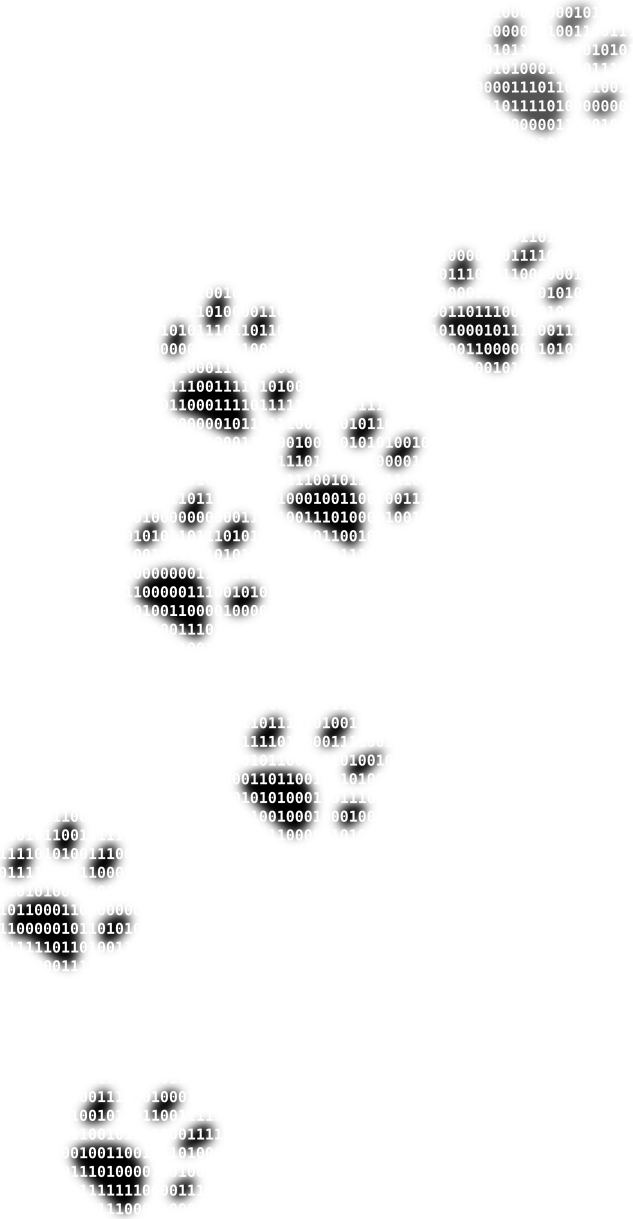
guide d'autodéfense numérique

tome 1
hors connexions



première édition
printemps 2010

ouvrage collectif

Guide d'autodéfense numérique
Tome 1 : Hors connexions

Ouvrage collectif
guide@boum.org

printemps 2010

Préface

Les revers de la mémoire numérique

De nos jours, les ordinateurs, l'Internet et le téléphone portable tendent à prendre de plus en plus de place dans nos vies. Le numérique semble souvent très pratique : c'est rapide, on peut parler avec plein de gens très loin, on peut avoir toute son histoire en photos, on peut écrire facilement des textes bien mis en page… Mais ça n'a pas que des avantages ; ou en tout cas, ça n'en a pas seulement pour nous, mais aussi pour d'autres personnes qu'on n'a pas forcément envie d'aider.

Il est en effet bien plus facile d'écouter discrètement des conversations par le biais des téléphones portables que dans une rue bruyante, ou de trouver les informations que l'on veut sur un disque dur, plutôt que dans une étagère débordante de papiers.

De plus, énormément de nos informations personnelles finissent par se retrouver publiées quelque part, que ce soit par nous-mêmes ou par d'autres personnes, que ce soit parce qu'on nous y incite — c'est un peu le fond de commerce du *web 2.0*, parce que les technologies laissent des traces, ou simplement parce qu'on ne fait pas attention.

Rien à cacher ?

« *Mais faut pas être parano : je n'ai rien à cacher !* » pourrait-on répondre au constat précédent…

Deux exemples tout bêtes tendent pourtant à montrer le contraire : personne ne souhaite voir ses codes secrets de carte bleue ou de compte *eBay* tomber entre n'importe quelles mains ; et personne non plus n'aimerait voir quelqu'un qui ne lui veut pas du bien débarquer chez lui parce que son adresse a été publiée sur Internet malgré lui…

Mais au-delà de ces bêtes questions de défense de la propriété privée, la confidentialité des données devrait être *en soi* un enjeu.

Tout d'abord, parce que ce n'est pas nous qui jugeons de ce qu'il est autorisé ou non de faire avec un ordinateur. Des personnes arrêtées pour des activités numériques qui ne plaisaient pas à leur gouvernement croupissent en prison dans tous les pays du monde — pas seulement en Chine ou en Iran.

De plus, ce qui est autorisé aujourd'hui, comment savoir ce qu'il en sera demain ? Les gouvernements changent, les lois et les situations aussi. Si on n'a pas à cacher aujourd'hui, par exemple, la fréquentation régulière d'un site web militant, comment savoir ce qu'il en sera si celui-ci se trouve lié à un processus de répression ? Les traces *auront été laissées* sur l'ordinateur… et pourraient être employées comme élément à charge.

Enfin et surtout, à l'époque des sociétés de contrôles de plus en plus paranoïaques, de plus en plus résolues à traquer la subversion et à voir derrière chaque citoyen un terroriste en puissance qu'il faut surveiller en conséquence, se cacher devient en soi

un enjeu *politique*, ne serait-ce que pour mettre des bâtons dans les roues de ceux qui nous voudraient transparents et repérables en permanence.

Quoi qu'il en soit, beaucoup de gens, que ce soient les gouvernants, les employeurs, les publicitaires ou les flics [1], ont un intérêt à obtenir l'accès à nos données, surtout au vu de la place qu'a pris l'information dans l'économie et la politique mondiales.

Tout ça peut amener à se dire que nous n'avons pas envie d'être contrôlables par quelque « Big Brother » que ce soit. Qu'il existe déjà ou que l'on anticipe son émergence, le mieux est sans doute de faire en sorte qu'il ne puisse pas utiliser, contre nous, tous ces merveilleux outils que nous offrent — ou que lui offrent — les technologies modernes.

Aussi, *ayons tous quelque chose à cacher, ne serait-ce que pour brouiller les pistes !*

Comprendre pour pouvoir choisir

Ce guide se veut une tentative de décrire dans des termes compréhensibles l'intimité (ou plutôt son absence) dans le monde numérique ; une mise au point sur certaines idées reçues, afin de mieux comprendre à quoi on s'expose dans tel ou tel usage de tel ou tel outil. Afin, aussi, de pouvoir faire le tri parmi les « solutions », toutes plus ou moins dangereuses si l'on ne se rend pas compte de ce contre quoi elles ne protègent pas.

À la lecture de ces quelques pages, on pourra avoir le sentiment que rien n'est vraiment sûr avec un ordinateur ; et bien, c'est vrai. Et c'est faux. Il y a des outils et des usages appropriés. Et souvent la question n'est finalement pas tant « doit-on utiliser ou pas ces technologies ? », mais plutôt « quand et comment les utiliser (ou pas) ? »

Prendre le temps de comprendre

Des logiciels simples d'utilisation meurent d'envie de se substituer à nos cerveaux… S'ils nous permettent un usage facile de l'informatique, ils nous enlèvent aussi prise sur les bouts de vie qu'on leur confie.

Avec l'accélération des ordinateurs, de nos connexions à Internet, est arrivé le règne de l'instantanéité. Grâce au téléphone portable et au Wi-Fi, faire le geste de décrocher un téléphone ou de brancher un câble réseau à son ordinateur pour communiquer est déjà désuet.

Être patient, prendre le temps d'apprendre ou de réfléchir deviendrait superflu : on veut tout, tout de suite, on veut *la* solution. Mais cela implique de confier de nombreuses décisions à de distants experts que l'on croit sur parole. Ce guide a pour but de proposer d'autres solutions, qui nécessitent de prendre le temps de les comprendre et de les appliquer.

Adapter ses pratiques à l'usage qu'on a du monde numérique est donc nécessaire dès lors qu'on veut, ou qu'on doit, apporter une certaine attention à son impact. Mais la traversée n'a que peu de sens en solitaire. Nous vous enjoignons donc à construire autour de vous votre radeau numérique, à sauter joyeusement à bord, sans oublier d'emmener ce guide et quelques fusées de détresse pour envoyer vos remarques à `guide@boum.org` (avec les précautions nécessaires).

1. On utilise ici le terme « flics » tel qu'il est défini dans l'introduction de *Face à la police / Face à la justice* [`http://guidejuridique.net/`].

Un « guide »

Ce guide est une tentative de rassembler ce que nous avons pu apprendre au cours de nos années de pratiques, d'erreurs, de réflexions et de discussions pour le partager.

Non seulement les technologies évoluent très vite, mais nous avons pu commettre des erreurs ou écrire des contre-vérités dans ces pages. Nous tenterons donc de tenir ces notes à jour à l'adresse : `https://guide.boum.org/`

Afin de rendre le tout plus digeste, nous avons divisé tout ce que nous souhaitions raconter en plusieurs tomes. Qu'on se trouve avec uniquement un ordinateur, que ce dernier soit connecté à un réseau ou qu'on soit chez soi ou au téléphone, cela représente des contextes différents, donc des menaces, des envies et des réponses différentes elles aussi.

Tome 1

Hors connexions

Sommaire

Première partie

Comprendre

Devant la grande complexité des outils informatiques et numériques, la quantité d'informations à avaler pour tenter d'acquérir quelques pratiques d'autodéfense peut paraître énorme. Elle l'est sûrement pour qui chercherait à tout comprendre en même temps…

Ce premier tome se concentrera donc sur l'utilisation d'un ordinateur « hors connexion » — on pourrait aussi bien dire *préalablement à toute connexion*. Mais ce sont aussi des connaissances plus générales qui valent *que l'ordinateur soit connecté ou non* à un réseau. On met donc de côté, jusqu'au second tome, les menaces spécifiquement liées à l'usage d'Internet et des réseaux.

Pour ce morceau *hors connexion*, comme pour les autres, on prendra le temps de s'attarder sur des notions de base, leurs implications en termes de sécurité / confidentialité / intimité[2]. Après l'analyse de cas concrets d'utilisation, on pourra se pencher sur quelques recettes pratiques.

Une dernière précision avant de nous jeter à l'eau : *l'illusion de sécurité est bien pire que la conscience nette d'une faiblesse*. Aussi, prenons le temps de bien lire les premières parties avant de nous jeter sur nos claviers… ou même de jeter nos ordinateurs par les fenêtres.

2. On souhaite ici faire appel à une notion un peu floue : quelque chose qui tournerait autour de la possibilité de décider ce qu'on révèle, à qui on le révèle, ainsi que ce que l'on garde secret ; quelque chose qui inclurait aussi une certaine attention à déjouer les tentatives de percer ces secrets. Le terme employé en anglais pour nommer ce qu'on évoque ici est *privacy*. Aucun mot français ne nous semble adapté pour porter tout le sens que l'on aimerait mettre derrière cette notion. Ailleurs, on rencontrera souvent le terme « sécurité », mais l'usage qui en est couramment fait nous donne envie d'éviter son usage.

Chapitre 1

Quelques bases sur les ordinateurs

Commençons par le commencement.

Un *ordinateur*, ce n'est pas un chapeau de magicien où on peut ranger des lapins et les ressortir quand on a besoin, et qui permettrait en appuyant sur le bon bouton d'avoir une fenêtre ouverte sur l'autre bout du monde.

Un ordinateur est composé d'un ensemble de machines plus ou moins complexes, reliées entre elles par des connexions électriques, des câbles, et parfois des ondes radios. Tout ce *matériel* stocke, transforme et réplique des signaux pour manipuler l'information que l'on peut voir sur un bel écran avec plein de boutons où cliquer.

Comprendre comment s'articulent ces principaux composants, comprendre les bases de ce qui fait fonctionner tout ça, c'est la première étape pour comprendre où sont les forces et les faiblesses de ces engins, à qui l'on confie pas mal de nos données.

1.1 Des machines à traiter les données

Les ordinateurs sont des machines inventées pour pouvoir s'occuper d'informations. Elles savent donc précisemment enregistrer, traiter, analyser et classer de l'information, même en très grande quantité.

Dans le monde numérique, copier une information ne coûte que quelques micro-watts, autant dire pas grand'chose : c'est essentiel d'avoir ça en tête si nous voulons limiter l'accès à des informations.

Il faut tout simplement considérer que *mettre une information sur un ordinateur* (et c'est encore plus vrai quand il est sur un réseau), *c'est accepter que cette information puisse nous échapper*.

Ce guide peut aider à limiter la casse, mais il faut malgré tout prendre acte de cette réalité.

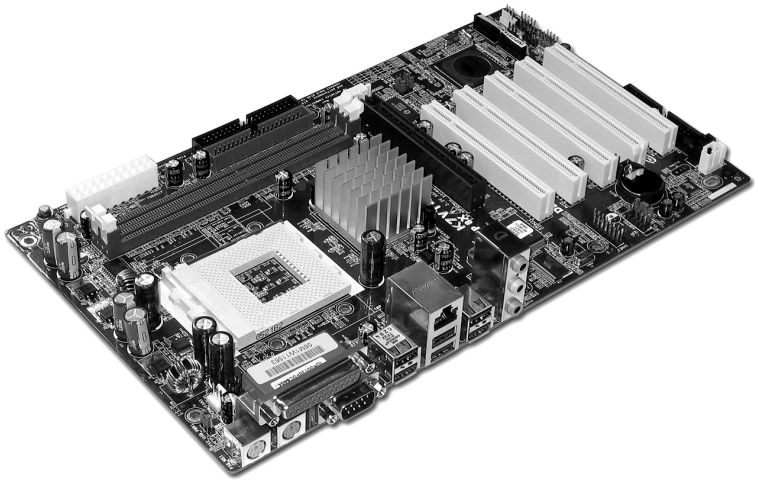
1.2 Le matériel

Somme de composants reliées entre eux, notre ordinateur est donc d'abord une accumulation d'objets, qu'on peut toucher, déplacer, bidouiller, casser.

L'ensemble *écran* / *clavier* / *tour* (ou unité centrale), ou l'ordinateur portable, est pratique quand on veut simplement brancher les fils aux bons endroits. Mais pour savoir ce qu'il advient de nos données, un examen plus fin est nécessaire.

On considère ici le contenu d'un ordinateur « classique », parfois appelé PC. Mais on retrouvera la plupart de ces composants avec de légères variations sur d'autres machines : Macs, téléphones portables, « box » de connexion à Internet, lecteur MP3, *etc.*

La carte-mère



Une carte-mère

Un ordinateur est surtout composé d'éléments électroniques. La *carte-mère* est un gros circuit imprimé qui permet de relier la plupart de ces éléments à travers l'équivalent de fils électriques. Sur la carte-mère viendront se brancher au minimum un processeur, de la mémoire vive, un système de stockage (disque dur), de quoi démarrer l'ordinateur (un BIOS) et d'autres cartes et périphériques selon les besoins.

On va rapidement faire un petit tour à travers tout ça pour avoir une vague idée de qui fait quoi, ce sera fort utile par la suite.

Le processeur

Le processeur (aussi appelé CPU, pour *central processing unit* ou « unité centrale de traitement » en français) est le composant qui s'occupe du traitement des données.

Pour se représenter le travail d'un processeur, l'exemple le plus concret sur lequel se baser est la calculatrice. Sur une calculatrice on entre des données (les nombres) et des opérations à faire dessus (addition, multiplication ou autres) avant d'examiner le résultat, éventuellement pour s'en servir ensuite comme base pour d'autres calculs.

Un processeur fonctionne exactement de la même manière. À partir de données (qui peuvent être la liste d'opération à effectuer), il se contente d'exécuter à la chaîne les traitements à faire. Il ne fait que ça, mais il le fait vraiment très vite.

Mais si le processeur n'est qu'une simple calculatrice, comment peut-on alors effectuer des traitements sur des informations qui ne sont pas des nombres, par exemple sur du texte, des images, du son ou un déplacement de la souris ?

Tout simplement en transformant en nombre tout ce qui ne l'est pas, en utilisant un code défini auparavant. Pour du texte, ça peut par exemple être `A = 65`, `B = 66`, *etc.* Une fois ce code défini, on peut *numériser* notre information. Avec le code précédent, on peut par exemple transformer « GUIDE » en `71, 85, 73, 44, 69`.

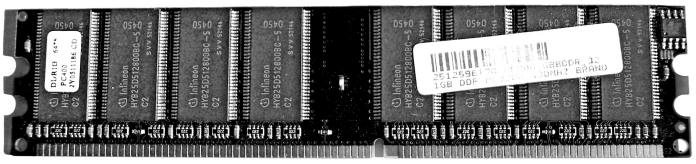
Les opérations que le processeur doit effectuer (ses *instructions*) sont également codées sous forme de nombres binaires. Un programme est donc une série d'instructions, manipulées comme n'importe quelles autres données.

La puce d'un microprocesseur Intel Pentium 60 Mhz dans son boîtier

À l'intérieur de l'ordinateur, tous ces nombres sont eux-mêmes représentés à l'aide d'états électriques : absence de courant, ou présence de courant. Il y a donc deux possibilités, ces fameux `0` et `1` que l'on peut croiser un peu partout. C'est pourquoi on parle de *bi*-naire. Et c'est uniquement à l'aide d'un paquet de fils et de plusieurs milliards de *transistors* (des interrupteurs, pas si différents de ceux pour allumer ou éteindre la lumière dans une cuisine) que le traitement des données se fait.

La mémoire vive

La mémoire vive (ou RAM, pour *Random Access Memory*) se présente souvent sous forme de *barrettes*, et se branche directement sur la carte-mère.



Une barrette de mémoire vive

La mémoire vive sert à stocker tous les logiciels et les documents ouverts. C'est à cet endroit que le processeur va chercher les données à traiter et entreposer le résultat des opérations. Ces informations doivent donc forcément s'y trouver sous une forme directement utilisable pour effectuer les calculs.

L'accès à la mémoire vive est très rapide : il suffit du temps nécessaire pour basculer les interrupteurs qui vont relier le processeur à la case de la mémoire à lire (ou à écrire).

Lorsque la mémoire vive n'est plus alimentée en électricité, les données qu'elle contient deviennent illisibles après quelques minutes ou quelques heures, selon les modèles.

Le disque dur

Un disque dur 3 pouces ½

Étant donné que la mémoire vive s'efface à partir du moment où elle n'a plus de courant, l'ordinateur a besoin d'un autre endroit où stocker données et programmes entre chaque allumage. On parle aussi de mémoire *persistante* ou de mémoire *morte* : une mémoire où les informations écrites restent, même sans alimentation électrique.

