durant la dernière guerre, l'opération « fortitude » a fait croire à Hitler que le débarquement aurait lieu dans le Pas-de-Calais.
- La formation, l'information, les sanctions. Un espion ou le personnel d'une base secrète reçoit une formation en sécurité. Des avis rappellent les principales mesures à suivre. Enfin des sanctions sont prévues pour les contrevenants.

Les moyens de la sécurité informatique

Pour assurer les objectifs de la sécurité informatique, plusieurs techniques sont utilisées :

- La cryptologie. Cette science a pour objectif principal de cacher des informations, notamment celles transitant sur les réseaux ou stockées dans des fichiers (comme les mots de passe). La cryptologie est également utilisée dans la vérification de l'intégrité des données ou dans l'authentification des individus ou des services.
- La sécurité physique. Par exemple, il faut enfermer ses serveurs à clé. C'est une mesure primordiale de sécurité.
- Les systèmes d'authentification. Actuellement, la connaissance d'un mot de passe est le moyen d'identification informatique le plus usuel. Les systèmes biométriques (reconnaissance de l'iris, du lobe de l'oreille…) sont de plus en plus utilisés.
- Les droits. Les droits associés aux fichiers limitent leur accès à telle ou telle personne ou tel groupe d'individus. Ce système reste la base de sécurisation des ordinateurs sous Unix/Linux.
- Les sommes de contrôle. Elles vérifient l'intégrité des fichiers. La présence de logiciels malveillant (virus, rootkit…) est principalement mise en évidence par cette technique.
- Les pare-feu. Un pare-feu filtre les paquets réseaux. C'est un outil de protection indispensable.
- Les sauvegardes. Suite à une attaque, on peut perdre des données vitales. La sauvegarde régulière des données est le seul recours.
- L'audit des principaux événements (connexion réussie, échouée…) du système. Il garantit le suivi des règles de sécurité et détecte les tentatives d'intrusion.

Les différents secteurs de la sécurité

La sécurité, dans une société, se divise en plusieurs secteurs :

- La sécurité globale. Elle s'occupe de la sécurité des locaux, des documents papier, du personnel dirigeant. Par ses objectifs et ses moyens, elle est très proche de la sécurité militaire.
- La sécurité réseau. Le responsable de la sécurité réseau est le plus souvent le responsable de la sécurité informatique tout court. En effet, le réseau est le cœur du système d'information d'une entreprise. Son service est responsable des pare-feu, des techniques anti-intrusion réseau, et du choix global des stratégies de sécurisation.
- La sécurité des serveurs. Les administrateurs doivent connaître les techniques de sécurisation offertes par les systèmes d'exploitation et par les applications pour mettre en œuvre les politiques édictées par le responsable de la sécurité.

  Remarque
  l'objectif principal de cet ouvrage est justement la sécurisation de serveurs sous Linux.

- La sécurité des applications. Les responsables d'applications doivent non seulement mettre en place et gérer leur application, mais aussi veiller à leur sécurisation. Ce sont eux notamment qui donnent les accréditations aux utilisateurs.
- La sécurité des postes de travail. Cette tâche est complexe. Elle est simplifiée par l'usage de postes uniquement Web ou de protocoles comme Citrix qui transforment le poste de travail en simple terminal.

Pour en savoir plus

Howto

Security-Howto

Security-QuickStart-Howto

Security-QuickStart-Redhat-Howto

Journaux

MISC – journal français, bimestriel, sur la sécurité informatique (www.miscmag.com)

Internet

Wikipedia - La sécurité informatique
http://fr.wikipedia.org/wiki/Portail:Sécurité_informatique
http://en.wikipedia.org/wiki/Computer_insecurity

CERT
http://www.cert.org

NIST : Computer Security Resource Center
http://csrc.nist.gov/

Linux-sec
http://www.linux-sec.net/

Manuel de sécurisation de Debian
http://www.debian.org/doc/manuals/securing-debian-howto/

RedHat Enterprise Linux 6 (RHEL-6) – Security Guide
http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide/Red_Hat_Enterprise_Linux-6-Security_Guide-en-US.pdf

Livres

Practical Unix & Internet Security, de S. Garfinkel (2003).

Computer Security Basics Computer Security Basics, par Rick Lehtinen, G.T. Gangemi, Sr (2006).

Computer Security: Principles and Practice, de William Stallings et L. Brown (2011)

Attaques et logiciels malveillants

La théorie

Les pirates informatiques ou « hackers » (Harrap's dixit), utilisent différents types d'attaques et de logiciels malveillants, en voici les principaux :

Les attaques

Exploit

Un *exploit* est un programme qui exploite une faille d'un logiciel. Au minimum, un exploit entraîne un déni de service. Dans le pire des cas il permet au pirate de prendre le contrôle du logiciel ou du système.

Voici les principales techniques utilisées par les *exploits* :

- Le dépassement de tampons (Buffer Overflow)
- L'injection de code (Code Injection)
- L'injection SQL

L'usurpation d'adresse IP (IP Spoofing)

L'attaque par usurpation d'adresse IP consiste à générer des paquets IP qui semblent pour la cible provenir d'un poste accrédité.

Session Hijacking

C'est une forme particulière d'IP Spoofing. Le pirate se substitue à un poste après sa phase d'authentification.

Replay

Ce type d'attaque consiste à enregistrer des informations d'authentification et à les réutiliser ultérieurement lors d'une autre session.

L'attaque de l'homme du milieu (Man-in-the-Middle)

C'est une forme particulière d'IP Spoofing. Le pirate joue l'intermédiaire entre un client et un serveur. Il est le serveur vis-à-vis du client et il est le client vis-à-vis du serveur. Il peut ainsi récolter des informations confidentielles étant donné qu'il a accès à tout le dialogue des correspondants.

Déni de service (DoS : Denial Of Service)

Cette attaque consiste à rendre inutilisable le système cible. Soit en le surchargeant de requêtes, soit en le faisant « planter ».

Phishing

Le phishing consiste à acquérir de manière frauduleuse (via une réponse à un e-mail) des informations sensibles (mot de passe, adresse IP…). C'est un exemple de « Social Engineering ».

Pharming

Le pharming consiste à rediriger le trafic d'un serveur vers un autre dont on a le contrôle.

L'attaque des mots de passe

Un des points faibles des protections informatiques est l'usage de mots de passe. Deux techniques permettent de les trouver plus ou moins rapidement :

- L'attaque au dictionnaire : on compare le mot de passe chiffré au chiffrement d'une base préétablie (appelée dictionnaire) de mots de passe.

- L'attaque exhaustive : on essaye de chiffrer toutes les combinaisons possibles des caractères entrant dans la composition d'un mot de passe. Cette attaque n'est possible que si les mots de passe sont courts (inférieurs à huit caractères).

Les logiciels malveillants (malicious code or Malware)

Virus

Un virus est un morceau de code inséré dans un logiciel de telle façon que quand le code du logiciel est exécuté, celui du virus l'est également. Le virus essaye également d'infecter d'autres logiciels sains. Le code inséré (le virus proprement dit) nuit généralement au fonctionnement du système.

Remarque
Le terme virus est souvent pris comme synonyme de logiciel malveillant.

Ver (Worm)

Un ver est un logiciel malveillant qui s'auto réplique, le plus souvent en envoyant des copies de lui-même sur d'autres ordinateurs. Contrairement à un virus, il n'est pas associé à un autre logiciel.

Cheval de Troie (Trojan Horse)

Un cheval de Troie est un morceau de code introduit dans un autre logiciel et qui effectue des actions malignes dans certaines circonstances. À la différence d'un virus, il ne peut se reproduire par lui-même. C'est la recopie du logiciel hôte qui assure sa reproduction.

Logiciel espion (Spyware)

Un logiciel espion est un logiciel qui s'installe sur un système hôte dans le but de collecter des informations. Ces informations sont le plus souvent récoltées par réseau.

Rootkit

Un rootkit est un ensemble de commandes qui se substitue aux commandes systèmes (ps, netstat…). Un rootkit est utilisable à partir d'une porte dérobée après qu'une intrusion a réussi. L'objectif d'un rootkit est le maintien du pirate sur le système sans que l'on puisse détecter sa présence.

Porte dérobée (BackDoor)

Une porte dérobée est un accès secret à un logiciel ou au système. Seul le pirate en a la connaissance et l'usage. C'est une sorte de cheval de Troie.

Pour en savoir plus

Internet

Portail : Sécurité informatique
http://fr.wikipedia.org/wiki/Portail:Sécurité_informatique

Notes d'informations du Certa
http://www.certa.ssi.gouv.fr/site/index2.html

Wikipedia: Logiciel malveillant
http://fr.wikipedia.org/wiki/Logiciel_malveillant

Livre

Hacking Exposed par Stuart McClure & all (2009).

Les politiques de sécurité

La théorie

Comme dans tout problème informatique, les actions entreprises doivent être le résultat d'une démarche logique et méthodique, en l'occurrence une politique de sécurité. Celle-ci permet d'agir et de faire des choix cohérents en répondant à un cahier des charges précis.