Pour ce faire, on utilise en général un *disque dur*. C'est souvent une coque en métal dans laquelle se trouvent plusieurs disques qui tournent sans s'arrêter. Sur ces disques se trouvent de minuscules morceaux de fer. Au-dessus de chaque disque se trouvent des *têtes de lecture*. À l'aide de champs magnétiques, ces dernières détectent et modifient la position des morceaux de fer. C'est la position des morceaux de fer qui permet de coder les information à stocker.

Ce mécanisme est *beaucoup plus lent* — 50 fois environ — que l'accès à la mémoire vive. Par contre, c'est plus simple d'y mettre *beaucoup plus d'informations*.

Les informations que l'on met donc généralement sur un disque dur sont, bien entendu, des documents, mais aussi les programmes et toutes les données qu'ils utilisent pour fonctionner, comme des fichiers temporaires, des journaux de bord, des fichiers de sauvegarde, des fichiers de configuration, *etc.*

Le disque dur conserve donc une mémoire quasi-permanente et quasi-exhaustive pour toutes sortes de traces qui parlent de nous, de ce que nous faisons, avec qui et comment, dès qu'on utilise un ordinateur.

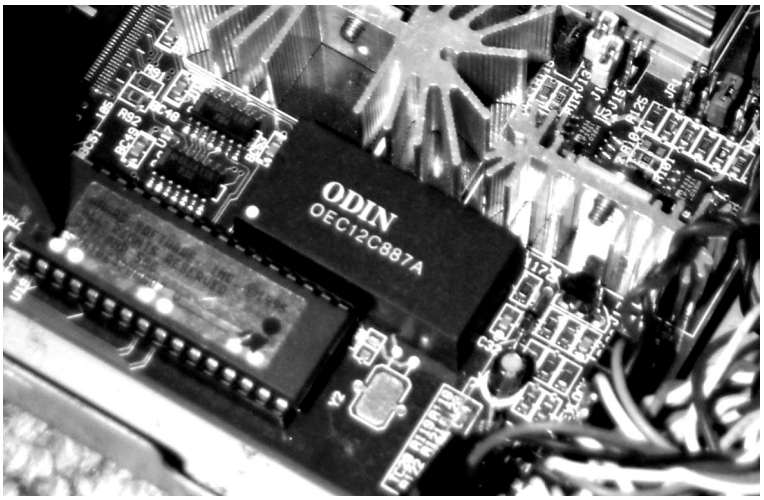
Les autres périphériques

Avec uniquement un processeur, de la mémoire vive et un support de stockage, on obtient déjà un ordinateur. Pas très causant, par contre. Donc on lui adjoint généralement d'autres *périphériques* comme un clavier, une souris, un écran, un adaptateur réseau (avec ou sans fil), un lecteur de DVD, *etc.*

Certains périphériques nécessitent des puces supplémentaires afin que le processeur puisse y accéder. Ces puces peuvent êtres soudées directement au circuit de la carte-mère (c'est typiquement le cas pour le clavier) ou alors nécessiter l'ajout d'un circuit supplémentaire, livré sous forme de carte (dite *fille*).

Afin de réduire le nombre de puces spécifiques (et donc coûteuses et compliquées à mettre au point), les systèmes d'accès aux périphériques tendent à s'uniformiser. Par exemple, la norme USB (pour *Universal Serial Bus*) est de plus en plus utilisée pour connecter imprimantes, claviers, souris, disques durs supplémentaires, adaptateurs réseaux ou ce qu'on appelle couramment des « clés USB ».

Le BIOS



Une puce de BIOS Award sur une carte-mère

Pour démarrer l'ordinateur, il faut donner au processeur un premier programme. Au moins pour pouvoir charger les programmes à exécuter ensuite.

C'est en général le rôle du BIOS (*Basic Input/Output System*, ou système d'entrée/sortie de base). Il s'agit d'un petit logiciel contenu dans une puce mémoire sur la carte mère. Cette mémoire fait partie d'un troisième type : la mémoire *flash*. C'est une mémoire qui garde les informations lorsqu'elle est éteinte, mais dont on ne peut remplacer le contenu que lors d'une opération qu'on appelle *flashage*. C'est aussi ce type de mémoire qu'on trouve dans les « clés USB ».

Ce premier programme qu'exécute l'ordinateur permet, entre autres, de choisir où se trouve le système d'exploitation que l'on veut utiliser (qui sera chargé à partir d'un disque dur, d'une clé USB, d'un CD-ROM, voire à partir du réseau).

1.3 Électricité, champs magnétiques et ondes radios

En ce qui concerne la confidentialité des informations qui circulent au sein d'un ordinateur, il faut déjà prendre acte de plusieurs choses après ce rapide tour de ce qui le compose.

Tout d'abord, l'essentiel de l'information circule sous forme de courants électriques. Rien n'empêche donc de mettre l'équivalent d'un bête *voltmètre* pour mesurer le courant qui passe, et ainsi pouvoir reconstituer n'importe quelles données manipulées par l'ordinateur sous une forme ou une autre.

Par ailleurs, tout courant qui circule a tendance à émettre un champ magnétique. Ces champs magnétiques peuvent rayonner à quelques mètres, voir plus [1]. Il est donc possible pour qui s'en donne les moyens de reconstituer le contenu d'un écran ou ce qui a été tapé sur un clavier, et cela, même derrière un mur, depuis la rue ou l'appartement contigu : ainsi, des chercheurs ont réussi à enregistrer les touches tapées sur des claviers filaires normaux à partir de leurs émissions électromagnétiques, à une distance allant jusqu'à 20 mètres [2].

Le même type d'opération est possible à partir de l'observation des légères perturbations que génère l'ordinateur sur le réseau électrique où il est branché. Il faut toutefois pour cela que l'attaquant soit branché sur le même réseau électrique.

Enfin, certains périphériques (claviers, souris, écouteurs, *etc.*) fonctionnent *sans fil*. Ils communiquent alors avec l'ordinateur par des ondes radio que n'importe qui autour peut capter et éventuellement décoder sans vergogne.

Bref, pour résumer, même si un ordinateur n'est pas connecté à un réseau, et quels que soient les programmes qui fonctionnent, il reste pour autant possible pour des personnes bien équipées de réaliser une « écoute » de ce qui se passe à l'intérieur de l'ordinateur.

1.4 Les logiciels

Au-delà de la somme d'éléments *physiques* qui constituent un ordinateur, il faut aussi se pencher sur les éléments moins palpables : les logiciels.

À l'époque des tous premiers ordinateurs, chaque fois qu'il fallait exécuter des traitements différents, il fallait intervenir physiquement pour changer la disposition des cables et des composants. On en est bien loin aujourd'hui : les opérations à réaliser pour faire les traitements sont devenues des données comme les autres. Des données qu'on appelle « programmes » qui sont chargées, modifiées, manipulées par d'autres programmes.

Les programmes sont généralement écrits pour essayer de ne faire qu'une seule chose, et de la faire bien, ceci surtout pour rester compréhensibles par les êtres humains qui les conçoivent. C'est ensuite l'interaction de dizaines de milliers de programmes entre eux qui permettra de réaliser les tâches complexes pour lesquelles sont généralement utilisés les ordinateurs de nos jours.

L'effet produit lorsqu'on clique sur un bouton, c'est donc le lancement d'une chaîne d'évènements, d'une somme impressionnante de calculs, qui aboutissent à des impulsions électriques venant à la fin modifier un objet physique (comme un CD qu'on

1. Berke Durak a réussi en 1995 à capter les ondes électromagnétiques [http://lambda-diode.com/electronics/tempest/] émises par la plupart des composants de son ordinateur avec un simple *walkman* capable de recevoir la radio.
2. Martin Vuagnoux et Sylvain Pasini ont réalisé d'effrayantes vidéos [http://lasecwww.epfl.ch/keyboard/] pour illustrer leur papier *Compromising Electromagnetic Emanations of Wired and Wireless Keyboards* publié en 2009.

veut graver, un écran qui modifie ses LEDs pour afficher une nouvelle page, ou un disque dur qui active ou désactive des micro-interrupteurs pour créer la suite binaire de données qui constituera un *fichier*).

Le système d'exploitation

Le but d'un système d'exploitation est avant tout de permettre aux logiciels de se partager l'accès aux composants matériels de l'ordinateur. Son rôle est aussi de permettre aux différents logiciels de communiquer entre eux. Un système d'exploitation est par ailleurs généralement livré avec des logiciels, au minimum de quoi permettre de démarrer d'autres logiciels.

La partie la plus fondamentale d'un système d'exploitation est son noyau qui s'occupe de coordonner l'utilisation du matériel par les programmes.

Pour chaque composant matériel de l'ordinateur que l'on veut utiliser, le noyau active un programme qu'on appelle « pilote » (ou *driver* en anglais). Il existe des pilotes pour les périphériques d'entrée (comme le clavier et la souris), de sortie (écran, imprimantes, *etc.*), de stockage (CD-ROM, clé USB, *etc.*).

Le noyau gère aussi l'exécution des programmes, en leur donnant des morceaux de mémoire et en répartissant le temps de calcul du processeur entre les différents programmes qui veulent le faire travailler.

Au-delà du noyau, les systèmes d'exploitation utilisés de nos jours, comme Windows, Mac OS X ou GNU/Linux (avec Debian, Ubuntu, Fedora, par exemple) incluent aussi de nombreux utilitaires ainsi que des environnements de bureaux graphiques qui permettent d'utiliser l'ordinateur en cliquant simplement sur des boutons.

Le système d'exploitation est en général stocké sur le disque dur. Cependant, il est aussi tout à fait possible d'utiliser un système d'exploitation enregistré sur une clé USB ou gravé sur un CD-ROM. Dans ce dernier cas, on parle de système *live* (vu qu'aucune modification ne pourra être faite sur le CD).

Les applications

On appelle « applications » les logiciels qui permettent réellement de faire ce qu'on a envie de demander à l'ordinateur. On peut citer comme exemple Mozilla Firefox comme naviguateur web, OpenOffice.org pour la bureautique ou encore GIMP ou Adobe Photoshop pour le traitement d'images.

Chaque système d'exploitation définit une méthode bien spécifique pour que les applications puissent accéder au matériel, à des données, au réseau, ou à d'autres ressources. Les applications que l'on souhaite utiliser doivent donc êtres conçues pour le système d'exploitation de l'ordinateur sur lequel on veut s'en servir.

Les bibliothèques

Plutôt que de réécrire dans toutes les applications des morceaux de programme chargés de faire les mêmes choses, les logiciels se les partagent dans des bibliothèques, ou *libraries* en anglais.

Il existe des bibliothèques pour l'affichage graphique (assurant une cohérence de ce qui est affiché à l'écran), pour lire ou écrire des formats de fichiers, pour interroger certains services réseaux, *etc.*.

Si l'on n'est pas programmeur, on a rarement besoin de toucher aux bibliothèques. Il peut toutefois être intéressant de connaître leur existence, ne serait-ce que parce

qu'un problème (comme une erreur de programmation) dans une bibliothèque peut se répercuter sur tous les logiciels qui l'utilisent.

1.5 Le rangement des données

On a vu qu'un disque dur (ou une clé USB) permettait de garder des données entre deux allumages d'un ordinateur.

Mais, histoire de s'y retrouver, les données sont agencées d'une certaine manière : un meuble dans lequel on aurait simplement entassé des feuilles de papier ne constitue pas vraiment une forme de rangement des plus efficaces…

Les partitions

Tout comme dans un meuble on peut mettre plusieurs étagères, on peut « découper » un disque dur en plusieurs *partitions*.

Chaque étagère pourra avoir une hauteur différente, un classement différent, selon que l'on souhaite y mettre des livres ou des classeurs, par ordre alphabétique ou par ordre de lecture.

De la même manière, sur un disque dur, chaque partition pourra être de taille différente et contenir un mode d'organisation différent : un système de fichiers.

Les systèmes de fichiers

Un système de fichiers sert avant tout à pouvoir retrouver des informations dans notre immense pile de données, comme la table des matières d'un livre de cuisine permet directement d'aller à la bonne page pour lire la recette du festin du soir.

Il peut être important de noter que la suppression d'un fichier ne fait qu'enlever une ligne dans la table des matières. En parcourant toutes les pages, on pourra toujours retrouver notre recette, tant que la page n'aura pas été réécrite — on développera cela plus tard.

page 31

On peut imaginer des milliers de formats différents pour ranger des données, et il existe donc de nombreux systèmes de fichiers différents. On parle de *formatage* lors de la création d'un système de fichiers sur un support.

Vu que c'est le système d'exploitation qui donne aux programmes l'accès aux données, un système de fichier est souvent fortement lié à un système d'exploitation particulier.

> Pour en citer quelques-un : les type NTFS, FAT32 sont ceux employés habituellement par les systèmes d'exploitation Windows ; le type *ext* (`ext3`, `ext4`) est souvent utilisé sous GNU/Linux ; les types HFS, HFS+ et HFSX sont employés par Mac OS X.

Si le logiciel adéquat existe, il est néanmoins possible de lire un système de fichiers « étranger » au système qu'on utilise. Windows est ainsi incapable de lire une partition *ext3*, à moins d'installer un logiciel approprié.

Une des conséquences de cela, c'est qu'il peut exister sur un ordinateur donné des espaces de stockage invisibles pour l'utilisateur parce que non reconnus par le système d'exploitation (ou non accessibles pour l'utilisateur), mais qui sont pourtant bel et bien présents.

Les formats de fichiers

Les données que l'on manipule sont généralement regroupées sous forme de fichiers. Un fichier a un contenu, mais aussi un nom, un emplacement (le dossier dans lequel il se trouve), une taille, et d'autres détails selon le système de fichiers utilisé.

Mais à l'intérieur de chaque fichier, les données sont elles-mêmes organisées différemment selon leur nature et les logiciels utilisés pour les manipuler. On parle de *format* de fichier pour les différencier.

En général, on met à la fin d'un fichier un code, qu'on appelle parfois *extension*, permettant d'indiquer le format du fichier.

Quelques exemples : pour la musique, on utilisera le format MP3 ou Ogg, pour un document texte d'OpenOffice.org ce sera OpenDocument Text (ODT), pour des images, on aura le choix entre le JPEG, le PNG, le format d'Adobe Photoshop (PSD), *etc.*

Il peut être intéressant de faire la différence entre les formats *ouverts*, dont les détails sont publics, et les formats *propriétaires*, souvent conçus pour être manipulés par un logiciel bien précis.

Les formats propriétaires ont parfois été observés à la loupe pour être ouverts par d'autres logiciels, mais leur compréhension reste souvent imparfaite et assujettie à des changements d'une version à l'autre d'une application. C'est typiquement le cas pour le format de Microsoft Word, souvent appelé `.doc`.

La mémoire virtuelle (*swap*)

Normalement, toutes les données auxquelles le processeur doit accéder, et donc tous les programmes et les documents ouverts, devraient se trouver en mémoire vive. Mais pour pouvoir ouvrir plein de programmes et de documents, les systèmes d'exploitation modernes trichent : ils échangent, quand c'est nécessaire, des morceaux de mémoire vive avec un espace du disque dur dédié à cet effet. On parle alors de « mémoire virtuelle », de *swap* en anglais ou encore d'« espace d'échange ».

Le système d'exploitation fait donc sa petite cuisine pour que le processeur ait toujours dans la mémoire vive les données auquelles il veut réellement accéder. Le *swap* est ainsi un exemple d'espace de stockage auquel on ne pense pas forcément, enregistré sur le disque dur, soit sous forme d'un gros fichier contigu (sous Microsoft Windows), soit dans une partition à part (avec Linux).

On reviendra dans la partie suivante sur les problèmes que posent ces questions de format et d'espaces de stockage en termes de confidentialité des données.

Chapitre 2

Traces à tous les étages

Le fonctionnement normal d'un ordinateur laisse de nombreuses traces de ce que l'on fait dessus. Parfois, elles sont *nécessaires* à son fonctionnement. D'autres fois, ces informations sont collectées pour permettre aux logiciels d'être « plus pratiques ».

2.1 Dans la mémoire vive

On vient de voir que le premier lieu de stockage des informations sur l'ordinateur est la mémoire vive. page 11

Tant que l'ordinateur est sous tension électrique, elle contient toutes les informations dont le système a besoin. Elle conserve donc nécessairement de nombreuses traces : frappes au clavier (y compris les mots de passe), fichiers ouverts, évènements divers qui ont rythmé la phase d'éveil de l'ordinateur.

En prenant le contrôle d'un ordinateur qui est allumé, il n'est pas très difficile de lui faire cracher l'ensemble des informations contenues dans la mémoire vive, par exemple vers une clé USB ou vers un autre ordinateur à travers le réseau. Et prendre le contrôle d'un ordinateur peut être aussi simple qu'y brancher un *iPod* quand on a le dos tourné [1]. Une fois récupérées, les nombreuses informations que contient la mémoire vive sur l'ordinateur et les personnes qui l'utilisent pourront alors être exploitées…

Par ailleurs, si ces données deviennent illisibles lors de la mise hors tension, cela prend néanmoins du temps, ce qui peut suffire pour qu'une personne mal intentionnée ait le temps de récupérer ce qui s'y trouve. On appelle cela une « *cold boot attack* » : l'idée est de copier le contenu de la mémoire vive avant qu'elle ait eu le temps de s'effacer, de manière à l'exploiter par la suite. Il est même techniquement possible de porter à très basse température la mémoire d'un ordinateur fraîchement éteint — auquel cas on peut faire subsister son contenu plusieurs heures, voire plusieurs jours [2].

Cette attaque doit cependant être réalisée peu de temps après la mise hors tension. Par ailleurs, si on utilise quelques gros logiciels (par exemple en retouchant une énorme image avec Adobe Photoshop ou GIMP) avant d'éteindre son ordinateur, les traces qu'on a laissées précédemment en mémoire vive ont de fortes chances d'être recouvertes. Mais surtout, il existe des logiciels spécialement conçus pour écraser le contenu de la mémoire vive avec des données aléatoires.

1. *0wned by an iPod* [http://md.hudora.de/presentations/#firewire-pacsec] présenté à la conférence *PacSec/core04* par Maximillian Dornseif. *Hacking Computers Over USB* [http://www.schneier.com/blog/archives/2006/06/hacking_compute.html] sur *Schneier on Security*.

2. *Least We Remember: Cold Boot Attacks on Encryption Keys* [http://citp.princeton.edu/memory/] présenté au *17th USENIX Security Symposium (Sec '08)*, par J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten.

2.2 Dans la mémoire virtuelle

page 17 Comme expliqué auparavant, le système d'exploitation utilise, dans certains cas, une partie du disque dur pour venir en aide à sa mémoire vive. Ça arrive en particulier si l'ordinateur est fortement sollicité, par exemple quand on travaille sur de grosses images, mais aussi dans de nombreux autres cas, de façon peu prévisible.