Établir une politique de sécurité

1re étape : Identifier ce qu'il faut protéger

La première étape est de bien identifier ce qu'il faut protéger : quels réseaux, ordinateurs, données. Ceci passe éventuellement par une classification des documents informatiques en « Très secret », « Secret », « Confidentiel », « Public ». Certaines des données publiques doivent éventuellement être protégées : soit de leur destruction, soit de leur altération. Les postes seront classés en Bastions (serveurs devant être hyper protégés), postes sensibles ou ordinateurs à accès libre.

2e étape : Analyser les risques

On commence par analyser l'existant et à mettre en évidence les risques potentiels. C'est ce que l'on appelle réaliser un audit de survol. Pour évaluer les risques, on se base sur des scénarios qui sont construits à partir des attaques classiques.

Pour apporter la preuve de ces risques, on peut réaliser des essais d'intrusion.

3e étape : Pondérer les risques

La plupart des sociétés qui utilisent l'informatique sont globalement confrontées aux mêmes risques. Mais leur taille, les conditions d'exploitation ou leur métier font qu'un même risque n'a pas la même importance. Par exemple, le risque d'avoir un employé malhonnête dans une entreprise de vingt personnes est beaucoup plus faible que dans une multinationale. Il faut donc pondérer les risques.

Risque = impact * probabilité

4e étape : Évaluer les contraintes

Pour assurer la sécurité on ne peut pas se permettre de faire n'importe quoi. Il faut tenir compte de plusieurs contraintes. Les principales sont souvent l'existant (matériels, logiciels, personnel, locaux…), le budget et le temps. Par exemple, il n'est pas possible de mettre au panier la plupart des logiciels sous prétexte qu'ils ne sont pas sûrs.

Il est très important d'établir un budget de la sécurité et de mettre en regard la protection pondérée qu'elle apporte. C'est-à-dire les pertes (en euros) qu'elle évite. C'est le même raisonnement que l'on suit quand on s'assure.

5e étape : Choisir les moyens

Il existe plusieurs techniques de base pour assurer la sécurité. Il faut faire un choix cohérent de ces techniques. On peut les diviser en deux catégories :

- Les moyens de sécurisation : ils constituent évidemment l'élément clé de la sécurité. Citons en vrac : les techniques cryptographiques, la sécurisation des serveurs et des postes de travail, l'authentification des utilisateurs et des applications, les raccordements aux réseaux locaux ou distants, les routeurs, les pare-feu, les VPN, les antivirus, les politiques d'acquisition de matériels et de logiciels, les politiques d'embauche et de classification...
- Les moyens de détection d'intrusion.

Enfin, il faut établir la conduite à tenir en cas d'incident.

6e étape : Adopter la politique de sécurité

Il faut enfin adopter, après amendements, une politique. Une mauvaise politique sera toujours préférable à pas de politique du tout. Et il faut bien sûr nommer un responsable, le monsieur « sécurité ». Dans une structure modeste, c'est le plus souvent le responsable réseau qui sera choisi.

7e étape : Tester

Après avoir adopté et mis en place une politique de sécurité, il faut s'assurer par un audit externe si celle-ci répond au cahier des charges.

L'élément clé de cet audit consiste, comme dans la deuxième étape, à réaliser des intrusions et à démontrer la résistance du système face à des scénarios de risques.

Remarque
L'établissement d'une politique de sécurité n'est pas une fin en soi ni une action définitive. Pour qu'elle soit efficace, l'établissement de cette politique doit s'effectuer de manière itérative. Établir une politique de sécurité, c'est la base pour une nouvelle analyse et pour une meilleure maîtrise des problèmes de sécurité.

Pour en savoir plus

RFC

Security Policy (rfc 2196)

Internet

Wikipedia – Méthode d'analyse de risques informatiques
http://fr.wikipedia.org/wiki/Méthode_d'analyse_de_risques_informatiques_optimisée

http://fr.wikipedia.org/wiki/Risques_en_sécurité_informatique

The SANS Security Policy Project
https://www2.sans.org/resources/policies/

Introduction to Security Policies, Part One: An Overview of Policies
http://www.securityfocus.com/infocus/1193

Livres

Information Security Policies Made Easy, Version 10 by Charles Cresson Wood (2005).

Management de la sécurité de l'information - Implémentation ISO 27001 - Mise en place d'un SMSI et audit de certification, par Alexandre Fernandez-Toro (2007)

Tableaux de bord de la sécurité réseau, par Cédric Llorens & all (2010).

Information Security: Policy, Processes, and Practices de Detmar W. Straub & all (2008)

Conduite à tenir en cas d'incident

La théorie

Supposons que vous découvrez une intrusion, que faire ?

1. Évaluer l'importance de l'incident et sa nature. Rassembler l'équipe anti-intrusion. Évaluer la solution de débrancher l'ordinateur du réseau.
2. Constituer un dossier d'analyse (qui peut servir ultérieurement comme dossier de preuves) et diagnostiquer le problème principalement en étudiant les journaux de bord. L'idéal est de faire une copie ou une sauvegarde du système infecté (ce qui est très facile s'il fonctionne dans une machine virtuelle).
3. Avertir éventuellement le CERT, les autorités, la direction, les utilisateurs.
4. Réparer les systèmes, reprendre l'activité.
5. Évaluer les dommages et les coûts de l'incident. Documenter l'incident.
6. Vérifier s'il n'est pas nécessaire de modifier les stratégies et les moyens de sécurité.

Les erreurs à ne pas commettre

1. Paniquer.
2. Ne pas avoir de plan de réponse aux incidents.
3. Résoudre le problème sans essayer de le comprendre.

Prévoyance

1. Constituer une équipe de réponse aux incidents (un « CERT » : Computer Emergency Response Team).
2. Rassembler les informations nécessaires (coordonnées de votre FAI (ISP), du correspondant du CERT de votre pays, en France c'est le CERTA…).
3. Prévoir des checklists pour diagnostiquer les problèmes.
4. Prévoir un plan de reprise d'activité (sauvegarde…).

Pour en savoir plus

Internet

Répondre aux incidents de sécurité informatique
http://www.microsoft.com/france/technet/securite/responding_sec_incidents.mspx

Les bons réflexes en cas d'intrusion sur un système d'information
http://www.certa.ssi.gouv.fr/site/CERTA-2002-INF-002.pdf

CERTA
Centre d'Expertise Gouvernemental de Réponse et de Traitement des Attaques informatiques
http://www.certa.ssi.gouv.fr/certa/cert.html

CERT : Steps for Recovering from a UNIX or NT System Compromise
http://www.cert.org/tech_tips/root_compromise.html

CERT : Incident Reporting Guidelines
http://www.cert.org/tech_tips/incident_reporting.html

- *AES, 3DES, RC4*
- *ECB, CBC, OFB*
- *RSA, DSA, DSS*
- *MD5, SHA1*
- *GPG, PGP*

2

La cryptologie

Objectifs

Ce chapitre présente la cryptologie. Cette science a pour but de dissimuler la signification des messages échangés entre deux entités. Les concepts et le vocabulaire de la cryptologie sont étudiés. À la fin du chapitre on sait utiliser des logiciels cryptographiques pour protéger ses fichiers ou ses systèmes de fichiers ainsi que ses e-mails.

Contenu

La cryptologie.

Les algorithmes de chiffrement symétrique.

Les fonctions de hachage.

Les algorithmes de chiffrement à clé publique.

La signature numérique.

Les protocoles cryptographiques.

Le protocole OpenPGP.

Le chiffrement des systèmes de fichiers.

Ateliers.

La cryptologie

La théorie

Le but principal de la cryptologie (*cryptology*) moderne est de résoudre des problèmes de sécurité associés à la communication en réseau.

Les deux principaux problèmes sont :

- L'échange confidentiel d'informations.
- L'authentification des participants.

Les techniques de base de la cryptologie

La base de la cryptologie est constituée d'algorithmes de chiffrement, appelés également méthodes cryptographiques. Ces algorithmes ont pour rôle de rendre incompréhensible un message pour toute personne autre que son destinataire.

Les algorithmes de chiffrement sont de deux types :

- Symétriques, à clé secrète.
- Asymétriques, à clé publique.

Pour résoudre les problèmes cryptographiques, les algorithmes de chiffrement ne suffisent pas. On utilise d'autres techniques, les deux principales sont :

- Les générateurs d'empreintes.
- Les générateurs de nombres aléatoires ou pseudo aléatoires.

Pour en savoir plus

Internet

Wikipedia - Cryptography (vocabulaire, histoire, concepts modernes, bibliographie…)
http://en.wikipedia.org/wiki/Cryptography

Wikipedia - History of cryptography
http://en.wikipedia.org/wiki/History_of_cryptography

Petite histoire de la cryptographie
http://www.apprendre-en-ligne.net/crypto/histoire/index.html

Livres

Histoire des codes secrets, par S. Singh (2009)

Practical Cryptography, par Niels Ferguson et Bruce Schneier (2003)

Les algorithmes de chiffrement symétrique

La théorie

Le principe d'utilisation d'une méthode (algorithme) de chiffrement symétrique est très simple : un message en clair est chiffré (crypté) en utilisant une méthode cryptographique avant d'être transmis au correspondant. Le chiffrement (cryptage) est réalisé par un logiciel ou par du matériel. Le message chiffré peut être intercepté par un tiers sans danger, il est incompréhensible.

L'algorithme n'a pas besoin d'être secret. Toute la sécurité repose sur le maintien secret d'une clé qui constitue l'élément variable dans le processus de chiffrement.

Un algorithme symétrique ou à clé secrète n'utilise qu'une seule clé qui ne doit être connue que des correspondants. Elle sert aussi bien au chiffrement qu'au déchiffrement.

Exemple d'algorithme : la méthode de Jules César

Cette méthode, de type symétrique (à clé secrète), est utilisée par les écoliers. Elle n'offre aucune sécurité, mais elle permet d'illustrer le fonctionnement d'une méthode cryptographique. On l'appelle méthode de « Jules César », car ce célèbre dictateur romain l'utilisait pour échanger des messages avec Cicéron (cf. l'ouvrage *De Bellum Gallicum*).

Un message en clair, par exemple BONJOUR, est chiffré lettre par lettre. Chaque lettre est décalée dans l'ordre alphabétique d'un nombre de lettres convenu d'avance. Ce nombre constitue la clé secrète qui ne doit être connue que des correspondants. En prenant trois comme décalage, le message en clair BONJOUR devient le cryptogramme ERQMRXU. Le cryptogramme peut circuler sur le réseau, sans danger, car une personne qui intercepte le message ne peut le déchiffrer, ne possédant pas la clé. Le correspondant effectue l'opération inverse. Il décale de trois lettres (mais dans le sens inverse) chaque lettre du cryptogramme pour obtenir le message d'origine.

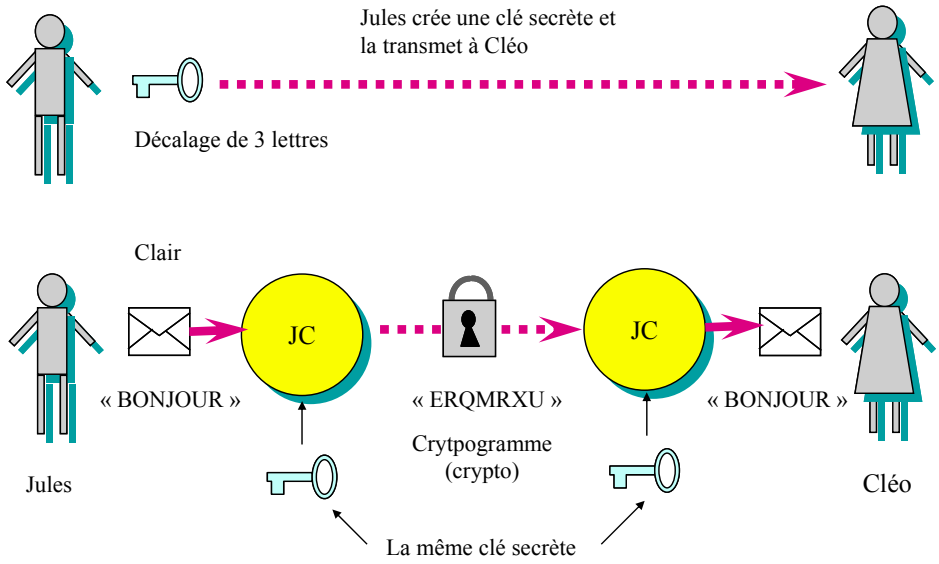


Fig. Le chiffrement de Jules César

Le problème de l'échange des clés

Les algorithmes de chiffrement symétriques peuvent assurer la confidentialité des échanges. La principale difficulté de leur utilisation réside dans le fait que les deux participants doivent partager secrètement une même clé.

Une solution à ce problème est d'utiliser un algorithme de chiffrement asymétrique pour échanger la clé secrète.

Panorama des méthodes

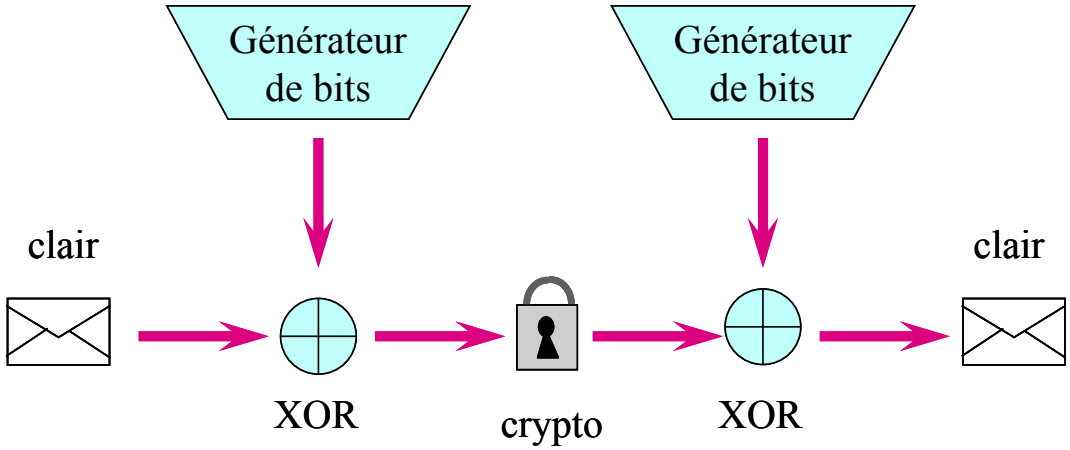


Fig. Méthode de flux (stream cipher)

Les méthodes à clés privées se divisent en deux types :

- Les méthodes codant un flux de bits (Rc4 ou Arcfour...).
- Les méthodes de codage de blocs (AES, 3DES, Blowfish, IDEA, Cast5...).

Remarque
L'AES (Advanced Encryption Standard), basé sur l'algorithme de Rijndael, est maintenant le nouveau standard de chiffrement. Il remplace l'ancien standard le Triple DES (3DES), lui-même remplaçant le DES (Data Encryption Standard).

Codage par flux

Un générateur pseudo aléatoire génère une suite de bits qui est combinée bit à bit avec le texte clair pour former le cryptogramme. La même suite aléatoire doit être utilisée pour reconstituer le clair. Comme on utilise l'opération XOR (ou exclusif) pour combiner les bits, les opérations de chiffrement et déchiffrement sont identiques.

Codage par blocs

Les ordinateurs utilisent des registres ou des composants matériels pour réaliser le chiffrement. Ce sont donc en fait des blocs de données qui sont en réalité chiffrés (ou déchiffrés) et les méthodes à clés secrètes ont donc été majoritairement conçues en ce sens.

Les modes d'utilisation des méthodes blocs

Dans les méthodes de codage de blocs, le message clair est divisé en blocs uniformes de bits, par exemple 64 bits, le message crypté est constitué également de blocs de même taille. Il existe plusieurs modes d'utilisation des méthodes blocs :

ECB

Dans la méthode ECB (Electronic Code Block), chaque bloc est toujours codé avec la même clé. Par conséquent, deux blocs identiques seront codés de la même manière. Cette méthode, la moins sûre, est utilisée notamment pour chiffrer un fichier en accès aléatoire.

CBC, Initial Vector (IV) et Padding

Dans la méthode CBC (Cipher Bloc Chaining), chaque bloc de texte clair est combiné par un « ou exclusif » avec le bloc chiffré précédent avant d'être chiffré. Pour le premier bloc, on

utilise un bloc dit vecteur d'initialisation ou IV (Initial Vector), choisi aléatoirement et qui est inscrit en clair dans le message crypté. Cette méthode est sans doute la plus sûre d'un point de vue cryptographique, mais une erreur de transmission interdit le déchiffrement du reste du message. D'autre part, si le dernier bloc n'est pas complet, il faut le remplir (PADDING). En outre, les opérations de chiffrement et de déchiffrement sont différentes. C'est souvent la méthode par défaut.

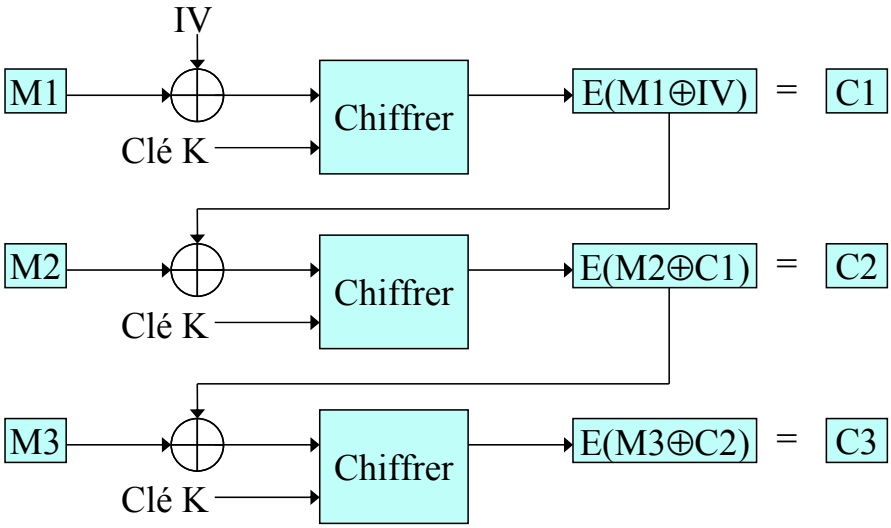


Fig. La méthode CBC (chiffrement)

CFB

Dans la méthode CFB (Cipher Feed Back), la clé change à chaque bloc. Elle permet un cryptage identique aux méthodes à codage de flux de bits. Comme dans celui-ci, un octet du texte chiffré est produit par un ou exclusif entre un octet de la clé et un octet du texte clair. Chaque octet du texte chiffré sert également à générer le flux de la clé que ce soit lors du chiffrement ou du déchiffrement. Hélas, comme dans le CBC, il y a propagation des erreurs.