La conséquence la plus gênante de ce système pourtant bien pratique, c'est que l'ordinateur va écrire sur le disque dur des informations qui se trouvent dans la mémoire vive… informations potentiellement sensibles, donc, *et qui resteront lisibles après avoir éteint l'ordinateur*.

page 34 Avec un ordinateur configuré de façon standard, il est donc illusoire de croire qu'un document lu à partir d'une clé USB, même ouvert avec un logiciel portable, ne laissera jamais de traces sur le disque dur.

2.3 Veille et hibernation

La plupart des systèmes d'exploitation permettent, depuis quelques années, de mettre un ordinateur « en pause ». C'est surtout utilisé avec les ordinateurs portables mais c'est également valable pour les ordinateurs de bureau.

Il y a deux grandes familles de « pause » : la veille et l'hibernation.

La veille

La *veille* (appelée aussi en anglais *suspend to ram* ou *suspend*) consiste à éteindre le maximum de composants de l'ordinateur tout en gardant sous tension de quoi pouvoir le rallumer rapidement.

Au minimum, la mémoire vive continuera d'être alimentée pour conserver l'intégralité des données sur lesquelles on travaillait — c'est-à-dire notamment les mots de passe et les clés de chiffrement.

page 37

Bref, un ordinateur en veille protège aussi peu l'accès aux données qu'un ordinateur allumé.

L'hibernation

L'*hibernation* ou *mise en veille prolongée*, appelée aussi en anglais *suspend to disk*, consiste à sauvegarder l'intégralité de la mémoire vive sur le disque dur pour ensuite éteindre complètement l'ordinateur. Lors de son prochain démarrage, le système d'exploitation détectera l'hibernation, re-copiera la sauvegarde vers la mémoire vive et recommençera à travailler à partir de là.

Sur les systèmes GNU/Linux, la copie de la mémoire se fait généralement dans le *swap*. Sur d'autres systèmes, ça peut être dans un gros fichier, souvent caché.

Vu que c'est le contenu de la mémoire vive qui est écrite sur le disque dur, ça veut dire que tous les programmes et documents ouverts, mots de passe, clés de chiffrement et autres, pourront être retrouvés par quiconque accèdera au disque dur. Et cela, aussi longtemps que rien n'aura été réécrit par-dessus.

page 37 Ce risque est toutefois limité par le chiffrement du disque dur : la phrase de passe sera alors nécessaire pour accéder à la sauvegarde de la mémoire vive.

2.4 Les journaux

Les systèmes d'exploitation ont une forte tendance à écrire dans leur journal de bord un historique détaillé de ce qu'ils fabriquent.

Ces journaux (aussi appelés *logs*) sont utiles au système d'exploitation pour fonctionner, et permettent de corriger des problèmes de configuration ou des *bugs*.

Cependant leur existence peut parfois être problématique. Les cas de figure existants sont nombreux, mais les quelques exemples suivants devraient être suffisants pour donner une idée de ce risque :

- sous GNU/Linux, le système garde la date, l'heure et le nom de l'utilisateur qui se connecte chaque fois qu'un ordinateur est allumé ;
- toujours sous GNU/Linux, la marque et le modèle de chaque lecteur amovible branché sont habituellement conservés ;
- sous Mac OS X, la date d'une impression et le nombre de pages sont inscrits dans les journaux ;
- sous Windows, le *moniteur d'évènements* enregistre le nom du logiciel, la date et l'heure de l'installation ou de la désinstallation d'une application.

2.5 Sauvegardes automatiques et autres listes

En plus de ces journaux, il est possible que d'autres traces de fichiers, même supprimés, subsistent sur l'ordinateur. Même si les fichiers et leur contenu ont été bien supprimés, une partie du système d'exploitation ou d'un autre programme peut en garder une trace délibérée.

Voici quelques exemples :

- sous Windows, Microsoft Office peut garder la référence d'un nom de fichier déjà supprimé dans le menu des « documents récents », et parfois même garder des fichiers temporaires avec le contenu du fichier en question ;
- sous GNU/Linux, un fichier d'historique peut contenir le nom d'un fichier préalablement supprimé. Et OpenOffice peut garder autant de traces d'un fichier supprimé que Microsoft Office. En pratique, il existe des dizaines de programmes fonctionnant ainsi ;
- lorsqu'on utilise une imprimante, le système d'exploitation copie souvent le fichier en attente dans la « file d'impression ». Le contenu de ce fichier, une fois la file vidée, n'aura pas disparu du disque dur pour autant ;
- sous Windows, lorsqu'on connecte un lecteur amovible, le système commence souvent par explorer son contenu afin de proposer des logiciels adaptés à sa lecture : cette exploration automatique laisse en mémoire la liste de tous les fichiers présents sur le support employé, même si aucun des fichiers qu'il contient n'est consulté.

Il est difficile de trouver une solution adéquate à ce problème. Un fichier, même parfaitement supprimé, continuera probablement à exister sur l'ordinateur pendant un certain temps sous une forme différente. Une recherche sur les données brutes du disque permettrait de voir si des copies de ces données existent ou pas… sauf si elles y sont seulement référencées, ou stockées sous une forme différente ; sous forme compressée, par exemple.

En fait, seul l'écrasement de la totalité du disque et l'installation d'un nouveau système d'exploitation permettent d'avoir la garantie que les traces d'un fichier ont bien été supprimées. Et dans une autre perspective, l'utilisation d'un système *live*, dont l'équipe de developpement porte une attention particulière à cette question, garantit que ces traces ne seront pas laissées ailleurs que dans la mémoire vive.

2.6 Les méta-données

Autour des informations contenues dans un fichier, il existe des informations sur ce contenu. Ces « données sur les données » s'appellent communément des « méta-données ».

page 16 Une partie des méta-données est enregistrée par le système de fichiers : le nom du fichier, la date et l'heure de création et de modification, et souvent bien d'autres choses.

page 17 Mais de nombreux formats de fichiers conservent également des méta-données à l'intérieur du fichier. Elles pourront donc être connues de quiconque aura accès au fichier.

Les méta-données enregistrées dépendent des formats et des logiciels utilisés. La plupart des fichiers audio permettent d'y enregistrer le titre du morceau et l'interprète. Les traitements de texte ou les PDFs enregistreront un nom d'auteur, la date et l'heure de création, et parfois même l'historique des dernières modifications…

La palme revient probablement aux formats d'images comme TIFF ou JPEG : ces fichiers de photo créés par un appareil numérique ou un téléphone portable contiennent un standard de méta-données appelé EXIF. Ce dernier peut contenir la date et l'heure de la prise de vue, la marque, le modèle et le numéro de série de l'appareil utilisé, ainsi qu'une version miniature de l'image. Et toutes ces informations ont tendance à rester après être passées par un logiciel de traitement d'image. Le cas de la miniature est particulièrement intéressant : de nombreuses photos disponibles sur Internet contiennent encore l'intégralité d'une photo recadrée… et des visages ayant été « floutés ». [3]

Pour la plupart des formats de fichiers ouverts, il existe toutefois des logiciels pour examiner et éventuellement supprimer les méta-données.

3. Maximillian Dornseif et Steven J. Murdoch, *Hidden Data in Internet Published Documents* [http://md.hudora.de/presentations/#hiddendata-21c3] présenté au *21C3*.

Chapitre 3

Logiciels malveillants, mouchards et autres espions

Au-delà des traces que le fonctionnement de tout système d'exploitation laisse au moins le temps où l'ordinateur fonctionne, on peut aussi trouver dans nos ordinateurs tout un tas de *mouchards*. Soit installés à notre insu (permettant par exemple de détourner les journaux vers d'autres fins), soit présents de manière systématique dans les logiciels qu'on aura installés. page 21

Ces mouchards peuvent participer à diverses techniques de surveillance, de la « lutte » contre le « piratage » de logiciels propriétaires, au fichage ciblé d'un individu, en passant par la collecte de données pour des pourriels (*spam*) ou autres arnaques.

La portée de ces dispositifs augmente fortement dès que l'ordinateur est connecté à Internet. Leur installation est alors grandement facilitée si on ne fait rien de spécial pour se protéger, et la récupération des données collectées se fait à distance.

Toutefois les gens qui récoltent ces informations sont inégalement dangereux : ça dépend des cas, de leurs motivations et de leurs moyens. Les sites Internet à la recherche de consommateurs à cibler, les multinationales comme Microsoft, les gendarmes de Saint-Tropez, ou la *National Security Agency* américaine… autant de structures souvent en concurrence entre elles et ne formant pas une totalité cohérente.

Pour s'introduire dans nos ordinateurs, ils n'ont pas accès aux mêmes passe-partout, et ne savent pas tous manipuler le pied-de-biche aussi bien : par exemple, l'espionnage industriel est une des raisons importantes de la surveillance plus ou moins légale [1], et il ne faut pas croire que Microsoft donne toutes les astuces de Windows à la police française.

1. Pour se faire une idée des problématiques liées à l'espionnage industriel, lire le dossier *Spy games* [http://www.brefonline.com/numeroERA_affichearticle.asp?idA=1886], *Bref Rhônes-Alpes*, février 2004.

Cependant, à l'heure où nous écrivons (printemps 2010), les services de sécurité français sont en passe de disposer des moyens de mettre en place une surveillance informatique très complète en toute légalité, en s'appuyant sur plusieurs « mouchards » présentés par la suite. Si les aspects techniques du système semblent encore en cours d'élaboration, ses aspects juridiques sont eux en cours de finalisation, à travers la Loi d'Orientation et de Programmation pour la Performance de la Sécurité Intérieure (LOPPSI). Ce texte inclut en effet des dispositions légales inédites permettant, dans le cadre d'une enquête, d'installer ces mouchards sur l'ordinateur personnel de n'importe quel suspect, sans nécéssairement disposer d'un accès physique à la machine. [2]

3.1 Les logiciels malveillants

Les logiciels malveillants [3] (que l'on appelle également *malwares*) sont des logiciels qui ont été développés dans le but de nuire : collecte d'informations, hébergement d'informations illégales, relai de pourriel *etc.* Les virus informatiques, les vers, les chevaux de Troie, les *spyware*, les *rootkits* (logiciels permettant de prendre le contrôle d'un ordinateur) et les *keyloggers* sont de cette engence. Certains programmes peuvent appartenir à plusieurs de ces catégories simultanément.

page 26

Afin de s'installer sur un ordinateur, certains logiciels malveillants exploitent les vulnérabilités du système d'exploitation [4] ou des applications. Ils s'appuient sur des erreurs de conception ou de programmation pour détourner le déroulement des programmes à leur avantage. Malheureusement, de telles « failles de sécurité » ont été trouvées dans de très nombreux logiciels, et de nouvelles sont trouvées constamment, tant par des gens qui cherchent à les corriger que par d'autres qui cherchent à les exploiter.

Un autre moyen courant est d'inciter la personne utilisant l'ordinateur à lancer le logiciel malveillant en le cachant dans un logiciel en apparence inoffensif. L'attaquant n'est alors pas obligé de trouver des vulnérabilités sérieuses dans des logiciels courants. Il est particulièrement difficile de s'assurer que des ordinateurs partagés par de nombreuses personnes ou des ordinateurs qui se trouvent dans des lieux publics, comme une bibliothèque ou un cybercafé, n'ont pas été corrompus : il suffit en effet qu'une seule personne un peu moins vigilante se soit faite avoir…

En outre, la plupart des logiciels malveillants « sérieux » ne laissent pas de signe immédiatement visible de leur présence, et peuvent même être très difficiles à détecter.

2. Pour plus de détails, nous recommandons la lecture de deux articles publiés sur PCInpact : *LOPPSI : la police sera autorisée à installer des chevaux de Troie* [http://www.pcinpact.com/actu/news/51027-police-opj-cheval-troie-loppsi.htm] et *Les chevaux de Troie de la police seront installables à distance* [http://www.pcinpact.com/actu/news/51077-loppsi-chevaux-troie-police-distance.htm].

3. Toute cette partie est grandement inspirée du passage consacré à la question dans le *Surveillance Self-Defense Guide* [https://ssd.eff.org/tech/malware] de l'*Electronic Frontier Foundation*.

4. D'après l'*Internet Storm Center* [http://isc.sans.org/survivaltime.html], une installation de Microsoft Windows sur laquelle les mises à jour de sécurité n'ont pas été faites se fait compromettre en moins de 4 minutes si elle est connectée directement à Internet.

En 2006, Joanna Rutkowska a présenté lors de la conférence *Black Hat* le *malware* nommé « Blue Pill ». Cette démonstration a montré qu'il était possible d'écrire un *rootkit* utilisant les technologies de virtualisation pour tromper le système d'exploitation et rendre ainsi vraiment très difficile d'identifier la présence du *malware*, une fois celui-ci chargé.

Ces logiciels peuvent voler les mots de passe, lire les documents stockés sur l'ordinateur (même les documents chiffrés, s'ils ont été déchiffrés à un moment), réduire à néant des dispositifs d'anonymat sur Internet, prendre des captures d'écran du bureau et se cacher eux-mêmes des autres programmes. Ils peuvent parfois utiliser le micro, la webcam ou d'autres périphériques de l'ordinateur. Il existe même un marché noir où l'on peut acheter de tels programmes, personnalisés pour différents objectifs.

Toutefois, il est beaucoup plus courant que ces logiciels travaillent à obtenir des numéros de cartes bancaires, des mots de passe de compte *eBay* ou de banques en ligne, à envoyer des pourriels ou à participer à attaquer un serveur en le saturant de demandes, plutôt qu'à espionner des organisations ou des individus spécifiques. Une infection lancée par des flics est néanmoins possible, même si elle nécessite la mise en œuvre de moyens coûteux et reste en général liée à une enquête particulière.

Pour donner un exemple venu des États-Unis, le FBI a écrit un logiciel nommé CIPAV pour *Computer and Internet Protocol Address Verifier*. Ce dernier a permis par exemple d'identifier un adolescent de quinze ans ayant envoyé par email des menaces d'attentat contre un lycée de Washington [5].

Personne ne sait combien d'ordinateurs sont infectés par des logiciels malveillants, mais certains estiment que c'est le cas pour **40 à 90 %** des installations de Windows. Il est donc fort probable d'en trouver sur le premier Windows que l'on croisera. Jusqu'à présent, utiliser un système d'exploitation minoritaire (tel Mac OS X ou GNU/Linux) diminue significativement les risques d'infection car ceux-ci sont moins visés, le développement de *malwares* spécifiques étant économiquement moins rentable.

On peut d'ores et déjà évoquer quelques moyens de limiter les risques :

- n'installer (ou n'utiliser) aucun logiciel de provenance inconnue : ne pas faire confiance au premier site web venu [6] ;
- prendre au sérieux les avertissements des systèmes d'exploitation récents qui tentent de prévenir les utilisateurs lorsqu'ils utilisent un logiciel peu sûr, ou lorsqu'ils indiquent qu'une mise à jour de sécurité est nécessaire ;
- enfin, limiter les possibilités d'installation de nouveaux logiciels : en limitant l'utilisation du compte « administrateur » et le nombre de personnes y ayant accès.

5. Source : *FBI's Secret Spyware Tracks Down Teen Who Made Bomb Threats* [http://www.wired.com/politics/law/news/2007/07/fbi_spyware], *Wired*, juillet 2007.

6. Ce conseil vaut tout autant pour les personnes utilisant GNU/Linux. En décembre 2009, le site gnome-look.org a diffusé un *malware* [http://lwn.net/Articles/367874/] présenté comme un économiseur d'écran. Ce dernier était téléchargeable sous forme de paquet Debian au milieu d'autres économiseurs et de fonds d'écran.

3.2 Les *keyloggers*, ou enregistreurs de frappe au clavier

Les enregistreurs de frappe au clavier (*keyloggers*), qui peuvent être « matériels » ou « logiciels », ont pour fonction d'enregistrer furtivement tout ce qui est tapé sur un clavier d'ordinateur, afin de pouvoir transmettre ces données à l'agence ou à la personne qui les a installés [7].

Leur capacité à enregistrer touche par touche ce qui est tapé sur un clavier, contournant ainsi tout dispositif de chiffrement, permet d'avoir directement accès aux phrases, mots de passe et autres données sensibles entrées lorsqu'il y a un enregistreur de frappe sur un clavier.

Les *keyloggers* matériels sont des dispositifs reliés au clavier ou à l'ordinateur. Ils peuvent ressembler à des adaptateurs, à des cartes d'extension à l'intérieur de l'ordinateur (PCI ou *mini*-PCI) et même s'intégrer à l'intérieur du clavier [8]. Ils sont donc difficiles à repérer si on ne les recherche pas spécifiquement…

Pour un clavier sans fil, il n'y a même pas besoin de *keylogger* pour récupérer les touches entrées : il suffit de capter les ondes émises par le clavier pour communiquer avec le récepteur, puis de casser le chiffrement utilisé, qui est assez faible dans la plupart des cas [9]. À moindre distance, il est aussi toujours possible d'enregistrer et de décoder les ondes électromagnétiques émises par les claviers avec un fil, y compris ceux qui sont intégrés dans un ordinateur portable…

page 14

Les *keyloggers* logiciels sont beaucoup plus répandus, parce qu'ils peuvent être installés à distance (via un réseau ou par le biais d'un logiciel malveillant par exemple), et ne nécessitent généralement pas un accès physique à la machine pour la récupération des données collectées (l'envoi peut par exemple se faire périodiquement par email). La plupart de ces logiciels enregistrent également le nom de l'application en cours, la date et l'heure à laquelle elle a été exécutée ainsi que les frappes de touches associées à cette application.

Aux États-Unis, le FBI utilise depuis de nombreuses années des *keyloggers* logiciels [10].

La seule manière de repérer les *keyloggers* matériels est de se familiariser avec ces dispositifs et de faire régulièrement une vérification visuelle de sa machine, à l'intérieur et à l'extérieur. Pour les *keyloggers* logiciels, les pistes sont les mêmes que pour les autres *malware*.

page 24

3.3 Des problèmes d'impression ?

On croyait avoir fait le tour des surprise que nous réservent nos ordinateurs… mais même les imprimantes se mettent à avoir leurs petits secrets.

Un peu de stéganographie

Première chose à savoir : de nombreuses imprimantes haut de gamme signent leur travail. Cette signature stéganographique [11] repose sur de très légers détails

7. Source : *Ordinateur & Sécurité Internet, Vie Privée, Anonymat et cætera* [http://www.bugbrother.com/security.tao.ca/keylog.html] Adaptation française du site http://security.resist.ca.
8. Pour se faire une idée, nombre de modèles sont en vente libre [http://www.google.com/products?q=keyloggers] pour une somme allant de 40 à 100 $.
9. Source : *Microsoft wireless keyboard hacked from 50 metres* [http://www.zdnet.com.au/news/security/soa/Microsoft-wireless-keyboard-hacked-from-50-metres/0,130061744,339284328,00.htm] paru sur *ZDNet Australia*, décembre 2007.
10. En 2000, l'usage d'un *keylogger* a permis au FBI [http://www.theregister.co.uk/2000/12/06/mafia_trial_to_test_fbi/] d'obtenir la phrase de passe utilisée par un ponte de la mafia de Philadelphie pour chiffrer ses documents.
11. Pour en savoir plus sur la stéganographie, nous conseillons la lecture de l'article de Wikipédia [https://secure.wikimedia.org/wikipedia/fr/wiki/Stéganographie] qui lui est consacré.

d'impression, souvent invisibles à l'œil nu, et insérés dans chaque document. Ils permettent d'identifier de manière certaine la marque, le modèle et dans certains cas le numéro de série de la machine qui a servi à imprimer un document. On dit bien « de manière certaine », car c'est pour cela que ces détails sont là : afin de pouvoir retrouver la machine à partir de ses travaux. Toutes les imprimantes ne sont pas pourvues de ce système, baptisé *watermarking*, mais c'est le cas pour nombre de modèles courants [12].

Par ailleurs, d'autres types de traces liées à l'usure de la machine sont aussi laissées sur les documents — et ce avec toutes les imprimantes. Car avec l'âge, les têtes d'impression se décalent, de légères erreurs apparaissent, les pièces s'usent, et tout cela constitue au fur et à mesure une signature propre à l'imprimante. Tout comme la balistique permet d'identifier une arme à feu à partir d'une balle, il est possible d'utiliser ces défauts pour identifier une imprimante à partir d'une page qui en est sortie.

Pour se protéger en partie de cela, il est intéressant de savoir que les détails d'impression ne résistent pas à la photocopie répétée : photocopier la page imprimée, puis photocopier la photocopie obtenue, suffit à faire disparaître de telles signatures. Par contre… on en laissera sûrement d'autres, les photocopieuses présentant des défauts, et parfois des signatures stéganographiques, similaires à ceux des imprimantes. Bref on tourne en rond, et le problème devient surtout de choisir *quelles* traces on veut laisser…

La mémoire, encore...

Certaines imprimantes sont suffisamment « évoluées » pour être plus proches d'un véritable ordinateur que d'un tampon encreur.

Elles peuvent poser des problèmes à un autre niveau, vu qu'elles sont dotées d'une mémoire vive : celle-ci, tout comme celle du PC, gardera la trace des documents qui ont été traités aussi longtemps que la machine est sous tension… ou qu'un autre document les recouvre.

page 11

La plupart des imprimantes lasers disposent d'une mémoire vive pouvant contenir une dizaine de pages. Les modèles plus récents ou ceux comportant des scanners intégrés peuvent, quant à eux, contenir plusieurs milliers de pages de texte…

Pire encore : certains modèles, souvent utilisés pour les gros tirages comme dans les centres de photocopies, disposent parfois de disques durs internes, auxquels l'utilisateur n'a pas accès, et qui gardent eux aussi des traces — et cette fois, même après la mise hors tension.

12. L'*Electronic Frontier Foundation* tente de maintenir une liste des constructeurs et de ces modèles [http://www.eff.org/issues/printers] d'imprimantes indiscrets.

Chapitre 4

Quelques illusions de sécurité...

Bien. On commence à avoir fait le tour des traces que nous pouvons laisser involontairement, et des informations que des personnes mal intentionnées pourraient nous arracher.

Reste maintenant à pourfendre quelques idées reçues.

4.1 Logiciels propriétaires, *open source*, libres

On a vu qu'un logiciel pouvait faire plein de choses qu'on n'aurait pas du tout envie qu'il fasse. Dès lors, il est indispensable de faire ce que l'on peut pour réduire ce problème autant que possible. De ce point de vue, les logiciels libres sont dignes d'une confiance bien plus grande que les logiciels dits « propriétaires » : nous allons voir pourquoi.

La métaphore du gâteau

Pour comprendre la différence entre ces deux types de logiciels, on utilise souvent la métaphore du gâteau. Pour faire un gâteau, il faut une recette : il s'agit d'une liste d'instructions à suivre, des ingrédients à utiliser et d'un procédé de transformation à effectuer. De la même façon, la recette d'un logiciel est appelée « code source ». Elle est écrite dans un langage fait pour être compréhensible par des êtres humains. Cette recette est ensuite transformée en un code compréhensible par le processeur, un peu comme la cuisson d'un gâteau nous donne ensuite la possibilité de le manger.

Les logiciels propriétaires ne sont disponibles que « prêts à consommer », comme un gâteau industriel, sans sa recette. Il est donc très difficile de s'assurer de ses ingrédients : c'est faisable, mais le processus est long et compliqué. Au demeurant, relire une série de plusieurs millions d'additions, de soustractions, de lectures et d'écritures en mémoire pour en reconstituer le but et le fonctionnement est loin d'être la première chose que l'on souhaite faire sur un ordinateur.

Les logiciels libres, au contraire, livrent la recette pour quiconque veut comprendre ou modifier le fonctionnement du programme. Il est donc plus facile de savoir ce qu'on donne à manger à notre processeur, et donc ce qui va s'occuper de nos données.

Les logiciels propriétaires : une confiance aveugle

Un logiciel « propriétaire » est donc un peu comme une « boîte » étanche : on peut constater que le logiciel fait ce qu'on lui demande, possède une belle interface graphique, *etc.* Sauf qu'on ne peut pas vraiment connaître en détail comment il procède !

On ne sait pas s'il se cantonne à faire ce qu'on lui demande, ou s'il fait d'autres choses en plus. Pour le savoir, il faudrait pouvoir étudier son fonctionnement, ce qui est difficile à faire sans son code source… il ne nous reste donc qu'à lui faire *aveuglément* confiance.

Windows et Mac OS X, les premiers, sont d'immenses boîtes hermétiquement fermées sur lesquelles sont installées d'autres boîtes tout aussi hermétiques (de Microsoft Office aux anti-virus…) qui font peut-être bien d'autres choses que celles qu'on leur demande.

Notamment, balancer des informations que ces logiciels pourraient grapiller sur nous ou permettre d'accéder à l'intérieur de notre ordinateur au moyen de *backdoors*, des « portes dérobées » [1] prévues dans le logiciel pour que ceux qui en ont la clé puissent pirater nos ordinateurs… En fait, vu que l'on ne peut pas savoir comment est écrit le système d'exploitation, on peut tout imaginer en la matière.

Dès lors, laisser reposer la confidentialité et l'intégrité de ses données sur des programmes auxquels on ne peut accorder sa confiance que les yeux fermés, relève de la plus pure illusion de sécurité. Et installer d'autres logiciels prétendant sur leur emballage veiller à cette sécurité à notre place, alors que leur fonctionnement n'est pas plus transparent, ne peut pas résoudre ce problème.

L'avantage d'avoir la recette : les logiciels libres

La confiance plus grande qu'on peut mettre dans un système *libre* comme GNU/Linux est principalement liée au fait de disposer de la « recette » qui permet de le fabriquer. Gardons en tête quand même qu'il n'y a rien de magique : les logiciels libres ne jetent aucun « sort de protection » sur nos ordinateurs.

Toutefois, GNU/Linux offre davantage de possibilités pour rendre un peu plus sûr l'usage des ordinateurs, notamment en permettant de configurer assez finement le système. Ça implique trop souvent des savoirs-faire relativement spécialisés, mais au moins c'est possible.

Par ailleurs, le mode de production des logiciels libres est peu compatible avec l'introduction de portes dérobées : c'est un mode de production collectif, plutôt ouvert et transparent, auquel participent des gens assez variés ; il n'est donc pas facile d'y mettre en toute discrétion des cadeaux à l'attention de personnes mal intentionnées.

Il faut toutefois se méfier des logiciels qualifiés d'*open source*. Ces derniers donnent eux aussi accès à leurs entrailles, mais ont des modes de développement plus fermés, plus opaques. La modification et la redistribution de ces logiciels est au pire interdite, et au mieux autorisée formellement mais rendue en pratique très pénible. Vu que seule l'équipe à l'origine du logiciel va pouvoir participer au développement, on peut considérer que, en pratique, personne ne lira en détail leur code source… et donc que personne ne vérifiera vraiment leur fonctionnement.

C'est le cas par exemple de TrueCrypt, un logiciel de chiffrement dont le code source est disponible, mais dont le développement est fermé et dont la licence restreint la modification et la redistribution. Pour ce qui nous occupe, le fait qu'un logiciel soit *open source* doit plutôt être considéré comme un argument commercial que comme un gage de confiance.

Sauf que… la distinction entre logiciels libres et *open source* est de plus en plus floue : des employés d'IBM et compagnie écrivent de grosses parties des logiciels libres les plus importants, et on ne va pas toujours regarder de près ce qu'ils écrivent. Par exemple, voici les statistiques des employeurs des gens qui développent le noyau Linux (qui

1. Au sujet des « portes dérobées » voir l'article de Wikipédia [https://secure.wikimedia.org/wikipedia/fr/wiki/Porte_dérobée].

est libre), exprimées en nombre de lignes de code source modifiées, sur une courte période de temps [2] :

Organisation	Pourcentage
(aucun)	18,6 %
Novell	16,9 %
Red Hat	9,9 %
Broadcom	5,6 %
Intel	5,2 %
(inconnu)	5,1 %
Google	2,7 %
IBM	2,0 %
Nokia	1,6 %
Microsoft	1,3 %
etc.	

Alors… il n'est pas impossible qu'une personne qui écrit un bout de logiciel dans un coin, et à qui la « communauté du libre » fait confiance, ait pu y glisser des bouts de code mal intentionné. Si on utilise uniquement des logiciels libres livrés par une distribution GNU/Linux non commerciale, il y a peu de chances que ce cas se présente, mais c'est une possibilité. On fait alors confiance aux personnes travaillant sur la distribution pour étudier le fonctionnement des programmes qui y sont intégrés.

Il est néanmoins important de rappeler que cette confiance ne peut valoir que si on n'installe pas n'importe quoi sur son système. Par exemple, sur Debian, les paquets officiels de la distribution sont « signés », ce qui permet de vérifier leur provenance. Mais si on installe des paquets ou des extensions pour Firefox trouvés sur Internet sans les vérifier, on s'expose à tous les risques mentionnés au sujet des logiciels malveillants. page 24

Pour conclure, et ne pas nous faire plus d'illusions : *libres ou pas, il n'existe pas de logiciel pouvant, à lui seul, assurer l'intimité de nos données* ; pour le faire, il n'existe que des pratiques, associées à l'utilisation de *certains logiciels*. Logiciels choisis parce que des éléments nous permettent de leur accorder un certain niveau de confiance.

4.2 Le mot de passe d'un compte ne protège pas ses données

Tous les systèmes d'exploitation récents (Windows, Mac OS X, GNU/Linux) offrent la possibilité d'avoir différents utilisateurs sur un même ordinateur. Il faut bien savoir que les mots de passe qui protègent parfois ces utilisateurs ne garantissent pas du tout la confidentialité des données.

Certes il peut être pratique d'avoir son espace à soi, avec ses propres réglages (marque-pages, fond d'écran…), mais une personne qui souhaiterait avoir accès à toutes les données qu'il y a sur l'ordinateur n'aurait aucun mal à y parvenir : il suffit de rebrancher le disque dur sur un autre ordinateur ou de le démarrer sur un autre système d'exploitation pour avoir accès à toutes les données écrites sur le disque dur. page 15

Aussi, si utiliser des comptes séparés et des mots de passe peut avoir quelques avantages (comme la possibilité de verrouiller l'écran quand on s'éloigne quelques minutes), il est nécessaire de garder en tête que cela ne protège pas réellement les données.

4.3 À propos de l'« effacement » des fichiers

On a déjà évoqué que le contenu d'un fichier devenu inaccessible ou invisible ne s'était pas pour autant volatilisé. On va maintenant détailler pourquoi. page 16

2. Source : *Who wrote 2.6.32* [http://lwn.net/Articles/363456/], *Linux Weekly News*, 24 novembre 2009.

La suppression d'un fichier n'en supprime pas le contenu…

… et ça peut être très facile de le retrouver.

En effet, lorsqu'on « supprime » un fichier — en le plaçant par exemple dans la *Corbeille* puis en la vidant — on ne fait que dire au système d'exploitation que le contenu de ce fichier ne nous intéresse plus. Il supprime alors son entrée dans l'index des fichiers existants. Il a ensuite le loisir de réutiliser l'espace que prenaient ces données pour y inscrire autre chose.

Mais il faudra peut-être des semaines, des mois ou des années avant que cet espace soit *effectivement* utilisé pour de nouveaux fichiers, et que les anciennes données disparaissent réellement. En attendant, si on regarde directement ce qui est inscrit sur le disque dur, on retrouve le contenu des fichiers. C'est une manipulation assez simple, automatisée par de nombreux logiciels (qui permettent de « récupérer » ou de « restaurer » des données).

Un début de solution : réécrire plusieurs fois par-dessus les données

Une fois que l'espace d'un disque dur a été réécrit, il devient diffile de retrouver ce qui s'y trouvait auparavant. Mais cela n'est pas pour autant impossible : lorsque l'ordinateur réécrit `1` par-dessus `0`, cela donne plutôt `0,95` et lorsqu'il réécrit `1` par-dessus `1`, cela donne plutôt `1,05`[3]… un peu comme on peut lire sur un bloc-notes ce qui a été écrit sur une page arrachée, par les dépressions créées sur la page vierge située en-dessous.

En revanche ça devient très difficile, voire impossible, de les récupérer quand on réécrit un grand nombre de fois par-dessus, et de différentes manières. La meilleure façon, donc, de rendre inaccessible le contenu de ces fichiers « supprimés », est d'utiliser des logiciels qui s'assureront de le réécrire plusieurs fois, pour terminer par du charabia incompréhensible.

Quelques limites des possibilités de réécriture

Même s'il est possible de réécrire plusieurs fois à un endroit donné d'un disque dur pour rendre inaccessibles les données qu'il contenait, cela ne garantit pas pour autant leur disparition complète du disque…

Les disques « intelligents »

Les disques durs modernes réorganisent leur contenu « intelligemment » : une partie du disque est reservée pour remplacer des endroits qui deviendraient défectueux. Ces opérations de remplacement sont difficilement détectables, et on ne peut jamais être vraiment sûr que l'endroit sur lequel on réécrit trente fois est bien celui où le fichier a été écrit initialement…

Pour les clés USB, on est même sûr que dans la plupart des cas on réécrit à un endroit différent. Comme la mémoire *flash*, utilisée par les clés USB et les disques durs SSD (*Solid State Disks*), arrête de fonctionner correctement après un certain nombre

3. Source : Secure Deletion of Data from Magnetic and Solid-State Memory [`http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html`] par Peter Gutmann, présenté au *6ème USENIX Security Symposium* en 1996.

d'écritures [4], ces derniers contiennent des puces chargées de réorganiser automatiquement le contenu pour répartir les informations au maximum d'endroits différents.

En prenant en compte ces mécanismes, il devient difficile de garantir que les données que l'on souhaite détruire auront bien disparu. Néanmoins, ouvrir un disque dur pour en examiner les entrailles ou recâbler les puces d'une clé USB demande du temps et d'importantes ressources matérielles et humaines… investissement qui ne sera pas forcément à la portée de tout le monde, tout le temps.

Les systèmes de fichiers « intelligents »

Un autre problème vient des systèmes de fichiers « intelligents ». page 16

Les systèmes de fichiers développés ces dernières années, comme NTFS ou *ext3*, sont « journalisés ». C'est-à-dire qu'ils gardent une trace des modifications successives faites sur les fichiers dans un « journal ». Après une extinction brutale de l'ordinateur, cela permet au système de se contenter de reprendre les dernières opérations à faire, plutôt que de devoir parcourir l'intégralité du disque pour corriger les incohérences. Par contre, cela peut ajouter, encore une fois, des traces sur les fichiers que l'on souhaiterait voir disparaître. page 21

> Le système de fichiers utilisé actuellement le plus souvent sous GNU/Linux, *ext3*, peut fonctionner avec plusieurs modes. Celui le plus courament utilisé ne met dans le journal que les noms des fichiers et d'autres méta-données, pas leur contenu.

D'autres techniques, moins courantes sur un ordinateur personnel, peuvent aussi poser problème : les systèmes de fichiers avec écriture redondante et continuant à écrire même en cas d'erreur, comme les systèmes de fichiers RAID ; les systèmes de fichiers qui effectuent des instantanés (*snapshots*) ; les systèmes de fichiers qui mettent en cache dans des dossiers temporaires, comme les clients NFS (système de fichiers par le réseau) ; les systèmes de fichiers compressés [5].

Enfin, il ne faut pas oublier que le fichier, même parfaitement supprimé, peut avoir laissé des traces ailleurs… page 19

Ce qu'on ne sait pas...

Pour ce qui est des CD-RW ou DVD±RW (ré-inscriptibles), il semble qu'aucune étude sérieuse n'ait été menée à propos de l'efficacité de la réécriture pour rendre des données irrécupérables. Un postulat prudent est donc de détruire méthodiquement les supports de ce type qui auraient pu contenir des données à faire disparaître.

Plein d'autres fois où l'on « efface »

Il faut noter qu'on ne supprime pas seulement des fichiers en les « mettant à la corbeille ». Par exemple, quand on utilise l'option « Effacer mes traces » du navigateur Firefox, ce dernier ne fait pas mieux que de *supprimer* les fichiers. Certes les données sont devenues inaccessibles pour Firefox, mais elles sont toujours accessibles en regardant directement le disque dur.

4. Les modèles bas de gamme ne fonctionneront plus correctement après avoir été écrits cent mille fois, et cinq millions pour les meilleurs ; d'après Wikipédia [https://secure.wikimedia.org/wikipedia/fr/wiki/Solid_State_Drive].

5. Source : page de manuel de `shred(1)` [http://manpages.debian.net/cgi-bin/man.cgi?query=shred&locale=fr].

Enfin, il est utile d'insister ici sur le fait que le *reformatage* d'un disque dur n'efface pas pour autant le contenu qui s'y trouvait. Tout comme la suppression des fichiers, cela ne fait que rendre disponible l'espace où se trouvait le contenu précédemment, les données restant physiquement présentes sur le disque. Tout comme détruire le catalogue d'une bibliothèque ne fait pas pour autant disparaître les livres présents dans les rayonnages…

On peut donc toujours retrouver des fichiers après un reformatage, aussi facilement que s'ils avaient été simplement « supprimés »…

Et pour ne laisser aucune trace ?