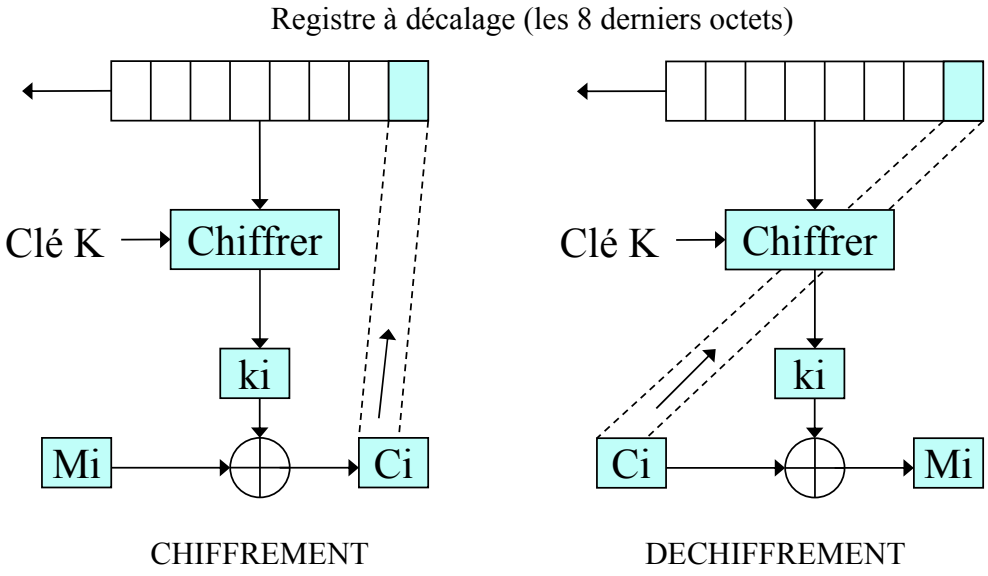


Fig. La méthode CFB

OFB

La méthode OFB (Ouput FeedBack) est similaire à la méthode CFB, mais le chiffrement et le déchiffrement sont identiques : on n'utilise pas une méthode pour crypter et une autre pour

décrypter. Chaque fois qu'un octet du flux de la clé est généré, il est transmis dans le registre à décalage qui sert à générer cette même clé. Le flux du texte chiffré est produit par un ou exclusif entre le flux de la clé et le flux du texte clair. Lors du déchiffrement, c'est l'inverse : Le flux du texte clair est produit par un ou exclusif entre le flux de la clé et le flux du texte chiffré. Cette méthode émule complètement une méthode par flux : pas de propagation des erreurs, pas besoin de PADDING, chiffrement et déchiffrement sont identiques.

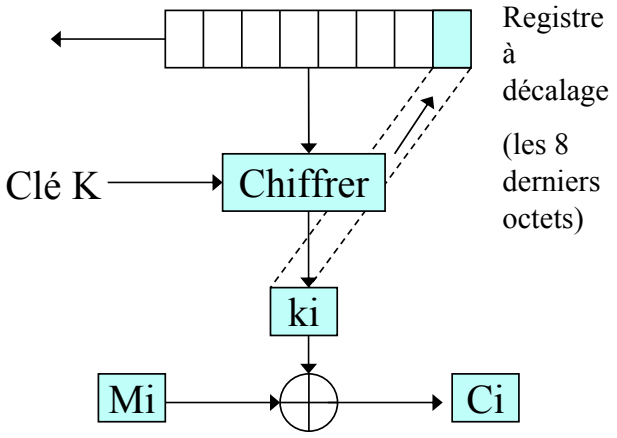


Fig. La méthode OFB

L'utilisation de graine

Une graine (*salt*) est un nombre aléatoire qui intervient, en plus d'un mot de passe, pour générer la clé. Si le mot de passe correspond à un mot du dictionnaire et donc ne correspond pas à une suite aléatoire de lettres, une graine doit être utilisée. En effet, si on dérive mécaniquement la clé du mot de passe, le même mot de passe entraîne alors la même clé. Bien sûr, pour pouvoir déchiffrer, la graine est ajoutée en clair en tête du message chiffré.

Le codage Base64

La méthode Base64 n'est pas une méthode cryptologique, en effet elle n'utilise pas de clé. Son objectif est de transformer un message binaire en ASCII et inversement. Son usage est par exemple indispensable quand on transmet un courrier électronique qui contient du binaire.

Comme un message chiffré est souvent constitué d'une suite de bits, il est souvent nécessaire de le coder ensuite en utilisant Base64 pour qu'il puisse être manipulé comme un texte : imprimé dans un journal, affiché dans une page Web, transmis par e-mail, etc.

La taille des clés

Les algorithmes à clés secrètes utilisent des clés plus ou moins longues. Un pirate dispose d'une méthode simple pour trouver le message en clair : c'est d'essayer toutes les clés possibles. Cette méthode, appelée « l'attaque exhaustive », est d'autant plus efficace que les clés sont courtes.

On exprime la taille d'une clé (et donc le nombre de clés possibles) en puissance de 2. Par exemple, une clé de 40 bits signifie qu'il existe 2**40 clés possibles (mille milliards). Les clés de taille 40 bits sont particulièrement vulnérables. Cette taille correspond à la taille des clés des logiciels américains destinés à l'exportation. Les clés de 56 bits, utilisées par le DES, sont cassables, mais avec des moyens considérables (des centaines d'ordinateurs). Signalons que la méthode de Jules César utilise des clés de 5 bits.

Remarque : Rocke Verser en 1997, avec l'aide de milliers d'ordinateurs connectés par Internet, a décrypté un message chiffré en DES 56 bits.

Les experts considèrent qu'un algorithme à clé secrète doit utiliser des clés de plus de 100 bits (128 bits par exemple) pour assurer une sécurité parfaite.

AES	Utilise des clés de 128, 192 ou 256 bits.
Camellia	Utilise des clés de 128, 192 ou 256 bits.
3DES	Utilise des clés de 112 bits.
DES	Utilise des clés de 56 bits.
RC4	Utilise des clés de 40 à 256 bits.
Blowfish	Utilise des clés de 32 à 448 bits.
CAST5	Utilise des clés de 128 bits.
IDEA	Utilise des clés de 128 bits.

Qualités d'un système de chiffrement symétrique

Il existe des centaines de systèmes de chiffrement symétrique, lequel (lesquels) choisir ? Voici quelques critères :

- Il doit résister à une attaque exhaustive. Pour ce faire la taille des clés utilisées doit être au moins de 128 bits.
- Il ne doit pas contenir de faille théorique qui permettrait de décrypter les messages sans explorer toutes les clés. Seuls des mathématiciens peuvent réaliser ce type d'attaque. En conséquence, la meilleure protection est la plus grande publicité. Ainsi il ne faut utiliser que les méthodes connues précitées (AES, Camellia, 3DES, RC4, Blowfish, Cast5, IDEA).
- Il doit être rapide.
- Il doit être simple. Dans ce cas, il sera plus facile de détecter une faille théorique et il sera également plus facile à optimiser de manière logiciel ou matériel.

Remarque : le standard AES (qui par défaut fonctionne en 256 bits) est particulièrement simple à comprendre et il a été choisi également car il est très rapide sous forme logiciel et matériel.

Le savoir concret

Les commandes

`openssl`	Commande générique de chiffrement.
`gpg`	Commande générique de chiffrement.
`vim`	L'éditeur `vim` dispose de l'option `-x` qui permet de chiffrer un fichier.

Chiffrer/déchiffrer un fichier avec GPG

a) Chiffrer un fichier (clair.txt)

```
$ cat  clair.txt
bonjour
$ gpg --symmetric clair.txt
Enter passphrase: password
```

b) Déchiffrer un fichier

```
$ gpg --decrypt clair.txt.gpg
gpg: CAST5 encrypted data
Enter passphrase: password
```

```
gpg: encrypted with 1 passphrase
bonjour
```

Chiffrer/déchiffrer un fichier avec OpenSSL

a) Chiffrer un fichier (clair.txt) avec la méthode AES 256 bits.

```
$ cat  clair.txt
bonjour
$ openssl aes-256-cbc -k secret -in clair.txt -out crypto.dat
$ rm clair.txt
```

b) Déchiffrer un fichier (l'option –d implique le déchiffrement).

```
$ openssl aes-256-cbc -d -k secret <  crypto.dat  > clair.txt
$ cat  clair.txt
bonjour
```

Les particularités des distributions

Debian/Ubuntu

Plusieurs commandes permettent de chiffrer des fichiers : `bcrypt`, `beecrypt`, `ccrypt` et `mcrypt`.

La commande `mcrypt` sert à chiffrer des fichiers. Elle prend en charge la plupart des algorithmes symétriques : Blowfish, DES, 3DES, CAST, RC2, IDEA, Twofish, RC6, GOST. Une option lui permet d'être compatible avec la commande `crypt` d'Unix (basée sur la fonction ISO crypt(3) dérivée du DES).

Pour en savoir plus

Man

gpg(1), openssl(1), enc(1), des_modes(7), vim(1)

Internet

Wikipedia – Les algorithmes de chiffrement symétriques
http://en.wikipedia.org/wiki/Symmetric-key_algorithm

Wikipedia – AES
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Wikipedia - 3DES
http://en.wikipedia.org/wiki/Triple_DES

Wikipedia - Les différents modes d'utilisation des méthodes blocs
http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

Wikipedia - IDEA
http://en.wikipedia.org/wiki/International_Data_Encryption_Algorithm

Wikipedia - RC4
http://en.wikipedia.org/wiki/RC4

Wikipedia – Blowfish
http://www.uqtr.ca/~delisle/Crypto/prives/blocs_blowfish.php

Wikipedia - Base64
http://fr.wikipedia.org/wiki/Base64

Les fonctions de hachage

La théorie

Les fonctions de hachage (ou générateurs d'empreintes ou « hash functions ») ont un rôle similaire à celui des calculs de « CRC ». Elles permettent de créer l'empreinte numérique d'un message. Cette empreinte dépend mathématiquement du message. Si l'on modifie un tant soit peu le message, l'empreinte associée est complètement différente. Par conséquent les fonctions de hachage vérifient qu'un message (ou un fichier) n'a pas été altéré.

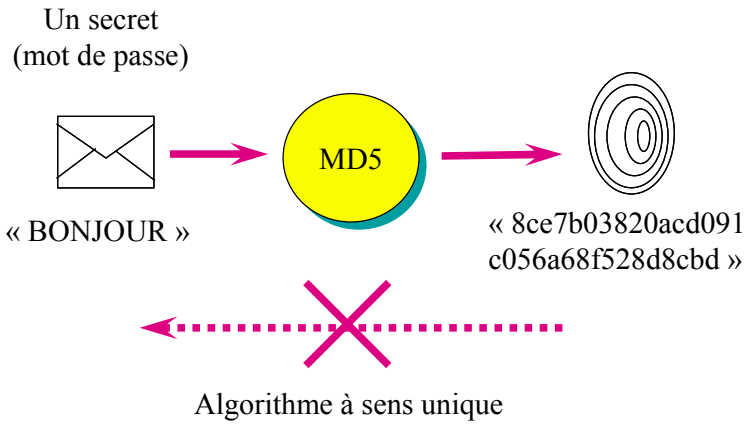


Fig. La génération d'une empreinte MD5

Panorama des fonctions de hachage

SHA-1	Standard FIPS.
MD5	Standard IETF.

MD5 et SHA sont les deux principaux algorithmes de hachage. Mais il y en a d'autres : MD2, MD4, N-Hash, Snefrou, Ripend160, MDC2...

Remarque : durant l'été 2004, le MD5 a été cassé par des chinois (Xiaoyun Wang...).

La mémorisation des mots de passe

Selon les systèmes Unix ou distributions Linux, on utilise soit des méthodes de chiffrement soit des fonctions de hachage pour mémoriser un mot de passe. Le fait qu'une fonction de hachage soit à sens unique n'est pas un inconvénient, car on ne fait que vérifier si le mot de passe fourni est équivalent, après chiffrement, à celui qui est stocké.

Pour éviter que deux utilisateurs possédant le même mot de passe aient également le même mot de passe chiffré, une graine (salt) composée de caractères aléatoires, est ajoutée au mot de passe avant son chiffrement. Cette graine est ajoutée en clair devant le mot de passe chiffré.

Remarque : du fait de l'augmentation de la puissance des ordinateurs domestiques (ou plutôt de leur carte graphique) et des consoles de jeux (PS2...), les mots de passe doivent être protégés par des algorithmes qui demandent beaucoup de temps de calcul. Actuellement, les systèmes Linux récents ont abandonné MD5 pour SHA-512 pour ces raisons.

Sécurité

Les différentes fonctions de hachage actuelles présentent des failles cryptologiques. En clair, des messages différents peuvent avoir les mêmes empreintes.

Par contre, l'usage de plusieurs fonctions offre une véritable sécurité. En effet, s'il est possible théoriquement (et pratiquement dans le cas du MD5) d'imiter une empreinte, il est quasi impossible d'en imiter plusieurs simultanément.

Le savoir concret

Les commandes

`sum`	Calcule une somme de contrôle (commande standard).
`md5sum`	Calcule une somme MD5.
`sha1sum`	Calcule une somme SHA-1.
`openssl`	Outil cryptographique générique.

Un exemple avec md5sum

```
$ md5sum /bin/bash
ceb0c3d1c38d515265ba6faa21fe47ed  /bin/bash
```

Remarque : si le MD5 ne peut plus être utilisé pour l'authentification, il est toujours utile pour vérifier la bonne copie (physique, réseau) d'un fichier.

Les particularités des distributions

Debian/Ubuntu

Le paquet `crack-md5` permet de casser du MD5.

Pour en savoir plus

RFC

MD5 (rfc1321)

Man

md5sum(1), sha1sum(1), gpg(1), dgst(1), sum(1)

Internet

Wikipedia - Les fonctions de hachage
http://fr.wikipedia.org/wiki/Fonction_de_hachage

Wikipedia - MD5
http://fr.wikipedia.org/wiki/MD5

Wikipedia - SHA-1
http://fr.wikipedia.org/wiki/SHA-1

SHA-1
http://www.itl.nist.gov/fipspubs/fip180-1.htm

MD5 collision DEMO
http://www.mscs.dal.ca/~selinger/md5collision/

BarWF: le logiciel le plus rapide pour casser MD5
http://3.14.by/en/md5

Cracker un mot de passe en GPGPU
http://www.presence-pc.com/tests/password-recovery-gpu-23381/6/

Les algorithmes de chiffrement à clé publique

La théorie

Un algorithme asymétrique, dit également à clé publique, utilise deux clés, une clé publique et une clé privée. Ces deux clés sont générées ensemble par un logiciel et elles dépendent mathématiquement l'une de l'autre. Comme son nom l'indique, la clé publique peut être publiée sans risque, mais la clé privée doit être conservée jalousement secrète par son propriétaire. La clé publique sert habituellement à crypter un message et la clé privée à le décrypter, mais l'inverse est possible.

Exemple : la méthode RSA

La méthode à clé publique la plus utilisée est le RSA, du nom de ses inventeurs, Rivest, Shamir et Adleman. La société RSA Data Security (RSADSI) en détenait les brevets, qui ont expiré en l'an 2000.

Alice veut protéger la confidentialité des messages de ses prétendants. Elle génère un couple de clés. Elle distribue sa clé publique à tous ses admirateurs, et conserve sa clé privée. Si Bob veut envoyer un message chiffré à Alice, il utilise l'algorithme RSA avec la clé publique d'Alice. Le message chiffré ne peut être lu que par Alice, grâce à sa clé privée.

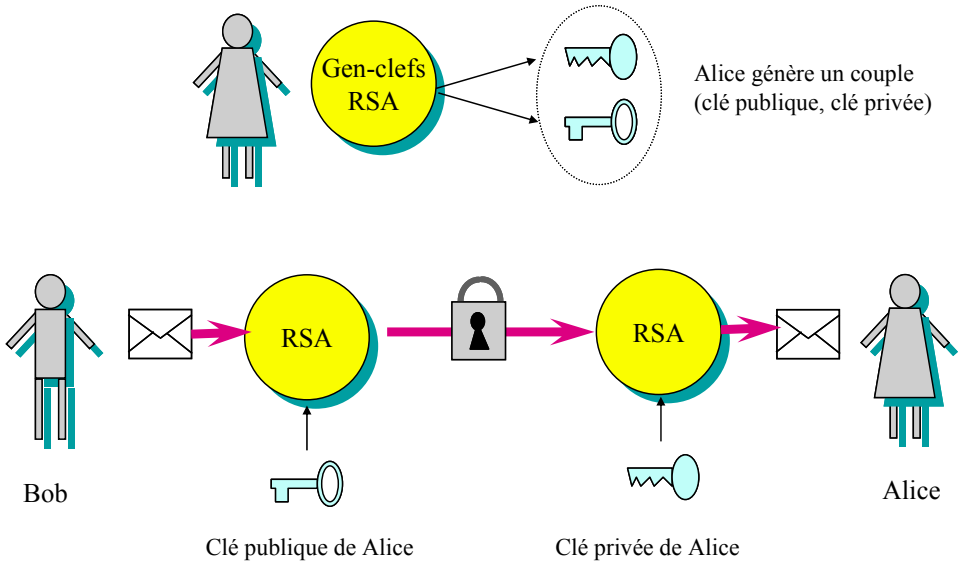


Fig. Le chiffrement asymétrique RSA

Remarque

Du fait de la lenteur des algorithmes à clés publiques, ils ne sont utilisés que pour chiffrer des messages très courts, par exemple des clés secrètes, appelées alors clés de session, utilisées par les algorithmes symétriques.