Pour régler radicalement le problème, il n'y a pas de méthode simple. La solution la moins difficile pour l'instant est d'utiliser l'ordinateur après l'avoir démarré avec un système *live* configuré pour n'utiliser que la mémoire vive. Alors, il est possible de ne rien écrire sur le disque dur ni sur la *swap*, et de ne garder les informations (tant que l'ordinateur est allumé) que dans la mémoire vive.

page 11

4.4 Les logiciels portables : une fausse solution

Ce qu'on appelle « logiciels portables », ce sont des logiciels qui ne sont pas installés sur un système d'exploitation donné, mais que l'on peut démarrer depuis une clé USB ou un disque dur externe — et donc, transporter avec soi afin d'en disposer sur n'importe quel ordinateur.

Il est devenu très facile de télécharger sur Internet de telles applications. Des « packs portables » ont ainsi été mis en ligne, comme Firefox avec Tor, ou Thunderbird avec Enigmail.

Toutefois, contrairement aux systèmes *live*, ils se servent du système d'exploitation installé sur l'ordinateur où on les utilise (la plupart du temps, ils sont prévus pour Windows).

L'idée qui est à leur origine est de permettre d'avoir toujours les logiciels dont on a besoin, sous la main, personnalisés à notre usage. Mais « transporter son bureau partout avec soi », par exemple, n'est pas forcément la meilleure manière de préserver la confidentialité de ses données.

Disons-le tout de suite : ces logiciels ne protègent pas plus les personnes qui s'en servent que des logiciels « non portables ». Pire, le discours faisant leur promotion participe à créer une illusion de sécurité avec d'énormes bêtises comme « *vous conservez toutes vos données sur votre clé et personne ne peut voir les sites que vous visitez, ni lire vos mails.* » [6] C'est malheureusement faux.

Principaux problèmes

Ces solutions « clé en main » posent donc quelques problèmes plutôt fâcheux…

6. Cet extrait provient des premières versions du texte de présentation de la FramaKey [http://forum.framasoft.org/viewtopic.php?t=8359], une compilation de logiciels portables réalisée par Framasoft [http://www.framasoft.net/], un site français de promotion du logiciel libre. Sur la nouvelle présentation de la FramaKey [http://www.framakey.org/], on peut lire maintenant « *le navigateur web et le client mail protègeront votre intimité et l'ordinateur hôte en laissant un minimum de traces* »… sans plus de précisions sur la nature de ces traces.

Il restera des traces sur le disque dur

Si le logiciel a été rendu « portable » correctement, il ne devrait pas laisser délibérément de traces sur le disque dur de l'ordinateur sur lequel on l'utilise. Mais en fait, le logiciel n'a jamais un contrôle absolu. Il dépend en effet largement du système d'exploitation sur lequel il est employé, qui peut avoir besoin d'écrire de la « mémoire virtuelle » sur le disque dur, ou d'enregistrer diverses traces de ce qu'il fait dans ses journaux et autres « documents récents ». Tout cela restera ensuite sur le disque dur.

page 20
page 21
page 21

Il n'y a aucune raison d'avoir confiance en un système inconnu

On a vu auparavant que beaucoup de systèmes ne faisaient absolument pas ce que l'on croit. Or, puisque le logiciel portable va utiliser le système installé sur l'ordinateur sur lequel on le lance, on souffrira de tous les mouchards et autres logiciels malveillants qui pourraient s'y trouver…

page 23

On ne sait pas qui les a compilés, ni comment

Les modifications apportées aux logiciels pour les rendre portables sont rarement vérifiées, alors même qu'elles ne sont généralement pas faites par les auteurs du logiciel lui-même. Dès lors, on peut soupçonner ces logiciels, encore plus que leurs versions non-portables, de contenir des failles de sécurité, qu'elles aient été introduites par erreur ou volontairement.

On traitera plus loin de la question de l'hygiène minimale à avoir dans le choix des logiciels qu'on installe ou télécharge.

Chapitre 5

Une piste pour se protéger : la cryptographie

La *cryptographie* est la branche des mathématiques qui s'occupe spécifiquement de protéger des messages. Jusqu'en 1999, l'usage de techniques cryptographiques était interdit au grand public. C'est devenu légal entre autres pour permettre aux services marchands sur Internet de se faire payer sans que les clients se fassent piquer leur carte bleue.

La *cryptanalyse* est le domaine consistant à « casser » les techniques cryptographiques, par exemple pour permettre de retrouver un message qui avait été protégé [1].

Lorsque l'on veut protéger des messages, on distingue trois aspects :

- confidentialité : empêcher les regards indiscrets ;
- authenticité : s'assurer de la source du message ;
- intégrité : s'assurer que le message n'a pas subi de modification.

On peut désirer ces trois choses en même temps, mais on peut aussi vouloir seulement l'une ou l'autre. L'émetteur d'un message *confidentiel* peut souhaiter nier en être l'auteur (et donc qu'on ne puisse pas l'*authentifier*). On peut aussi imaginer vouloir certifier la provenance (*authentifier*) et l'*intégrité* d'un communiqué officiel qui sera diffusé publiquement (donc loin d'être *confidentiel*).

Dans tout ce qui suit, on va parler de *messages*, mais les techniques cryptographiques s'appliquent de fait à n'importe quels nombres, donc à n'importe quelles données, une fois numérisées.

À noter, la cryptographie ne cherche pas à cacher les messages, mais à les protéger. Pour cacher des messages, il est nécessaire d'avoir recours à des techniques stéganograpiques (comme celles utilisées par les imprimantes évoquées auparavant), dont nous ne parlerons pas ici. page 26

5.1 Protéger des données des regards indiscrets

Comme l'ont bien compris des gamins utilisant des codes pour s'échanger des messages ou des militaires communiquant leurs ordres, la piste la plus sérieuse pour que des données ne puissent être comprises que par les personnes « dans le secret », c'est celle du *chiffrement*.

1. Pour un bon aperçu des différentes méthodes, qu'on appelle des « attaques », couramment utilisées en cryptanalyse, on peut se référer à la page de Wikipédia [https://secure.wikimedia.org/wikipedia/fr/wiki/Cryptanalyse].

Le chiffrement d'un fichier ou d'un support de stockage permet de le rendre illisible pour toute personne qui n'a pas le code d'accès (souvent une *phrase de passe*). Il sera certes toujours possible d'accéder au contenu, mais les données ressembleront à une série de nombres aléatoires, et seront donc illisibles.

Souvent on dit *crypter* et *décrypter* à la place de *chiffrer* et *déchiffrer*, ce qui peut porter à confusion ; les termes sont cependant synonymes.

Comment ça marche ?

Grosso modo, il y a seulement trois grandes idées pour comprendre comment on peut chiffrer des messages [2].

La première idée : la *confusion*. Il faut obscurcir la relation entre le message originel et le message chiffré. Un exemple très simple est le « chiffre de César » :

texte en clair	:	ASSAUT	DANS	UNE	HEURE
		↓↓↓↓↓↓	↓↓↓↓	↓↓↓	↓↓↓↓↓
texte chiffré	:	DVVDXW	GDQV	XQH	KHXUH

A + 3 lettres = D

Sauf qu'avec le chiffre de César, il est facile d'analyser la fréquence des lettres et de retrouver les mots.

Alors la deuxième grande idée, c'est la *diffusion*. Cela permet d'éclater le message pour le rendre plus difficile à reconnaître. Un exemple de cette technique, c'est la transposition par colonne :

$$\begin{pmatrix}A\\D\\E\end{pmatrix}\begin{pmatrix}S\\A\\H\end{pmatrix}\begin{pmatrix}S\\N\\E\end{pmatrix}\begin{pmatrix}A\\S\\U\end{pmatrix}\begin{pmatrix}U\\U\\R\end{pmatrix}\begin{pmatrix}T\\N\\E\end{pmatrix} \xrightarrow[\text{diffusion en 3 points}]{} \begin{matrix}\text{ADE SAH SNE}\\ \text{ASU UUR TNE}\end{matrix}$$

Dans ces deux petits exemples, on aurait pu décider de décaler de 6 caractères au lieu de 3, ou d'éclater les colonnes en utilisant 2 lignes au lieu de 3. On appelle ce morceau qui peut changer la *clé* de chiffrement. La méthode, on appelle ça un *algorithme*.

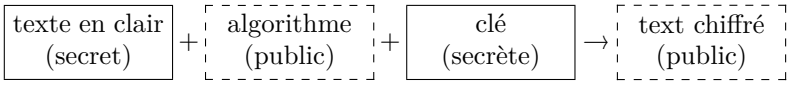
Ce qui nous amène à la troisième grande idée : *le secret réside seulement dans la clé*. Après quelques millénaires, on s'est aperçu que c'était une mauvaise idée de partir du principe que personne n'arriverait à comprendre l'algorithme de chiffrement. Tôt au tard, une personne finira bien par le découvrir… par la force si nécessaire.

De nos jours, l'algorithme peut donc être détaillé sur Wikipédia en long, en large et en travers, permettant à n'importe qui de vérifier qu'il n'a pas de point faible particulier, c'est-à-dire que la seule solution pour déchiffrer un texte sera de disposer de la *clé* qui a été employée avec celui-ci.

Vous voulez un dessin ?

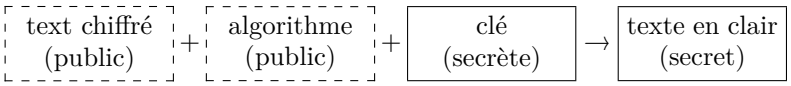
Concrètement, pour assurer la *confidentialité* de nos données, on utilise deux opérations :

Chiffrer



2. Le passage qui suit est une adaptation très partielle de la bande dessinée de Jeff Moser sur l'algorithme AES [http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html].

Déchiffrer



Pour un exemple d'usage pratique, prenons le message suivant [3] :

```
Les spaghetti sont dans le placard.
```

Après avoir chiffré ce message en utilisant le logiciel GnuPG avec l'algorithme AES256, et comme phrase de passe « *ceci est un secret* », on obtient :

```
-----BEGIN PGP MESSAGE-----

jA0ECQMCRM0lmTSIONRg0lkBWGQI76cQOocEvdBhX6BM2AU6aYSPYymSqj8ihFXu
wV1GVraWuwEt4XnLc3F+OxT3EaXINMHdH9oydA92WDkaqPEnjsWQs/oSCeZ3WXoB
9mf9y6jzqozEHw==
=T6eN
-----END PGP MESSAGE-----
```

Voici donc l'aspect que prend un texte après chiffrement : son contenu est devenu parfaitement imbuvable. Les données « en clair », lisibles par tout le monde, ont été transformées en un autre format, incompréhensible pour qui ne possède pas la clé.

Pour le déchiffrement, il nous suffira d'utiliser de nouveau GnuPG, avec notre texte chiffré, cette fois. Ce dernier nous demandera la phrase de passe, et si cette dernière est correcte, on obtiendra enfin l'information qui nous manquait pour préparer le déjeuner.

Pour un disque dur…

Pour réaliser le chiffrement d'un support de stockage (disque dur, clé USB, *etc.*), le système d'exploitation va se charger de réaliser « à la volée » les opérations de chiffrement et de déchiffrement. page 12

Ainsi, chaque fois que des données devront êtres lues du disque dur, elles seront déchiffrées au passage afin que les logiciels qui en ont besoin puissent y accéder. À l'inverse, chaque fois qu'un logiciel demandera à écrire des données, elles seront chiffrées avant d'atterrir sur le disque dur.

Pour que ces opérations fonctionnent, il est nécessaire que la clé de chiffrement se trouve en mémoire vive aussi longtemps que le support aura besoin d'être utilisé. page 11

Par ailleurs, la clé de chiffrement ne peut pas être changée. Une fois que cette dernière a servi à chiffrer des données inscrites sur le disque, elle devient indispensable pour pouvoir les relire. Pour pouvoir changer la clé, il faudrait donc relire puis réécrire l'intégralité des données du disque…

Pour éviter cette opération pénible, la plupart des systèmes utilisés pour chiffrer les supports de stockage utilisent donc une astuce : la clé de chiffrement est en fait un grand nombre, totalement aléatoire, qui sera lui-même chiffré à l'aide d'une *phrase de*

3. Ce message est d'une très haute importance stratégique pour des personnes qu'on inviterait chez soi. Il est donc crucial de le chiffrer.

passe[4]. Cette version chiffrée de la clé de chiffrement est généralement inscrite sur le support de stockage au début du disque, « *en tête* » des données chiffrées.

Avec ce système, changer le code d'accès devient simple, vu qu'il suffira de remplacer uniquement cet *en-tête* par un nouveau.

Résumé et limites

La cryptographie permet donc de bien protéger ses données, en chiffrant tout ou partie de son disque dur comme de tout autre support de stockage (clé USB, CD, *etc.*), ou de ses communications — point sur lequel nous reviendrons dans un autre tome de ce guide. De plus, les ordinateurs modernes sont suffisamment puissants pour que nous puissions espérer faire du chiffrement une routine, plutôt que de le réserver à des circonstances spéciales ou à des informations particulièrement sensibles (sinon, cela identifie tout de suite ces dernières comme importantes, alors qu'il vaut mieux les dissoudre dans la masse).

On peut ainsi mettre en place une phrase de passe pour chiffrer tout un disque dur, et/ou donner à certaines personnes une partie chiffrée avec leur propre phrase de passe. Il est également possible de chiffrer individuellement tel ou tel fichier, ou un email, ou une pièce jointe, avec une phrase de passe encore différente.

Cependant, bien qu'il soit un outil puissant et essentiel pour la sécurité des informations, **le chiffrement a ses limites** — en particulier lorsqu'il n'est pas utilisé correctement.

Comme expliqué auparavant, lorsqu'on accède à des données chiffrées, il est nécessaire de garder deux choses en tête. Premièrement, une fois les données déchiffrés, ces dernières se trouvent *au minimum* dans la mémoire vive. Deuxièmement, tant que des données doivent être chiffrées ou déchiffrées, la mémoire vive contient également la *clé de chiffrement*.

Toute personne qui dispose de la clé de chiffrement pourra lire *tout ce qui a été chiffré avec*, et aussi s'en servir pour chiffrer elle-même des données.

Il faut donc faire attention aux éléments suivants :

- Le système d'exploitation et les logiciels ont accès aux données et à la clé de chiffrement autant que nous, alors ça dépend de la confiance qu'on met en eux — encore une fois, il s'agit de ne pas installer n'importe quoi n'importe comment.
page 24
- Quiconque obtient un accès physique à l'ordinateur allumé a, de fait, accès au contenu de la mémoire vive. Lorsqu'un disque chiffré est activé, celle-ci contient, en clair, les données sur lesquelles on a travaillé depuis l'allumage de l'ordinateur (même si elles sont chiffrées sur le disque). Mais elle contient surtout, comme dit plus haut, la clé de chiffrement, qui peut donc être recopiée. Donc il vaut mieux s'habituer, quand on ne s'en sert pas, à éteindre les ordinateurs, et à désactiver (démonter, éjecter) les disques chiffrés.
page 19
- Dans certains cas, il peut être nécessaire de prévoir des solutions matérielles pour pouvoir couper le courant facilement et rapidement[5] ; ainsi les disques chiffrés redeviennent inaccessibles sans la phrase de passe — à moins d'effectuer une *cold boot attack*.
page 19
- Il reste également possible qu'un enregistreur de frappe ait été installé sur l'ordinateur, et que celui-ci enregistre la phrase de passe.
page 26

4. Le système LUKS, utilisé sous GNU/Linux, permet même d'utiliser plusieurs versions chiffrées de la clé de chiffrement. Chacune de ces versions pourra être chiffrée avec une *phrase de passe* différente, ce qui permet à plusieurs personnes d'accéder aux mêmes données sans pour autant avoir à retenir le même secret.

5. Pour cette raison, il est de bon ton de ne pas laisser la batterie branchée dans un ordinateur portable quand elle n'est pas utilisée. Il suffit alors d'enlever le câble secteur pour l'éteindre.

Par ailleurs, une certaine limite « légale » vient s'ajouter aux possibles attaques. En France, toute personne qui chiffre ses données est en effet censée donner le code d'accès aux autorités lorsqu'elles le demandent, comme l'explique l'*article 434–15–2 du Code Pénal*[6] :

```
Est puni de trois ans d'emprisonnement et de 45 000 euros d'amende
le fait, pour quiconque ayant connaissance de la convention secrète
de déchiffrement d'un moyen de cryptologie susceptible d'avoir été
utilisé pour préparer, faciliter ou commettre un crime ou un délit,
de refuser de remettre ladite convention aux autorités judiciaires
ou de la mettre en œuvre, sur les réquisitions de ces autorités
délivrées en application des titres II et III du livre Ier du code
de procédure pénale.

Si le refus est opposé alors que la remise ou la mise en œuvre de
la convention aurait permis d'éviter la commission d'un crime ou
d'un délit ou d'en limiter les effets, la peine est portée à cinq
ans d'emprisonnement et à 75 000 euros d'amende.
```

À noter là-dedans : *susceptible* et *sur les réquisitions*. C'est-à-dire que la loi est assez floue pour permettre d'exiger de toute personne détentrice de données chiffrées qu'elle crache le morceau. On peut éventuellement se voir demander la phrase de passe d'un support qui ne serait pas le nôtre… et que nous n'aurions donc pas. On notera que personne, à notre connaissance, n'a pour l'instant jamais été condamné pour ça.

Enfin, il peut être judicieux de rappeler que les mathématiques utilisées dans les algorithmes cryptographiques ont parfois des défauts. Et beaucoup plus souvent encore, les logiciels qui les appliquent comportent des faiblesses. Certains de ces problèmes peuvent, du jour au lendemain, transformer ce qu'on pensait être la meilleure des protections en une simple affaire de « double clic »…

5.2 S'assurer de l'intégrité de données

Nous avons vu quelques pistes pour assurer la *confidentialité* de nos données. Toutefois, il peut être aussi important de pouvoir s'assurer de leur *intégrité*, c'est-à-dire de vérifier qu'elles n'aient pas subi de modification (par accident ou malveillance). On peut également vouloir s'assurer de la provenance de nos données, en assurer l'*authenticité*.

Concrètement, après la lecture de ces pages, on peut comprendre à quel point il est critique de pouvoir s'assurer que les logiciels que l'on souhaite installer sur nos ordinateurs n'auraient pas été modifiées en route pour s'y voir cachés des logiciels malveillants. page 24

La puissance du hachoir

L'essentiel des techniques pour assurer l'intégrité ou l'authenticité reposent sur des outils mathématiques que la cryptographie a baptisés « fonctions de hachage ».

Ces dernières fonctionnent comme des *hachoirs*, capables de réduire n'importe quoi en tout petits morceaux. Et si notre hachoir fonctionne bien pour être utilisé en cryptographie, on sait que :

6. Le terme légal est « cryptologie ». Une recherche sur ce mot sur Légifrance [http://www.legifrance.gouv.fr] donnera une liste exhaustive des textes de loi concernant ce domaine.