Les avantages des méthodes à clés publiques

La cryptographie à clé publique offre plusieurs avantages :

- Elle diminue le nombre de clés nécessaires à la communication entre un nombre important de personnes.
- Elle permet la signature d'un document numérique.
- Elle permet l'authentification mutuelle de deux correspondants.

Diminution du nombre de clés

Si quatre personnes veulent communiquer avec des méthodes à clés secrètes, il faut six clés secrètes pour couvrir toutes les combinaisons de dialogue secret. Il en faut seulement quatre avec RSA. Avec les méthodes à clés secrètes, le nombre de clés devient exponentiel au fur et à mesure que le nombre de personnes augmente.

L'authentification mutuelle

Bob veut envoyer un message à Alice, mais il désire que le message reste secret et que seule Alice puisse le lire et qu'Alice soit sûre de son origine.

1. Bob chiffre son message avec sa clé secrète.
2. Bob surchiffre le résultat avec la clé publique d'Alice.
3. Bob transmet le message à Alice.
4. Alice utilise sa clé privée pour déchiffrer le message.
5. Alice utilise ensuite la clé publique de Bob pour déchiffrer le résultat précédent. Elle obtient le message en clair qui ne peut provenir que de Bob et elle seule a pu le déchiffrer.

Panorama des méthodes à clés publiques

- Diffie-Hellman.
- RSA (Ronald Rivest, Adi Shamir et Leonard Adleman).
- DSA ou DSS (Digital Signature Algorithm ou Digital Signature Standard) est un standard américain du NIST (National Institute of Standards and Technologies).
- Les méthodes elliptiques.

Remarques

1) La méthode Diffie-Hellman ne permet pas de chiffrer des fichiers ou de signer des données. Elle sert principalement à établir des connexions sécurisées en créant dynamiquement une clé secrète entre deux interlocuteurs de telle manière qu'un pirate à l'écoute des échanges ne puisse la deviner.

2) Le DSA, à l'origine n'a été conçu que pour signer des messages.

L'attaque de l'homme du milieu

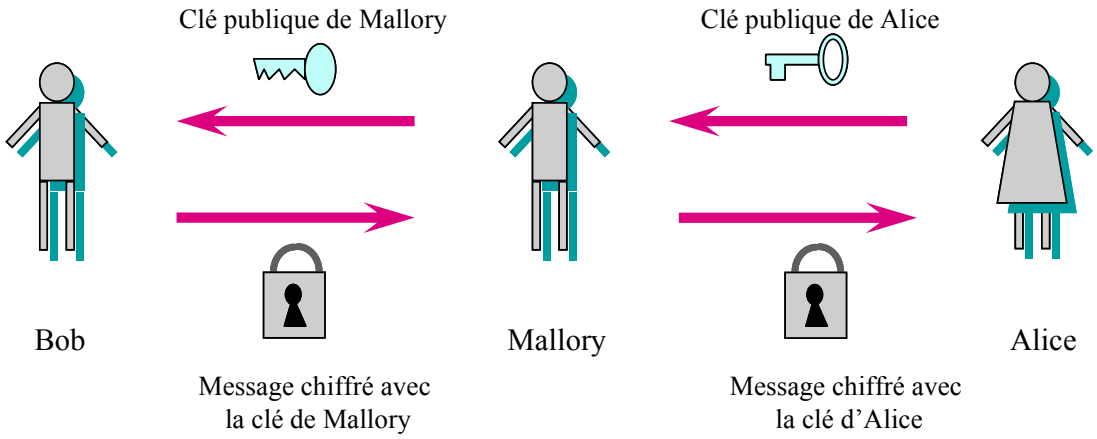


Fig. L'attaque de l'homme du milieu

L'utilisation de méthodes asymétriques résout le problème de l'échange de clés symétriques, mais il est sensible à une attaque subtile, l'attaque de l'homme du milieu :

1. Alice envoie à Bob sa clé publique. Mallory intercepte cette clé et envoie à Bob sa propre clé publique.
2. Bob envoie un message à Alice, chiffré avec la clé de Mallory. Mallory intercepte le message, le déchiffre avec sa clé privée. Il le rechiffre avec la clé publique d'Alice et l'envoie à Alice.

Pour lutter contre ce type d'attaque, les correspondants doivent s'authentifier. L'utilisation de certificats x509 est une solution à ce problème.

Les standards PKCS

La société RSA Data Security a publié des documents ayant pour objectif de normaliser l'utilisation des techniques à clés publiques. Ces documents sont nommés PKCS (Public Key Cryptography Standards). Ils sont devenus rapidement des standards de facto.

Voici une brève description de ces documents.

PKCS#1	Décrit le chiffrement/déchiffrement RSA.
PKCS#3	Décrit l'implémentation de l'algorithme Diffie-Hellman.
PKCS#5	Décrit une méthode pour chiffrer les clés privées.
PKCS#6	Extension de la norme X509 (rendue obsolète par la norme x509v3).
PKCS#7	Syntaxe récursive pour décrire des données chiffrées ou signées comme des signatures électroniques, des certificats… PKCS#7 est compatible avec PEM.
PKCS#8	Décrit une syntaxe pour les clés privées.
PKCS#10	Décrit une syntaxe pour des demandes de certificats.
PKCS#11	API, appelée Cryptoki (Cryptographic Token API Standard), permettant de réaliser des opérations cryptographiques avec du matériel (smartcard…).
PKCS#12	Décrit une syntaxe pour stocker dans un fichier des clés publiques d'utilisateur, des clés privées protégées, des certificats.

La taille des clés

Comme dans le cas des méthodes symétriques, plus la taille des clés est grande et meilleure est la sécurité.

En revanche, la taille des clés à utiliser n'a pas de rapport avec une attaque exhaustive. La taille minimale à utiliser dépend de l'algorithme considéré. Les experts considèrent par exemple, que pour protéger des informations grâce à la méthode RSA, il faut utiliser des clés d'au moins 1024 bits. Des clés de 2048 bits assureront même une sécurité à plus long terme. La méthode RSA est sensible à une attaque par factorisation de grands nombres. Si les mathématiques évoluent, on peut imaginer même que ces clés n'offrent plus de sécurité.

La sécurité d'une méthode cryptographique

La sécurité d'une méthode cryptographique ne repose pas entièrement sur la taille des clés. Une méthode cryptographique peut être attaquée « mathématiquement ». C'est ainsi que la méthode des sacs à dos, une des premières méthodes asymétriques a été « cassée ». Inversement, le DES qui a subi l'assaut de milliers de mathématiciens résiste toujours. Par contre, le DES traditionnel avec des clés de 56 bits a été abandonné pour faire place au TripleDES (DES3) qui se sert de clés de 56x2 ou 56x3 bits.

En cryptologie on fera toujours plus confiance à un algorithme ou un protocole qui a été publié qu'à une technique propriétaire conservée secrète. La publicité faite autour d'une

technique cryptographique assure qu'il n'y a pas de faille. Les découvrir devient en effet un challenge de la communauté scientifique. Dans certains cas, ces challenges peuvent même rapporter beaucoup d'argent.

Le savoir concret

Les commandes

`openssl`	Crée des clés RSA ou DSA
`gpg`	Implémente le protocole PGP qui utilise les méthodes asymétriques.

Pour en savoir plus

Man

random(4), gpg(1), openssl(1), dh(1), dsa(1), dsaparam(1), genrsa(1), gendsa(1), pkcs7(1), pkcs12(1), rsa(1), rsautl(1)

Internet

WIkipedia – Les algorithmes cryptographiques à clé asymétrique
http://fr.wikipedia.org/wiki/Cryptographie_asymétrique
http://en.wikipedia.org/wiki/Public-key_cryptography

Le protocole de Diffie et Hellman
http://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange
http://www.bibmath.net/crypto/moderne/difhel.php3

RSA
http://en.wikipedia.org/wiki/RSA
http://www.bibmath.net/crypto/moderne/rsa.php3

DSA
http://en.wikipedia.org/wiki/Digital_Signature_Algorithm
http://www.itl.nist.gov/fipspubs/fip186.htm

Les courbes elliptiques
http://en.wikipedia.org/wiki/Elliptic_curve_cryptography

Les standards PKCS
ftp://ftp.rsa.com/pub/pkcs/
http://en.wikipedia.org/wiki/PKCS

La signature numérique

La théorie

Les algorithmes à clés asymétriques, nommés également algorithmes à clés publiques, offrent la possibilité d'effectuer des signatures électroniques. On les appelle aussi signatures numériques, « Digital signatures », ou encore sceaux numériques.