- avec les petits morceaux, impossible de reconstituer l'objet original sans essayer tous les objets de la terre ;
- le même objet, une fois passé au hachoir, donnera toujours les mêmes petits morceaux ;
- deux objets différents doivent donner des petits morceaux différents.

Lorsque ces propriétés sont réunies, il nous suffit alors de comparer les petits morceaux issus de deux objets différents pour savoir si c'étaient les mêmes.

Les petits morceaux qui sortent de notre hachoir s'appellent plus couramment une *somme de contrôle* ou une *empreinte*. Elle est généralement écrite sous une forme qui ressemble à :

```
f9f5a68a721e3d10baca4d9751bb27f0ac35c7ba
```

Vu que notre hachoir fonctionne avec des données de n'importe quelle taille et de n'importe quelle forme, comparer des empreintes peut nous permettre de comparer plus facilement des images, des CD, des logiciels, *etc.*

Notre hachoir n'est pas magique pour autant. On imagine tout de même bien qu'en réduisant n'importe quoi en petits cubes de taille identique, on peut se retrouver avec les mêmes petits cubes issus de deux objets différents. Cela s'appelle une *collision*. Ce carambolage mathématique n'est heureusement dangereux que lorsqu'il est possible de le provoquer… ce qui est déjà arrivé pour plusieurs fonctions de hachage après quelques années de recherche.

Vérifier l'intégrité d'un logiciel

Prenons un exemple : Alice[7] a écrit un programme et le distribue sur des CD, que l'on peut trouver dans des clubs d'utilisateurs de GNU/Linux. Bob a envie d'utiliser le programme d'Alice, mais se dit qu'il aurait été très facile pour une administration mal intentionnée de remplacer un des CD d'Alice par un logiciel malveillant.

Il ne peut pas aller chercher un CD directement chez Alice, qui habite dans une autre ville. Par contre, il a rencontré Alice il y a quelque temps, et connaît sa voix. Il lui téléphone donc, et Alice lui donne la *somme de contrôle* du contenu du CD :

$$\text{CD d'Alice} \xrightarrow[\text{SHA256}]{} \begin{matrix}\texttt{94d93910609f65475a189d178ca6a45f}\\ \texttt{22b50c95416affb1d8feb125dc3069d0}\end{matrix}$$

Bob peut ensuite la comparer avec celle qu'il génère à partir du CD qu'il s'est procuré :

$$\text{CD de Bob} \xrightarrow[\text{SHA256}]{} \begin{matrix}\texttt{94d93910609f65475a189d178ca6a45f}\\ \texttt{22b50c95416affb1d8feb125dc3069d0}\end{matrix}$$

Comme les nombres sont les mêmes, Bob est content, il est sûr de bien utiliser le même CD que celui fourni par Alice.

Calculer ces sommes de contrôle ne leur prend pas beaucoup plus de temps que la lecture complète du CD… soit quelques minutes tout au plus.

Maintenant, mettons-nous dans la peau d'Ève, qui a été payée pour prendre le contrôle de l'ordinateur de Bob à son insu. Pour cela, elle veut créer un CD qui ressemble à celui d'Alice, mais qui contient un logiciel malveillant.

7. Les prénoms utilisés dans cet exemple sont les prénoms utilisés traditionnelement dans les scénarios de cryptographie. Alice et Bob cherchent à communiquer tout en échappant à la surveillance d'Ève. Ce dernier prénom vient de la consonnance en anglais avec *eavesdropping*, que l'on peut traduire par « écouter aux portes ».

Malheureusement pour elle, la fonction de hachage ne va que dans un sens. Elle doit donc commencer par se procurer le CD original d'Alice.

Ensuite, elle modifie ce CD pour y introduire le logiciel malveillant. Cette première version ressemble de très près à l'original. Cela pourrait duper plus d'une personne qui ne ferait pas attention, mais elle sait que Bob verifiera la somme de contrôle du CD qui lui permettra d'installer la nouvelle version.

Comme Alice utilise la fonction de hachage SHA256, qui n'a pas de défaut connu, il ne reste à Ève qu'à essayer un très grand nombre de variation des données de son CD, cela dans l'espoir d'obtenir une *collision*, soit la même somme de contrôle que celle d'Alice.

Malheureusement pour elle, et heureusement pour Bob, même avec de nombreux ordinateurs puissants, les chances de réussite d'Ève dans un temps raisonnable (mettons, quelques années) sont extrêmement faibles.

Il suffit donc de se procurer une *empreinte*, ou *somme de contrôle* par des intermédiaires de confiance pour vérifier l'*intégrité* de données. Tout l'enjeu est ensuite de se procurer ces empreintes par un moyen de confiance : de pouvoir vérifier leur *authenticité*...

Vérifier un mot de passe

Un autre exemple d'utilisation des fonctions de hachage concerne la vérification de l'*authenticité* d'une demande d'accès.

Si l'accès à un ordinateur est protégé par un mot de passe, comme l'ouverture d'une session sous GNU/Linux [8], il faut que l'ordinateur puisse vérifier que le mot de passe est le bon. Mais les mots de passe ne sont pas enregistrés sur l'ordinateur, car il serait trop facile de les lire.

Mais alors comment l'ordinateur s'assure-t-il que le mot de passe tapé au clavier est le bon ?

Lorsque l'on choisisit un mot de passe pour son ordinateur, le système enregistre en fait, grâce à une fonction de hachage, une empreinte du mot de passe. Pour vérifier l'accès, il « hache » de la même manière le mot de passe que l'on a saisi. Et si les empreintes sont les mêmes, il considère que le mot de passe était le bon.

Il est donc possible de vérifier que le mot de passe correspond, sans garder le mot de passe lui-même !

5.3 Symétrique, asymétrique ?

Les techniques de chiffrement mentionnées jusqu'ici reposent sur une seule clé secrète, qui permet à la fois d'effectuer le chiffrement et le déchiffrement. On parle dans ce cas de chiffrement *symétrique*.

Ceci en opposition avec le chiffrement *asymétrique* qui n'utilise pas la même clé pour chiffrer et déchiffrer. Autrement appelé « chiffrement à clé publique », ce dernier est surtout utilisé pour la communication « en ligne », on en parlera donc en détail dans le prochain tome [9].

Une des propriétés les plus intéressantes de la cryptographie asymétrique que l'on peut évoquer brièvement est la possibilité de réaliser des *signatures numériques*. Comme

8. Rappelons-nous que ces mots de passe ne servent pas à protéger les données [page 31] !
9. Pour aller plus loin dès maintenant, on pourra notamment se référer au site http://www.cryptage.org/ et à sa bibliographie.

son équivalent papier, une signature numérique permet d'apposer une marque de reconnaissance sur des données.

Ces signatures numériques utilisant la cryptographie asymétrique constituent la façon la plus simple de vérifier la provenance d'un logiciel. On sera donc amené à s'en servir plus loin…

Deuxième partie

Choisir des réponses adaptées

La panique s'est désormais emparée de nous. Tout ce qu'on fait sur un ordinateur nous trahit, jour après jour. Qui plus est lorsqu'on croit, à tort, « être en sécurité ».

Mais avant de retourner au pigeon voyageur et à la cache secrète derrière la bibliothèque, qu'on ouvre en tirant sur un faux livre (solutions rustiques à ne pas oublier totalement, ceci dit…), il y a un peu de marge. Pas tant que ça, mais tout de même.

C'est cette marge que ce texte s'appliquera dorénavant à cartographier.

Dans cette partie, c'est en expliquant quelques idées, tout aussi importantes qu'elles sont générales, que nous brosserons le tableau d'une méthodologie sommaire permettant à quiconque de répondre à la question suivante : *comment décider d'un ensemble de pratiques et d'outils adéquats à notre situation ?* Nous décrirons ensuite quelques situations-types, que nous nommons des *cas d'usage*, afin d'illustrer notre propos.

Chapitre 6

Évaluation des risques

Quand on se demande quelles mesures mettre en place pour protéger des données ou des communications numériques, on se rend assez vite compte qu'en la matière, on avance un peu à l'aveuglette.

D'abord parce que la plupart des solutions qu'on pourrait mettre en place ont aussi leurs inconvénients : parfois elles sont très pénibles à déployer, à entretenir ou à utiliser ; parfois on a le choix entre diverses techniques, dont aucune ne répond complètement au « cahier des charges » que l'on s'est fixé ; parfois elles sont bien trop nouvelles pour avoir l'assurance qu'elles fonctionnent réellement ; *etc.*

6.1 Que veut-on protéger ?

Dans le cadre de ce texte, ce qu'on veut protéger rentre en général dans la vaste catégorie de l'*information* : par exemple, le contenu de messages électroniques, des fichiers de données (photo, tracts, carnet d'adresses) ou l'existence même d'une correspondance entre telle et telle personne.

Le mot « protéger » recouvre différents besoins :

- **confidentialité** : cacher des informations aux yeux indésirables ;
- **intégrité** : conserver des informations en bon état, et éviter qu'elles ne soient modifiées sans qu'on s'en rende compte ;
- **accessibilité** : faire en sorte que des informations restent accessibles aux personnes qui en ont besoin.

Il s'agit donc de définir, pour chaque ensemble d'informations à protéger, les besoins de confidentialité, d'intégrité et d'accessibilité. Sachant que ces besoins entrent généralement en conflit, on réalise dès maintenant qu'il faudra, par la suite, poser des priorités et trouver des compromis entre eux : en matière de sécurité informatique, on a rarement le beurre et l'argent du beurre.

6.2 Contre qui veut-on se protéger ?

Et surtout, rapidement, se pose la question des capacités des personnes qui en auraient après ce que l'on veut protéger. Et là, ça se corse, parce qu'il n'est par exemple pas facile de savoir ce que les personnes les plus qualifiées peuvent réellement faire, et de quels moyens et de quels budgets elles bénéficient. En suivant l'actualité, et par divers autres biais, on peut se rendre compte que cela varie beaucoup selon à qui on

a affaire. Entre le gendarme du coin et la *National Security Agency* américaine, il y a tout un fossé sur les possibilités d'actions, de moyens et de techniques employées.

page 37

Par exemple, le chiffrement est un des moyens les plus adaptés pour éviter qu'une personne qui allumerait, déroberait ou saisirait judiciairement un ordinateur accède à toutes les données qui y résident. Mais les lois en vigueur en France ont prévu le coup : dans le cadre d'une enquête, toute personne doit donner la clé de chiffrement afin de permettre aux enquêteurs d'avoir accès aux données, sans quoi elle risque des peines assez lourdes. Cette loi permet à des enquêteurs ayant peu de moyens techniques d'agir contre ce type de protection, même si en réalité, nous ne connaissons aucun cas où cette loi a été appliquée. En parallèle, des organismes disposent de plus de moyens, tels la NSA ou la DGSE, et rien n'est sûr concernant leurs possibilités. Quelle avance ont-ils dans le domaine du cassage de cryptographie ? Sont-ils au courant de failles dans certaines méthodes, qu'ils n'auraient pas dévoilées, et qui leur permettraient de lire les données ? Sur ces sujets, il n'y a évidemment aucun moyen d'être sûr de ce que ces entités peuvent faire, mais en même temps leur champ d'intervention est limité, et il y a peu de cas pour lesquels on risque d'être confronté à elles.

Un facteur important est aussi à prendre en compte : le coût. En effet, plus les moyens mis en place sont importants, plus les technologies utilisées sont complexes, et plus leur coût est élevé ; ça signifie qu'ils ne seront utilisés que dans des cas précis et tout aussi importants aux yeux des personnes concernées. Par exemple, il y a peu de chances de voir un ordinateur soumis à d'intenses tests dans de coûteuses expertises pour une affaire de vol à l'étalage.

Dès lors, avant même de chercher une solution, la question est de savoir qui pourrait tenter d'accéder à nos informations sensibles, afin de discerner s'il est nécessaire de chercher des solutions compliquées ou pas. Sécuriser complètement un ordinateur est de toutes façons de l'ordre de l'impossible, et dans cette histoire, il s'agit plutôt de mettre des bâtons dans les roues de celles et ceux qui pourraient en avoir après ce que l'on veut protéger. Plus l'on pense grands les moyens de ces personnes, plus les bâtons doivent être nombreux et solides.

Évaluer les risques, c'est donc avant tout se poser la question de quelles sont les données que l'on veut protéger, et de qui peut être intéressé par ces données. À partir de là, on peut avoir une vision de quels moyens ils disposent (ou en tout cas, dans la mesure du possible, essayer de se renseigner) et en conséquence, définir une *politique de sécurité* adaptée.

Chapitre 7

Définir une politique de sécurité

Une chaîne n'a que la solidité de son maillon le plus faible. Rien ne sert d'installer trois énormes verrous sur une porte blindée placée à côté d'une frêle fenêtre délabrée. De même, chiffrer une clé USB ne rime pas à grand-chose si les données qui y sont stockées sont utilisées sur un ordinateur qui en conservera diverses traces *en clair* sur son disque dur.

page 37
page 19

Ces exemples nous apprennent quelque chose : de telles « solutions » ciblées ne sont d'aucune utilité tant qu'elles ne font pas partie d'un ensemble de pratiques articulées de façon cohérente. Qui plus est, les informations qu'on veut protéger sont le plus souvent en relation avec des pratiques hors du champ des outils numériques. C'est donc de façon globale qu'il faut évaluer les risques et penser les réponses adéquates.

page 49

De façon globale, mais localisée : à une situation donnée correspond un ensemble singulier d'enjeux, de risques, de savoirs-faire… et donc de possibilités d'action. Il n'existe pas de solution miracle convenant à tout le monde, et qui réglera tous les problèmes d'un coup de baguette magique. La seule voie praticable, c'est d'en apprendre suffisamment pour être capables d'imaginer et de mettre en place une politique de sécurité adéquate à sa propre situation.

7.1 Une affaire de compromis

On peut toujours *mieux* protéger ses données et ses communications numériques. Il n'y a de limite ni aux possibilités d'attaque et de surveillance, ni aux dispositifs qu'on peut utiliser pour s'en protéger. Cependant, à chaque protection supplémentaire qu'on veut mettre en place correspond un effort en termes d'apprentissage, de temps ; non seulement un effort initial pour s'y mettre, pour installer la protection, mais aussi, bien souvent, une complexité d'utilisation supplémentaire, du temps passé à taper des phrases de passe, à effectuer des procédures pénibles et répétitives, à porter son attention sur la technique plutôt que sur l'usage qu'on voudrait avoir de l'ordinateur.

Dans chaque situation, il s'agit donc de trouver un **compromis** convenable entre la facilité d'utilisation et le niveau de protection souhaité.

Parfois, ce compromis **n'existe** tout simplement **pas** : on doit parfois conclure que les efforts qui seraient nécessaires pour se protéger contre un risque plausible seraient trop pénibles, et qu'il vaut mieux courir ce risque… ou bien, tout simplement, ne pas utiliser d'outils numériques pour stocker certaines données ou pour parler de certaines choses. D'autres moyens existent, à l'efficacité prouvée de longue date : certains manuscrits de la Bible ont survécu des siècles durant, enfouis dans des jarres entreposées dans des grottes…

7.2 Comment faire ?

Il s'agit de répondre à la question suivante : quel ensemble de pratiques, d'outils me protégeraient de façon suffisante contre les risques évalués précédemment ?

page 49

Vous pouvez par exemple partir de vos pratiques actuelles, et vous mettre dans la peau de l'adversaire — aussi nauséabonde soit-elle — pour vous poser les questions suivantes :

1. Face à une telle politique de sécurité, quels sont les angles d'attaque les plus praticables ?
2. Quels sont les moyens à mettre en œuvre pour ce faire ?
3. Croyez-vous que ces moyens puissent être utilisés par les adversaires ?

Si vous répondez « oui » à la troisième question, prenez le temps de vous renseigner sur les solutions qui permettraient de vous protéger contre ces attaques, puis imaginez les modifications de pratiques entraînées par ces solutions et la politique de sécurité qui en découle. Si ça vous semble praticable, remettez-vous dans la peau de l'adversaire, et posez-vous à nouveau les questions énoncées ci-dessus.

Réitérez ce processus de réflexion, recherche et imagination jusqu'à trouver une voie praticable, un compromis tenable.

En cas d'incertitude, il est toujours possible de demander à une personne digne de confiance et plus compétente en la matière de se mettre dans la peau de l'adversaire : elle sera ravie de constater que vous avez fait vous-mêmes le gros du travail de réflexion, ce qui l'encouragera certainement à vous aider sur les points qui restent hors de votre portée.

7.3 Quelques règles

Avant de s'intéresser de plus près à l'étude de cas concrets et des politiques de sécurité qu'il serait possible de mettre en place, il existe quelques grands principes, quelques grandes familles de choix…

Complexe vs. simple

En matière de sécurité, une solution simple doit toujours être préférée à une solution complexe.

Tout d'abord, parce qu'une solution complexe offre plus de « surface d'attaque », c'est-à-dire plus de lieux où peuvent apparaître des problèmes de sécurité… ce qui ne manquera pas d'arriver.

Ensuite, parce que plus une solution est complexe, plus il faut de connaissances pour l'imaginer, la mettre en œuvre, la maintenir… mais aussi pour l'examiner, évaluer sa pertinence et ses problèmes. Ce qui fait qu'en règle générale, plus une solution est complexe, moins elle aura subi les regards acérés — et extérieurs — nécessaires pour établir sa validité.

Enfin, tout simplement, une solution complexe, qui ne tient pas en entier dans l'espace mental des personnes qui l'ont élaborée, a plus de chances de générer des problèmes de sécurité issus d'interactions complexes ou de cas particuliers difficiles à déceler.

Par exemple, plutôt que de passer des heures à mettre en place des dispositifs visant à protéger un ordinateur particulièrement sensible contre les intrusions provenant du réseau, autant l'en débrancher. On peut même parfois retirer physiquement la carte réseau…

page 13

Liste blanche, liste noire

Le réflexe courant, lorsqu'on prend connaissance d'une menace, est de chercher à s'en prémunir. Par exemple, après avoir découvert que tel logiciel laisse des traces de nos activités dans tel dossier, on nettoiera régulièrement cet emplacement. Jusqu'à découvrir que le même logiciel laisse aussi des traces dans un autre dossier, et ainsi de suite.

C'est le principe de la liste noire : une liste des dossiers où sont enregistrés les fichiers temporaires, de logiciels qui envoient des rapports, *etc.* ; cette liste est complétée au fil des découvertes et des mauvaises surprises ; sur cette base, on essaie de faire au mieux pour se prémunir de chacune de ces menaces. Autrement dit, une liste noire fonctionne sur la base de la *confiance-sauf-dans-certains-cas*.

Le principe de la liste blanche est inverse, car c'est celui de la *méfiance-sauf-dans-certains-cas*. On interdit *tout*, *sauf* ce qu'on autorise explicitement. On interdit l'enregistrement de fichiers sur le disque dur, sauf à tel endroit, à tel moment. On interdit aux logiciels d'accéder au réseau, sauf certains logiciels bien choisis. page 113

Voilà pour les principes de base.