Une signature numérique, comme une signature manuscrite, a pour objectif d'identifier l'auteur d'un document et d'en prévenir toute falsification.

Le principe

Bob crée un document, un logiciel source par exemple, et il calcule une empreinte (via les algorithmes MD5 ou SHA) à partir de ce document. Il utilise ensuite sa clé privée pour chiffrer cette donnée. Il associe ensuite le résultat au document.

Chacun peut recalculer la somme de contrôle du document et vérifier avec la clé publique de Bob que le document est authentique et que Bob en est l'auteur.

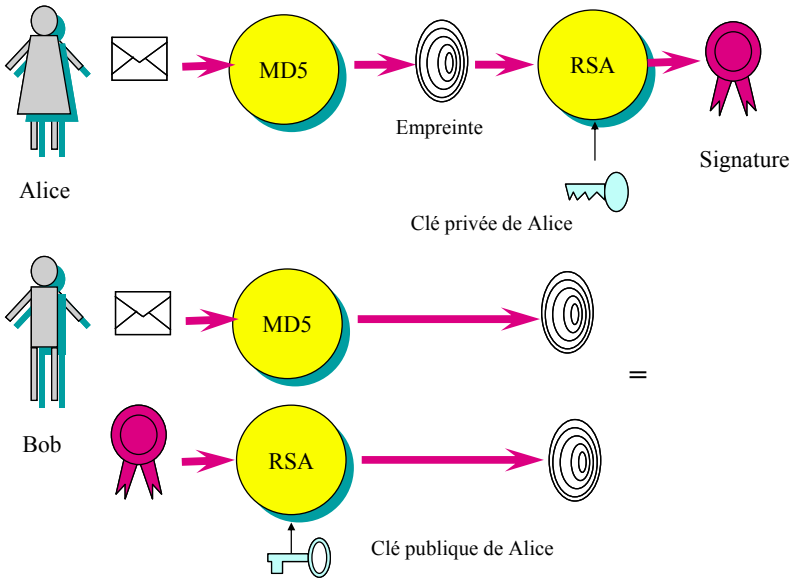


Fig. La signature numérique

Le savoir concret

Les commandes

`gpg`	Logiciel de cryptographie, permet notamment la signature de document.
`rpm`	La commande `rpm` peut vérifier la signature d'un paquetage.
`yum`	La commande de gestion de paquetage reposant sur `rpm`. Elle peut invoquer la vérification des signatures des paquetages.

Les fichiers

/etc/yum.conf	Le fichier de configuration de `yum`. gpgcheck= Indique si l'on vérifie les signatures. gpgkey= L'emplacement de la clé gpg publique de vérification.

Les particularités des distributions

Debian

Le paquetage `apt` contient la commande `apt-key` qui gère les clés servant à authentifier les paquetages.

Remarque
Actuellement, `apt-get` ne fait que signaler qu'un paquet n'est pas signé. Les versions futures seront sans doute plus sévères (cf. apt-secure(8)).

Quelques paquetages complémentaires :

`debsigs`	Gère les signatures stockées dans un paquetage Debian.
`debian-keyring`	Les clés des développeurs Debian.
`debsig-verify`	Vérifie les signatures d'un paquetage Debian.
`dpkg-sig`	Crée et vérifie les signatures d'un paquetage Debian.

Pour en savoir plus

Man

gpg(1), gpgv(1), openssl(1), yum.conf(5), rpm(8)

Internet

Wikipedia - La signature numérique
http://fr.wikipedia.org/wiki/Signature_num%C3%A9rique

DSS (le standard américain)
http://www.itl.nist.gov/fipspubs/fip186.htm

Debian: SecureApt
http://wiki.debian.org/SecureApt

Les protocoles cryptographiques

La théorie

Les protocoles cryptographiques résolvent des problèmes de sécurité. Ils constituent la matière la plus concrète de la cryptologie. En tant que protocoles, ils sont constitués de règles d'échanges entre parties. Pour résoudre les problèmes de sécurité (confidentialité…), ils utilisent des algorithmes cryptographiques (de chiffrement, de générateurs d'empreintes…).

Les problèmes de sécurité

La plupart des protocoles de sécurité permettent l'échange de données de manière confidentielle et authentifiée.

Mais il existe d'autres problèmes réseau qui nécessitent de la sécurité, par exemple :

- Acheter sur Internet.
- Jouer au poker en réseau.
- Voter de manière électronique.

La sécurité

Il faut privilégier les protocoles éprouvés. Leur normalisation ainsi que celle des algorithmes sous-jacents (AES, 3DES…) est une garantie de sécurité. L'usage de clés symétriques faibles (inférieures à 100 bits) doit être prohibé.

Remarque
Un protocole comme SSL protège les données en transit. Il n'offre aucune protection si votre poste de travail est infecté par un logiciel qui scrute la mémoire et a accès aux données avant leur chiffrement.

Panorama des protocoles

Cram-MD5	Protocole d'authentification
PGP	Protocole pour chiffrer et signer des e-mails
S/MIME	Idem, mais basé sur une PKI x509
Kerberos	Protocole assurant l'authentification et la confidentialité
SSH	Idem
SSL	Idem, basé sur une PKI x509
IPSEC	Protocole VPN

Un exemple de protocole : Cram-md5

Cram-md5 est un protocole d'authentification. Voici son principe :

Le serveur envoie une suite aléatoire d'octets au client. Elle sert de challenge. Le client répond avec son nom suivi d'un espace et d'une empreinte MD5 en hexadécimal. Celle-ci est calculée à partir du mot de passe de l'utilisateur et du challenge. Le serveur fait le même calcul et le compare avec la réponse du client. S'il y a correspondance, l'utilisateur est authentifié.

Pour en savoir plus

Internet

Wikipedia - Category:Cryptographic protocols
http://en.wikipedia.org/wiki/Category:Cryptographic_protocols

Wilkipedia - Cram-MD5
http://en.wikipedia.org/wiki/CRAM-MD5

Livre

Applied cryptography, par Bruce Schneier (2003).

Le protocole OpenPGP

La théorie

PGP est un protocole cryptographique qui assure la confidentialité et l'authentification d'e-mails. Il dérive du protocole interne au logiciel PGP créé par Phil Zimmermann en 1991. L'IETF a normalisé ce protocole sous le nom d'OpenPGP.

Le protocole PGP

Un message PGP est constitué d'une suite de paquets. Chaque paquet commence par un en-tête qui décrit son type et sa longueur. Voici les principaux types de paquets :

- Clé publique.
- Clé secrète chiffrée par une clé symétrique.
- Clé de session chiffrée par une clé symétrique.
- Clé de session chiffrée par une clé publique.
- Signature.
- Données chiffrées par une clé de session (symétrique).

Par exemple, un paquet « Clé de session chiffrée par une clé publique » contient les informations suivantes :

- La version.
- L'identification de la clé publique.
- L'algorithme asymétrique utilisé.
- La clé de session chiffrée par la clé publique.

Algorithme de chiffrement et de déchiffrement.

1. Saisie du message à envoyer.
2. Génération d'une clé de session aléatoire.
3. La clé de session est chiffrée avec la clé publique du correspondant. Cette information, précédée de l'identifiant de cette clé est écrite en tête du message PGP.
4. Le logiciel crypte le message à envoyer (préalablement compressé) avec la clé de session et ajoute le résultat au message PGP. Ce dernier peut être transmis au correspondant.
5. Le destinataire déchiffre la clé de session avec sa clé privée.
6. Le destinataire déchiffre le message avec la clé de session.

Le savoir concret

La commande gpg

La commande GPG (Gnu Privacy Guard) implémente le protocole OpenPGP et permet donc de chiffrer et signer des e-mails.

Exemples

Créer une paire de clés asymétriques

```
$ gpg --gen-key
```

Exporter une clé

```
$ gpg --armor  --export  "Alice DeNice" > cle.asc
```

Importer une clé

```
$ gpg --armor  --import cle.asc
```

Chiffrer un message

```
$ gpg -r Alice --encrypt  msg.txt
```

Déchiffrer un message

```
$ gpg  msg.txt.gpg
```

Signer un message

```
$ gpg --clearsign  -u Alice clair.txt
```

Vérifier une signature

```
$ gpg --verify   clair.txt.asc
```

Lister ses clés publiques

```
$ gpg --list-public-keys
```

Lister ses clés secrètes

```
$ gpg --list-secret-keys
```

Les particularités des distributions

SuSE

SuSE possède le paquetage `gpg` ainsi que le paquetage `gpg2`. Ce dernier contient une version évoluée de GPG orienté desktop. Cette version inclus notamment GPGSM (qui apporte la compatibilité S/MIME).

Debian

Quelques paquetages :

`gnupg`	Le logiciel GPG.
`kgpg`	Interface graphique à GPG.
`gpgp`	Interface graphique à GPG – remplace PGP.