Toute politique de sécurité basée sur le principe de la *liste noire* a un gros problème : une telle liste n'est jamais complète, car elle prend uniquement en compte les problèmes qui ont déjà été repérés. C'est une tâche sans fin, désespérante, que de tenir à jour une liste noire ; qu'on le fasse nous-mêmes ou qu'on le délègue à des gens ayant des connaissances informatiques pointues, quelque chose sera forcément oublié.

L'ennui, c'est que malgré leurs défauts rédhibitoires, les outils basés sur une approche *liste noire* sont légion (comme nous allons le voir), au contraire de ceux s'appuyant sur la méthode *liste blanche*, qui nous est donc, sans doute, moins familière.

Mettre en œuvre l'approche *liste blanche* requiert donc un effort initial qui, s'il peut être important, est bien vite récompensé : apprendre à utiliser un système *live* qui n'écrit rien sur le disque dur sans qu'on lui demande, ça prend un temps non négligeable, mais une fois que c'est fait, c'en est fini des longues séances de nettoyage de disque dur, toujours à recommencer, et inefficaces car basées sur le principe de *liste noire*. page 101

Une autre illustration nous est fournie par les logiciels antivirus, qui visent à empêcher l'exécution de programmes mal intentionnés. Vu qu'ils fonctionnent sur le principe de la liste noire, leurs bases de données doivent perpétuellement être mises à jour, systématiquement en retard. Une réponse à ce problème, avec l'approche liste blanche, est d'empêcher l'exécution de tout programme qui n'a pas été enregistré au préalable, ou de limiter les possibilités d'action de chaque programme ; ces techniques, nommées *Mandatory Access Control*, nécessitent aussi de maintenir des listes, mais il s'agit dans ce cas de listes *blanches*, et le symptôme d'une liste obsolète sera le dysfonctionnement d'un logiciel, plutôt que le piratage de l'ordinateur.

Aussi, il est bien plus intéressant de se donner les moyens, lorsque c'est possible, de s'appuyer sur des listes blanches les plus vastes possible, afin de pouvoir faire plein de choses chouettes avec des ordinateurs, dans une certaine confiance. Et de s'appuyer, quand la liste blanche adéquate n'existe pas, sur des listes noires solides, de provenance connue, en gardant en tête le problème intrinsèque à cette méthode ; listes noires qu'on aidera éventuellement à compléter, en partageant nos découvertes.

On n'est pas des robots

Certaines pratiques très exigeantes peuvent être diablement efficaces… jusqu'à ce qu'on commette une erreur. Alors comme on finira forcément par en faire une, il vaut mieux les prévoir plutôt que de payer les pots cassés.

Par exemple, une clé USB destinée à n'être utilisée que sur des ordinateurs utilisant un système libre, et qu'on fait vraiment attention à ne pas laisser traîner, peut *quand même* finir par être oubliée sur une table… et être branchée sur Windows par une personne qui l'aura confondue avec une autre. Mais si elle a été formatée dès le départ avec un système de fichiers incompatible avec Windows, ça devrait limiter la casse…

page 16

Bref, on n'est pas des robots. Il vaut mieux se donner de solides garde-fous matériels, que de s'imposer une vigilance sans bornes — ça permet aussi de garder l'esprit tranquille.

Date limite de consommation

Une fois une politique de sécurité définie, il ne faut pas oublier de la revoir de temps en temps ! Le monde de la sécurité informatique évolue très vite, et une solution considérée comme raisonnablement sûre à l'heure actuelle peut très bien être aisément attaquable l'an prochain.

N'oublions pas non plus de penser dans nos politiques de sécurité qu'il est important de surveiller la vie des logiciels dont on dépend : leurs problèmes, avec une incidence sur la sécurité, leurs mises à jour, avec parfois de bonnes ou de mauvaises surprises… Tout cela prend un peu de temps, et autant le prévoir dès le départ.

Cas d'usages

Trêve de théorie, illustrons maintenant ces notions avec quelques *cas d'usage* : à partir de situations données, nous indiquerons des pistes permettant de définir une politique de sécurité adéquate. Bon nombre des solutions techniques retenues seront expliquées dans la partie suivante, vers laquelle nous renverrons au besoin.

page 83

Vu qu'ils s'inscrivent tous dans le contexte hors-connexions de ce premier tome, ces cas d'usage auront quelque chose d'artificiel : ils partent tous du principe que les ordinateurs en jeu ne sont jamais connectés à des réseaux, et en particulier à Internet.

Chapitre 8

Cas d'usage : un nouveau départ, pour ne plus payer les pots cassés

(ou comment faire le ménage sur un ordinateur après des années de pratiques insouciantes)

8.1 Contexte

Prenons un ordinateur utilisé sans précautions particulières pendant plusieurs années. Cette machine pose sans doute un ou plusieurs des problèmes suivants :

1. son disque dur conserve des traces indésirables du passé ; page 19
2. le système d'exploitation est un logiciel propriétaire (exemple : Windows), et truffé de logiciels malveillants. page 23

Par ailleurs, des fichiers gênants y sont stockés de façon parfaitement transparente. En effet, cet ordinateur est utilisé pour diverses activités populaires, parmi lesquelles certaines, osons l'avouer, sont parfaitement légales, telles que :

- écouter de la musique et regarder des films pris sur Internet ;
- aider des sans-papiers à préparer leurs dossiers pour la préfecture ;
- dessiner une jolie carte de vœux pour Mamie ;
- fabriquer de menus faux papiers simplifiant grandement les démarches administratives (gonfler des fiches de paie, quand on en a marre de se voir refuser des locations, appart' après appart') ;
- tenir à jour la comptabilité familiale ;
- fabriquer des textes, musiques ou vidéos « terroristes » — plus précisemment menaçant, selon la définition européenne du terrorisme [1], « *de causer [...] des destructions massives [...] à une infrastructure [...] susceptible [...] de produire des pertes économiques considérables* », « *dans le but de [...] contraindre indûment des pouvoirs publics [...] à accomplir ou à s'abstenir d'accomplir un acte quelconque* » ; par exemple, des employés de France Télécom qui, lors d'une lutte, menaceraient de mettre hors d'état de nuire le système de facturation, et d'ainsi permettre aux usagers de téléphoner gratuitement.

8.2 Évaluer les risques

Que veut-on protéger ?

Appliquons au cas présent les catégories définies lorsque nous parlions d'évaluation page 49

1. Décision-cadre 2002/475/JAI du Conseil de l'Union Européenne, relative à la lutte contre le terrorisme, 13 juin 2002 [http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002F0475:FR:NOT].

des risques :

- confidentialité : éviter qu'un œil indésirable ne tombe trop aisément sur les informations stockées dans l'ordinateur ;
- intégrité : éviter que ces informations ne soient modifiées à notre insu ;
- accessibilité : faire en sorte que ces informations restent accessibles quand on en a besoin.

Ici, accessibilité et confidentialité sont prioritaires.

Contre qui veut-on se protéger ?

Cette question est importante : en fonction de la réponse qu'on lui donne, la politique de sécurité adéquate peut varier du tout au tout.

Geste généreux, conséquences judiciaires

Cet ordinateur pourrait être saisi lors d'une perquisition.

Par exemple, votre fils a généreusement donné un gramme de *shit* à un ami fauché, qui, après s'être fait pincer, a informé la police de la provenance de la chose… à la suite de quoi votre fils est pénalement considéré comme trafiquant de stupéfiants. D'où la perquisition.

Dans ce genre de cas, l'ordinateur a de grandes chances d'être examiné par la police, mettant en péril l'objectif de confidentialité. La gamme de moyens qui seront probablement mis en œuvre va du gendarme de Saint-Tropez, allumant l'ordinateur et cliquant partout, à l'expert judiciaire qui examinera de beaucoup plus près le disque dur ; il est en revanche improbable que des moyens extra-légaux, usuellement aux mains des services spéciaux et des militaires, soient utilisés dans cette affaire.

Cambriolage

Cet ordinateur pourrait être dérobé lors d'un cambriolage.

Au contraire de la police, les voleurs n'ont sans doute pas grand-chose à faire de vos petits secrets… et ne vous dénonceront pas. Au pire vous feront-ils chanter à propos de la récupération de vos données. Il est cependant improbable qu'ils mettent en œuvre de grands moyens pour les retrouver sur le disque dur de l'ordinateur.

8.3 Définir une politique de sécurité

Posez-vous maintenant, en vous mettant dans la peau de l'adversaire, les questions exposées dans notre méthodologie.

page 51

Première étape : quand ouvrir les yeux suffit pour voir

1. Angle d'attaque le plus praticable : brancher le disque dur sur un autre ordinateur, examiner son contenu, y trouver tous vos petits secrets.
2. Moyens nécessaires : un autre ordinateur, dont le gendarme de Saint-Tropez se servira pour trouver le plus gros de vos secrets ; un expert judiciaire, lui, saurait aussi retrouver les fichiers que vous croyiez avoir effacés ; Nostradamus en déduirait la date de levée de vos semis.
3. Crédibilité de l'attaque : grande.

page 37 Il faut donc adapter vos pratiques. Contre ce type d'attaque, chiffrer le disque dur est la réponse évidente : installer et utiliser un tel système est désormais relativement simple. page 105

Les étapes pour y arriver seraient alors :

1. Lancer un système *live* afin d'effectuer les opérations suivantes dans un contexte relativement sûr :
   - sauvegarder temporairement, sur un disque externe ou une clé USB chiffrés, les fichiers qui doivent survivre au grand nettoyage ;
   - éjecter/démonter et débrancher ce support de stockage externe ;
   - effacer « pour de vrai » l'intégralité du disque dur **interne** de l'ordinateur.
2. Installer un système d'exploitation libre, en précisant au programme d'installation de chiffrer le disque dur, mémoire virtuelle (*swap*) comprise. page 17
3. Recopier vers le nouveau système les données préalablement sauvegardées.
4. Mettre en place ce qu'il faut pour supprimer des fichiers de façon « sécurisée », afin de pouvoir…
5. Effacer le contenu des fichiers qui se trouvent sur le support de sauvegarde temporaire, qui pourra éventuellement resservir.

Et ensuite, de temps à autre, faire en sorte que les données supprimées sans précautions particulières ne soient pas récupérables par la suite.

Pour effectuer ces étapes, se référer aux recettes suivantes :

- chiffrer une clé USB, voir page 143 ;
- utiliser un système *live*, voir page 101 ;
- sauvegarder des données, voir page 151 ;
- effacer « pour de vrai », voir page 127 ;
- installer un système chiffré, voir page 105.

Cette voie semblant praticable, posons-nous, de nouveau, les mêmes questions.

Seconde étape : le tiroir de la commode n'était pas chiffré

1. Angle d'attaque : l'équivalent des fichiers qu'on cherche à protéger traîne peut-être dans la pièce voisine, dans le troisième tiroir de la commode, sur papier ou sur une clé USB.
2. Moyens nécessaires : perquisition, cambriolage, ou autre visite impromptue.
3. Crédibilité de l'attaque : grande, c'est précisément contre ce type de situations qu'on cherche à se protéger ici.

Là encore, on constate qu'une politique de sécurité doit être pensée comme un tout. Sans un minimum de cohérence dans les pratiques, rien ne sert de s'embêter à taper des phrases de passe longues comme un jour sans pain. page 51

Il est donc temps de trier les papiers dans la commode, et de nettoyer toute clé USB, CD, DVD contenant des données qu'on compte désormais chiffrer :

1. sauvegarder sur un support chiffré les données à conserver
2. pour les clés USB et disques durs externes : effacer pour de vrai leur contenu ; page 127
3. pour les CD et DVD : les détruire, et se débarrasser des résidus ;
4. décider que faire des données préalablement sauvegardées : les recopier sur le disque dur nouvellement chiffré ou les archiver. page 79

Troisième étape : la loi comme moyen de cœrcition

1. Angle d'attaque : la police a le droit d'exiger que vous lui donniez accès aux informations chiffrées, comme expliqué dans le chapitre consacré à la cryptographie.
2. Moyens nécessaires : suffisamment de persévérance dans l'enquête pour appliquer cette loi.
3. Crédibilité de l'attaque : encore faut-il que la police considère pouvoir trouver des éléments à charge sur l'ordinateur, avec suffisamment de foi pour pousser le bouchon jusque-là. Dans le strict cadre de l'enquête qui part du gramme de *shit*, c'est peu probable, mais pas du tout impossible.

page 37

Si la police en arrive à exiger l'accès aux données chiffrées, se posera, *en pratique*, la question suivante : les informations contenues dans l'ordinateur font-elles encourir plus de risques que le refus de donner la phrase de passe ? Après, c'est selon comment on le sent. Céder, dans cette situation, ne remet pas en cause tout l'intérêt de chiffrer, au départ, son disque dur : ça permet tout au moins de savoir ce qui a été dévoilé, quand, et à qui.

Ceci dit, il peut être bon de s'organiser pour vivre de façon moins délicate une telle situation : le nouvel objectif pourrait être d'avoir un disque dur suffisamment « propre » pour que ce ne soit pas la catastrophe si on cède face à la loi, ou si le système cryptographique utilisé est cassé.

Comme premier pas, il est souvent possible de faire un compromis concernant l'accessibilité, pour des fichiers concernant des projets achevés dont on n'aura pas besoin souvent ; on traitera ceci dans le cas d'usage sur l'archivage, qu'il pourra être bon d'étudier après celui-ci.

page 79

Ensuite, c'est donc toute la question de la compartimentation qui se pose ; en effet, s'il est possible d'augmenter globalement, de nouveau, le niveau de sécurité de l'ensemble des activités pratiquées… ce serait trop pénible à l'usage. Il convient donc de préciser les besoins respectifs, en termes de confidentialité, de ces diverses activités. Et, à partir de là, faire le tri et décider lesquelles, plus « sensibles » que les autres, doivent bénéficier d'un traitement de faveur.

page 65

Le prochain cas d'usage étudiera de tels traitements de faveur, mais patience, mieux vaut pour l'instant terminer la lecture de celui-ci !

Quatrième étape : en réseau

Tout ceci est valable pour un ordinateur hors-ligne. D'autres angles d'attaques sont imaginables, s'il est connecté à un réseau. Le second tome de ce guide les étudiera.

Et au-delà de ces problèmes, plusieurs autres angles d'attaque demeurent encore envisageables contre une telle politique de sécurité.

Angle d'attaque : une brèche dans le système de chiffrement utilisé

Comme il a déjà été expliqué en ces pages, tout système de sécurité finit par être cassé. Si l'algorithme de chiffrement utilisé est cassé, ça fera la une des journaux, tout le monde sera au courant, et il sera possible de réagir.

Mais si c'est sa mise en œuvre dans le noyau Linux qui est cassée, ça ne passera pas dans *Libé*, et il y a fort à parier que seuls les spécialistes de la sécurité informatique seront au courant.

Lorsqu'on ne côtoie pas de tels êtres, une façon de se tenir au courant est de s'abonner aux annonces de sécurité de Debian [2]. Les emails reçus par ce biais sont rédigés en anglais, mais ils donnent l'adresse de la page où on peut trouver leur traduction française. La difficulté, ensuite, est de les interpréter…

Ceci étant dit, même si le système de chiffrement utilisé est « cassé », encore faut-il que les adversaires le sachent… le gendarme de Saint-Tropez n'en saura rien, mais un expert judiciaire, si.

Par ailleurs, dans le rayon science-fiction, rappelons qu'il est difficile de connaître l'avance qu'ont, en la matière, militaires et agences gouvernementales — comme la NSA.

Angle d'attaque : *cold boot attack*

1. Angle d'attaque : la *cold boot attack* est décrite dans le chapitre consacré aux traces. page 19
2. Moyens nécessaires : accéder physiquement à l'ordinateur pendant qu'il est allumé ou éteint depuis peu, par exemple lors d'une perquisition.
3. Crédibilité de l'attaque : à notre connaissance, cette attaque n'a jamais été utilisée, du moins de façon publique, par des autorités. Sa crédibilité, dans cette affaire, est donc très faible.

Il peut sembler superflu de se protéger contre une telle attaque dans la situation décrite ici, mais mieux vaut prendre, dès maintenant, de bonnes habitudes, plutôt que d'avoir de mauvaises surprises dans quelques années. Quelles habitudes ? En voici quelques-unes qui rendent plus difficile cette attaque :

- éteindre l'ordinateur lorsqu'on ne s'en sert pas ;
- prévoir la possibilité de couper le courant facilement et rapidement : interrupteur de multiprise aisément accessible, ôter la batterie d'un ordinateur portable quand il est branché sur le secteur (… il suffit alors de débrancher le cordon secteur pour éteindre la machine).

Angle d'attaque : l'œil et la vidéo-surveillance

Avec le système chiffré imaginé à la première étape, la confidentialité des données repose sur le fait que la phrase de passe soit gardée secrète. Si elle est tapée devant une caméra de vidéo-surveillance, un adversaire ayant accès à cette caméra ou à ses éventuels enregistrements pourra découvrir ce secret, puis se saisir de l'ordinateur et avoir accès aux données. Plus simplement, un œil attentif, dans un bar, pourrait voir la phrase de passe pendant qu'elle est tapée.

Monter une telle attaque nécessite de surveiller les personnes utilisant cet ordinateur, jusqu'à ce que l'une d'entre elles tape la phrase de passe au mauvais endroit. Ça peut prendre du temps et c'est coûteux.

Dans la situation décrite ici, une telle attaque relève de la pure science-fiction ; à l'heure actuelle, rares sont les organisations susceptibles de mettre en œuvre des moyens aussi conséquents, mis à part divers services spéciaux : anti-terroristes, espionnage industriel…

Pour se prémunir d'une telle attaque, il convient de :

- choisir une longue phrase de passe, qui rend impossible la mémorisation « à la volée » par un observateur humain ; page 93

2. La liste de diffusion se nomme debian-security-announce [http://lists.debian.org/debian-security-announce/].

- vérifier autour de soi, à la recherche d'éventuels yeux (humains ou électroniques) indésirables, avant de taper sa phrase de passe.

Angle d'attaque : la partie non-chiffrée et le BIOS

page 105 Comme expliqué dans la recette dédiée, un système « chiffré » ne l'est pas entièrement : le petit logiciel qui nous demande, au démarrage, la phrase de passe de chiffrage du *reste* des données, est, lui, stocké en clair sur la partie du disque dur qu'on nomme `/boot`. Un attaquant ayant accès à l'ordinateur peut aisément, en quelques minutes, page 23 modifier ce logiciel, y installer un *keylogger*, qui conservera la phrase de passe, pour venir la chercher plus tard, ou, tout simplement, l'enverra par le réseau.

Si cette attaque est montée à l'avance, l'adversaire pourra déchiffrer le disque dur quand il se saisira de l'ordinateur, lors d'une perquisition par exemple.