Pour en savoir plus

Man

gpg(1),

RFC

PGP (rfc 1991), OpenPGP v2 (rfc 2440)

Internet

Wikipedia – PGP
http://fr.wikipedia.org/wiki/Pretty_Good_Privacy

GPG
http://www.gnupg.org/
http://www.gnupg.org/gph
http://www.gnupg.org/docs.html

Site serveur de clés PGP
http://www.keyserver.net

Secure my email - encryption and digital signatures
(comment chiffrer vos e-mails avec Thunderbird et Evolution)
http://www.secure-my-email.com/

Livres

PGP Pretty Good Privacy, par Simson Garfinkel (décrit PGP et son histoire) (1994).

PGP et GPG Assurer la confidentialité de ses e-mails et fichiers, par Michael W. Lucas (2006)

Le chiffrement des systèmes de fichiers

La théorie

Un programme comme `gpg` permet de chiffrer un fichier. Si l'on désire qu'un ensemble de fichiers soit protégé par le cryptage, il est préférable d'utiliser le chiffrement complet du système de fichiers qui l'abrite.

Panorama des solutions

CFS	Système de chiffrement de FS utilisant NFS.
encFS	Successeur de CFS, ce système chiffre les données dans l'espace utilisateur.
Cryptoloop	Chiffrement effectué par le noyau au niveau du pilote loop.
dm-crypt	Successeur de Cryptoloop. Utilise le pilote device-mapper.

Le savoir concret

encFS - Commandes

`encfs`	Crée un FS chiffré.
`fusermount`	Monte le FS.

Cryptoloop - Commande

`losetup`	Crée un périphérique bloc à partir d'un fichier, permet le chiffrement.

dm-crypt (ou LUKS: Linux Unified Key Setup)

Commande

`cryptsetup`	Gère le chiffrement d'un périphérique bloc (création…).

Fichier

/etc/crypttab	Spécifie l'emplacement des clés et les méthodes de chiffrement des périphériques blocs activés au démarrage.

Les particularités des distributions

SuSE

Les paquetages suivants gèrent le chiffrement des FS :

`util-linux-crypto`	Inclut la commande `cryptsetup`.
`encfs`	Crée un FS chiffré.

Debian

Les paquetages suivants gèrent le chiffrement des FS :

`cfs`	Le chiffrement d'arborescence dans l'espace utilisateur (utilise NFS). Il est composé des commandes `cattach` et `cdetach` et du démon `cfsd`.
`encfs`	Chiffre des FS virtuels.
`aespipe`	Chiffre en AES, peut être utilisé avec les pilotes loop-aes-modules.

`cryptsetup` Configure le cryptage des périphériques blocs (dm-crypt).

`cryptmount` Monte des FS cryptés (compatible dm-crypt).

Lors de l'installation d'un système Debian, on peut choisir les LVM cryptés.

Pour en savoir plus

Man

encFS : encfs(1), fusermount(1)

cryptoloop : losetup(8)

dm-crypt : cryptsetup(8), crypttab(5)

kpartx(8)

Howto

Cryptoloop-HOWTO

Internet

Using CFS
http://www.linuxjournal.com/article/6381

encFS
http://arg0.net/wiki/encfs

http://arg0.net/wiki/encfs/intro2

Cryptoloop
http://en.wikipedia.org/wiki/Cryptoloop

dm-crypt: a device-mapper crypto target
http://www.saout.de/misc/dm-crypt/

TrueCrypt – Solution OpenSource de chiffrement de disques (clé USB…) multi-plateformes (Windows Vista/XP, Mac OS X, Linux)
http://www.truecrypt.org

RedHat documentation: Luks
http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/sect-Security_Guide-LUKS_Disk_Encryption.html

ATELIERS

Tâche 1 : Chiffrer des fichiers, des sauvegardes

Nous sommes connectés sous un compte utilisateur (guest).

1. Créer un fichier texte et ensuite le chiffrer avec openssl.

```
[guest@linux01 ~]$ vi clair.txt
bonjour
salut
[guest@linux01 ~]$ openssl rc4 -e -in clair.txt -out crypto.dat
enter rc4 encryption password: secret
Verifying - enter rc4 encryption password: secret
[guest@linux01 ~]$ rm clair.txt
```

Le fichier crypto.dat résulte du chiffrement du fichier clair.txt. Le mot de passe permettant de générer la clé de chiffrement est le mot « secret ». Après le chiffrement, le fichier d'origine, clair.txt n'est plus nécessaire, on le détruit.

2. Déchiffrer le fichier chiffré précédemment.

```
[guest@linux01 ~]$ openssl rc4 -d -in crypto.dat -out clair.txt
enter rc4 decryption password: secret
[guest@linux01 ~]$ more clair.txt
bonjour
salut
```

3. Idem, mais on utilise la redirection des entrées/sorties standards. Le mot de passe est donné en argument.

```
[guest@linux01 ~]$ openssl rc4 -d -k secret < crypto.dat > clair2.txt
[guest@linux01 ~]$ tail -1 clair2.txt
salut
```

4. Initialiser GPG avant son utilisation.

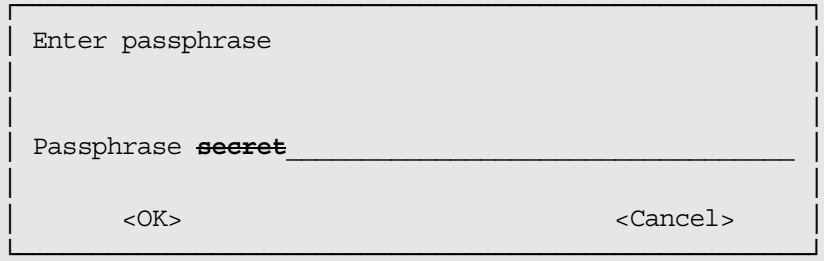
Avant d'utiliser GPG, il faut créer l'arborescence ~/.gnupg.

```
[guest@linux01 ~]$ gpg
gpg: Go ahead and type your message ...
Ctrl-D
gpg: processing message failed: Unknown system error
```

```
[guest@linux01 ~]$ ls -ld .gnupg
drwx------. 2 guest guest 4096 Sep 23 10:53 .gnupg
[guest@linux01 ~]$
```

5. Chiffrer et déchiffrer un fichier avec GPG.

```
[guest@linux01 ~]$ gpg --symmetric clair.txt
```

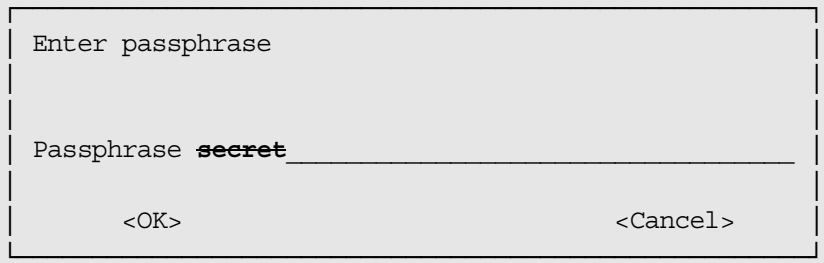



```
can't connect to `/home/guest/.gnupg/S.gpg-agent': No such file or directory
gpg-agent[25486]: directory `/home/guest/.gnupg/private-keys-v1.d' created
```

Remarque : un premier écran s'affiche pour demander un mot de passe. Le mot saisi ne s'affiche pas (il est remplacé par des étoiles). Un deuxième écran demande confirmation de la saisie.

6. Déchiffrer le fichier précédent, on utilise les entrées/sorties standards.

```
[guest@linux01 ~]$ gpg --decrypt clair.txt.gpg
gpg: 3DES encrypted data
```




```
can't connect to `/home/guest/.gnupg/S.gpg-agent': No such file or directory
gpg: encrypted with 1 passphrase
bonjour
salut
gpg: WARNING: message was not integrity protected
```

7. Chiffrer une archive TAR. On liste le contenu de l'archive.

```
[guest@linux01 ~]$ su -
Password: secret
[root@linux01 ~]# tar cf - /etc | openssl rc4 -e -k secret > /root/etc.tar.rc4
tar: Removing leading `/' from member names
[root@linux01 ~]# openssl rc4 -d -k secret < /root/etc.tar.rc4 | tar tvf - |
tail -3
-rw-r--r-- root/root     16384 2007-12-28 13:35:28 etc/pki/nssdb/secmod.db
-rw-r--r-- root/root     65536 2007-03-22 00:11:25 etc/pki/nssdb/cert8.db
-rw-r--r-- root/root     16384 2007-03-22 00:11:25 etc/pki/nssdb/key3.db
[root@linux01 ~]# exit
```

8. Chiffrer un fichier avec vim

Quand on active `vim` avec l'option -x, la commande vous demande un mot de passe pour protéger votre fichier. Ensuite ce mot de passe vous est démandé lors de chaque accès.

```
[root@linux01 ~]# rpm -q vim-enhanced
vim-enhanced-7.2.411-1.4.el6.i686
[guest@linux01 ~]$ vim -x fichier.txt
[guest@linux01 ~]$ file fichier.txt
```