Les moyens nécessaires pour cette attaque sont, somme toute, assez limités : *a priori*, point n'est besoin d'être Superman pour avoir accès, pendant quelques minutes, à la pièce où réside l'ordinateur.

Cependant, là aussi, dans la situation décrite pour ce cas d'usage, nous sommes en pleine science-fiction. Mais la réalité a parfois tendance à dépasser la fiction…

> La seule protection praticable contre cette attaque est de stocker les programmes de démarrage, dont ce petit dossier non-chiffré (`/boot`), sur un support externe, comme une clé USB, qui sera conservé en permanence dans un endroit plus sûr que l'ordinateur. C'est l'*intégrité* de ces données, et non leur *confidentialité*, qui est alors à protéger. Cette pratique exige pas mal de compétences et de rigueur ; nous ne la developperons pas dans ce guide.
> De telles pratiques mettent la barre plus haut, mais il reste un mais : une fois obtenu l'accès physique à l'ordinateur, si `/boot` n'est pas accessible, et donc pas modifiable, il reste possible d'effectuer le même type d'attaque sur le BIOS de la machine. C'est légèrement plus difficile, car la façon de faire dépend du modèle d'ordinateur utilisé, mais c'est possible. Nous ne connaissons aucune façon de s'en protéger.

Angle d'attaque : les logiciels malveillants

page 23 Nous avons appris dans un chapitre précédent que des logiciels installés à notre insu sur un ordinateur peuvent nous dérober des données. Dans le cas présent, un tel logiciel est en mesure de transmettre la clé de chiffrage du disque dur à un adversaire… qui obtiendra ensuite, grâce à cette clé, l'accès aux données chiffrées, quand il aura accès physique à l'ordinateur.

Installer un logiciel malveillant sur le système Debian dont il est question ici requiert des compétences de plus haut niveau que les attaques étudiées ci-dessus, mais aussi plus de préparation. Une telle attaque relève donc, ici aussi, de la science-fiction, du moins en ce qui concerne la situation qui nous occupe. Dans d'autres situations, il conviendra parfois d'être extrêmement prudent quant à la provenance des données et logiciels qu'on injecte dans l'ordinateur, en particulier lorsqu'il est connecté à Internet… un cas qui, rappelons-le, n'est pas notre propos dans ce premier tome.

page 116 La recette concernant l'installation de logiciels donne quelques pistes fort utiles sur la

façon d'installer de nouveaux logiciels proprement. Le second tome de ce guide, qui sera consacré aux réseaux, et à Internet en particulier, prolongera cette étude.

Angle d'attaque : la force brute

Attaquer un système cryptographique par « force brute » est la plus simple, la plus stupide, et la plus lente des manières. Mais quand on ne peut mettre en œuvre un autre type d'attaque…

Pour le disque dur chiffré lors de l'étape 1, ça demande énormément de temps (de nombreuses années) et/ou énormément d'argent et des compétences pointues.

Ce qu'on peut se dire, c'est qu'*a priori*, si une organisation est prête à mobiliser autant de ressources pour avoir accès à vos données, elle gagnerait amplement à mettre en place une des autres attaques, moins coûteuses et tout aussi efficaces, listées ci-dessus.

Chapitre 9

Cas d'usage : travailler sur un document sensible

9.1 Contexte

Après avoir pris un nouveau départ, l'ordinateur utilisé pour mener ce projet à bien a été équipé d'un système chiffré. Bien. Survient alors le besoin de travailler sur un projet particulier, plus « sensible », par exemple :

page 57
page 105

- un tract doit être rédigé ;
- une affiche doit être dessinée ;
- un livre doit être maquetté puis exporté en PDF ;
- une fuite d'informations doit être organisée pour divulguer les affreuses pratiques d'un employeur ;
- un film doit être monté et gravé sur DVD.

Dans tous ces cas, les problèmes à résoudre sont à peu près les mêmes.

Comme il serait trop pénible d'augmenter globalement, de nouveau, le niveau de sécurité de l'ordinateur, il est décidé que ce projet particulier doit bénéficier d'un traitement de faveur.

Conventions de vocabulaire

Par la suite, nous nommerons :

- les *fichiers de travail* : l'ensemble des fichiers nécessaires à la réalisation de l'œuvre : les images ou *rushes* utilisés comme bases, les documents enregistrés par le logiciel utilisé, *etc.* ;
- l'*œuvre* : le résultat final (tract, affiche, *etc.*)

En somme, la matière première, et le produit fini.

9.2 Évaluer les risques

Partant de ce contexte, tentons maitenant de définir les risques auxquels exposent les pratiques décrites dans ce cas d'usage.

Que veut-on protéger ?

Appliquons au cas présent les catégories définies lorsque nous parlions d'évaluation des risques : page 49

- confidentialité : éviter qu'un œil indésirable ne découvre trop aisément l'œuvre et/ou les fichiers de travail ;
- intégrité : éviter que ces documents ne soient modifiés à notre insu ;
- accessibilité : faire en sorte que ces documents restent accessibles quand on en a besoin.

Ici, accessibilité et confidentialité sont prioritaires.

Accessibilité, car l'objectif principal est tout de même de réaliser l'œuvre. S'il fallait se rendre au pôle Nord pour ce faire, le projet risquerait fort de tomber à l'eau.

Et pour ce qui est de la confidentialité, tout dépend de la publicité de l'œuvre. Voyons donc ça de plus près.

Œuvre à diffusion restreinte

Si le contenu de l'œuvre n'est pas complètement public, voire parfaitement secret, il s'agit de dissimuler à la fois l'œuvre *et* les fichiers de travail.

Œuvre diffusée publiquement

Si l'œuvre a vocation à être publiée, la question de la confidentialité se ramène à celle de l'anonymat.

C'est alors, principalement, les fichiers de travail qui devront passer sous le tapis : en effet, les découvrir sur un ordinateur incite fortement à penser que ses propriétaires ont réalisé l'œuvre… avec les conséquences potentiellement désagréables que cela peut avoir.

Mais ce n'est pas tout : si l'œuvre, ou ses versions intermédiaires, sont stockées sur cet ordinateur (PDF, *etc.*), leur date de création est très probablement enregistrée dans le système de fichiers et dans des méta-données. Le fait que cette date soit antérieure à la publication de l'œuvre peut aisément amener des adversaires à tirer des conclusions gênantes quant à sa généalogie. page 16 page 22

Contre qui veut-on se protéger ?

Pour faire simple, reprenons les possibilités décrites dans le cas d'usage « un nouveau départ » : l'ordinateur utilisé pour réaliser l'œuvre peut être dérobé, plus ou moins fortuitement, par de quelconques flics, voire par de braves voleurs travaillant à leur compte. page 57

9.3 Accro à Windows ?

La première question qui se pose est : quel système d'exploitation utiliser ?

Ça dépend, évidemment, des logiciels utilisés pour ce projet :

S'ils fonctionnent sous GNU/Linux, continuons la lecture de ce chapitre pour étudier les options qui s'offrent à nous.

S'ils fonctionnent exclusivement sous Windows, c'est dommage. Mais nous étudions tout de même un chemin praticable qui permet de limiter la casse. Allons donc voir

page 72 à quoi ressemble ce chemin, en ignorant les paragraphes suivants, qui sont consacrés à GNU/Linux.

9.4 Un tour d'horizon des outils disponibles

Les problèmes attenants à la situation de départ sont les mêmes que ceux du cas d'usage « un nouveau départ ». Mais avant de mettre sur la table de potentielles politiques de sécurité, lançons-nous dans un rapide tour d'horizon des outils et méthodes disponibles. page 57

Liste noire vs. liste blanche

Vu qu'on a déjà un système Debian chiffré, on peut, de prime abord, imaginer le configurer finement pour qu'il conserve moins de traces de nos activités sur le disque dur. Le problème de cette approche, c'est qu'elle est de type « liste noire », et nous en avons expliqué les limites en ces pages : quel que soit le temps consacré, quelle que soit l'expertise mise au travail, même avec une compréhension particulièrement poussée des entrailles du système d'exploitation utilisé, on oubliera toujours une petite option bien cachée, il restera toujours des traces indésirables auxquelles on n'avait pas pensé. page 53

Au contraire, certains systèmes *live* fonctionnent sur le principe de la « liste blanche » : tant qu'on ne le demande pas explicitement, aucune trace n'est laissée sur le disque dur. page 101

En envisageant uniquement le critère « confidentialité », le système *live* bat donc l'autre à plate couture. En termes de temps et de difficulté de mise en œuvre, en revanche, la comparaison est plus mitigée.

Le beurre, ou l'argent du beurre ?

Un système *live* est en effet amnésique ; c'est certes son principal atout, mais cette propriété est aussi source d'inconvénients. Par exemple, dans le cas où notre système *live* préféré ne fournit pas un logiciel donné, qui est pourtant indispensable au projet, il faut, au choix :

- faire du lobbying auprès des auteurs du système *live* pour qu'ils y ajoutent le logiciel souhaité ;
- installer le logiciel dans le système *live* au début de chaque session de travail ;
- fabriquer une version personnalisée de ce système *live*, intégrant ce logiciel, ce qui n'est pas (encore) une opération aisée à l'heure où nous écrivons.

… et il arrive qu'aucune de ces solutions ne soit praticable sans risquer la crise de nerfs.

Pour autant, l'autre hypothèse envisagée, elle non plus, n'est pas de tout repos : limiter les traces laissées par le système Debian chiffré dont il est question — c'est-à-dire : allonger la liste noire des traces indésirables — est une tâche infinie, qui de plus requiert une bonne compréhension du système d'exploitation utilisé, et aux résultats toujours largement insatisfaisants. C'est pourquoi, après avoir poussé fort loin le bouchon dans cette direction, pendant de longues années, les personnes qui ont participé à la rédaction de ce guide sont en passe de jeter l'éponge, et conseillent désormais l'utilisation de systèmes *live* adéquats pour travailler sur des documents sensibles… dans la mesure du possible.

9.5 Quelques pistes pour décider

Tentons maintenant de dissiper la confusion qui a pu être créée par ce tour d'horizon.

Il n'est pas toujours simple de se décider entre ces deux options.

Si les logiciels nécessaires au projet sont installés sur notre système live préféré, alors la réponse est simple : autant l'utiliser. C'est la solution la plus sûre et, dans ce cas, la moins difficile à mettre en place. Auquel cas, allons étudier une politique de sécurité basée là-dessus.

cf. ci-contre

Sinon… ça se complique. Nous avons donné il y a peu trois pistes permettant d'utiliser tout de même un système *live* auquel il manque un logiciel : demander aux auteurs de ce système que le manque soit comblé ; installer le logiciel manquant à chaque démarrage du système *live* ; se concocter une version personnalisée du système *live*. Même si elles demandent des efforts, ces pistes méritent d'être tentées, dans l'ordre où nous les listons. Si l'une d'entre elles fonctionne, la question est réglée, il suffit de mettre en place une politique de sécurité basée sur l'utilisation d'un système *live*.

cf. ci-contre

Si l'hypothèse du système *live* semble, à ce stade, désespérément impraticable, il va falloir se résoudre, bon gré mal gré, à se passer de système *live*… et limiter la casse, autant que faire se peut. Auquel cas, on prendra soin d'étudier attentivement une politique de sécurité à base de système Debian chiffré.

cf. ci-contre

9.6 Travailler sur un document sensible… sur un système *live*

Après avoir présenté le contexte dans le début de ce cas d'usage, et avoir décidé d'utiliser un système *live*, reste à mettre cette solution en place… et à étudier ses limites.

page 65

Le système *live*

Tous les systèmes *live* ne sont pas particulièrement destinés à des pratiques « sensibles ». Il importe donc de choisir un système spécialement conçus pour (tenter de) ne laisser aucune trace sur le disque dur de l'ordinateur sur lequel il est utilisé.

L'outil expliquant comment utiliser un système *live* donne une méthode pour choisir, télécharger et installer un système *live* « discret ».

page 101

La clé USB chiffrée

Toutes les données utilisées au cours de la réalisation du document sensible seront enregistrées sur un support de stockage amovible, tel qu'une clé USB.

La plupart des systèmes d'exploitation gardant des traces, comme le numéro de série, des clés USB qu'on y a branchées. Il est donc préférable de se munir d'une clé USB neuve, qui ne sera jamais connectée à autre chose qu'un système *live*.

Cette clé USB doit être chiffrée. L'opération de chiffrement devra donc, elle aussi, être effectuée à partir du système *live* choisi.

Pour cela, il faut démarrer le système *live* précédemment installé, puis suivre les étapes nécessaires pour réaliser le chiffrement de la clé USB.

page 95
page 143

Limites

Certaines limites, communes à cette méthode et à celle basée sur une Debian chiffrée, sont exposées plus loin.

plus bas
page 78

9.7 Travailler sur un document sensible… sur une Debian chiffrée

Après avoir présenté le contexte dans le début de ce cas d'usage, et décidé, malgré tous les problèmes que ça pose, de ne pas utiliser un système *live*, essayons maintenant de trouver une façon de limiter quelque peu les dégâts.

page 65
page 101

Première étape : d'où l'on part

Après avoir pris un nouveau départ, l'ordinateur utilisé pour mener ce projet a été équipé d'un système chiffré. Considérons aussi que la seconde étape (*Le tiroir de la commode n'était pas chiffré*) de ce nouveau départ a été passée. Restent, en chantier, toutes les étapes suivantes, qui détaillaient les principales attaques possibles dans cette situation. Tentons maintenant de faire face à certaines d'entre elles.

page 57
page 105
page 57

Seconde étape : des tiroirs troués, mais des tiroirs tout de même

La troisième étape du nouveau départ suggérait d'avoir un disque dur suffisamment « propre » pour que ce ne soit pas la catastrophe si on cède face à la loi, ou si le système cryptographique utilisé est cassé… et c'est ce que nous allons tenter de faire.

page 57

En guise de premier jet, voici une méthode possible :

1. Avant de commencer à travailler sur ce projet particulier, créer un nouvel utilisateur, qui lui sera dédié, sur le système Debian utilisé.
2. Toute séance de travail sur ce projet devra avoir lieu en se connectant, sur le système, en tant que cet utilisateur dédié.
3. Tout fichier de travail lié à ce projet sera stocké sur un support amovible et chiffré (clé USB, disque dur externe).
4. À la fin du projet, du ménage sera effectué :
   - archiver les fichiers qui en valent la peine ; ces fichiers sont certes déjà stockés sur un support externe chiffré, mais d'autres aspects entrent en ligne de compte : il est donc nécessaire de se référer malgré tout au cas d'usage ;
   - effacer « pour de vrai » le support externe chiffré ;
   - supprimer du système l'utilisateur dédié au projet ;
   - effacer « pour de vrai » les fichiers appartenant à cet utilisateur présents sur le disque dur du système ;
   - effacer « pour de vrai » l'espace libre du disque dur.

page 79

En procédant ainsi, **la plupart** des traces **les plus évidentes** de ce projet sont séparées du reste du système :

- les fichiers de travail sont stockés sur un support externe chiffré, qui peut être convenablement « rangé » quand il ne sert pas ;
- les fichiers de configuration de l'utilisateur dédié, ainsi qu'une bonne partie de l'historique de ses activités, sont stockés dans son dossier personnel.

Ces deux emplacements étant convenablement nettoyés lorsque le projet est achevé, **si** la catastrophe (céder face à la loi, découverte d'un problème dans le système cryptographique) arrive **après coup**, les traces résiduelles sur le disque dur seront moins évidentes, et moins nombreuses, que si l'on avait procédé de façon ordinaire.

Pour mettre en place une telle méthode de travail, il faudra se référer, après avoir terminé la lecture du présent cas d'usage, aux recettes et cas d'usage suivants :

- créer un utilisateur, voir page 153 ;
- chiffrer une clé USB, voir page 143 ;
- archiver un projet achevé, voir page 79 ;
- supprimer un utilisateur, voir page 155 ;
- effacer « pour de vrai », voir page 127.

Troisième étape : troués, nos tiroirs ?

Examinons maintenant les trous qui fragilisent les tiroirs de l'étape précédente.

Si la catastrophe arrive pendant la réalisation du projet

Si la catastrophe arrive pendant la réalisation du projet, le disque dur de l'ordinateur utilisé ne contient certes pas, explicitement, les fichiers de travail, mais l'ensemble des traces qu'il contient, sans faire preuve, suffit probablement à quiconque [1] pour bâtir une intime conviction sur la nature du travail effectué.

1. La boulangère du village, le journaliste du Figaro, voire… un juge.

Si la catastrophe arrive plus tard

Même si la catastrophe arrive après la fin du projet, c'est-à-dire : après le nettoyage conseillé ici, il serait malvenu de se sentir immunisé, car comme le début de ce cas d'usage l'explique, l'inconvénient majeur de la méthode décrite ici est qu'elle est basée sur le principe de liste noire, principe abondamment décrié en ces pages… et il restera donc toujours des traces indésirables, auxquelles on n'avait pas pensé, sur le disque dur de l'ordinateur utilisé, en plus de celles qu'on connait bien désormais : journaux, mémoires vive et « virtuelle », sauvegardes automatiques.

page 65
page 53
page 19

… et c'est encore plus compliqué que ça

Certaines limites, communes à cette méthode et à celle basée sur un système *live*, sont exposées plus loin.

page 69
page 78

9.8 Travailler sur un document sensible… sous Windows

page 65 Après avoir présenté le contexte dans le début de ce cas d'usage et décidé, malgré tous les problèmes que ça pose, d'utiliser Windows, essayons maintenant de trouver une façon de limiter quelque peu la casse.

Point de départ : une passoire et une boîte de rustines desséchées

Partons d'un ordinateur muni, de la façon la plus classique qui soit, d'un disque dur sur lequel Windows est installé. Nous ne nous appesantirons pas sur cette situation, la première partie de cet ouvrage ayant abondamment décrit les multiples problèmes qu'elle pose. Une passoire, en somme, pleine de trous de sécurité.

On peut donc imaginer coller quelques rustines sur cette passoire. Faisons-en rapidement le tour.

Un disque dur, ça se démonte et ça se cache. Certes. Mais il y a les périodes où l'on s'en sert, parfois plusieurs jours ou semaines d'affilée. Cette rustine est basée sur deux hypothèses quelque peu osées :

- *Nous avons de la chance.* Il suffit en effet que l'accident (perquisition, cambriolage, *etc.*) survienne au mauvais moment pour que toute la confidentialité désirée soit réduite à néant ;
- *Notre discipline est parfaitement rigoureuse.* En effet, si l'on oublie, ou qu'on ne prend pas le temps, d'aller « ranger » le disque dur quand on n'en a plus besoin, et que l'accident survient à ce moment-là, c'est perdu, fin de la partie.

Par ailleurs, des outils existent pour chiffrer des données sous Windows. Quelle que soit la confiance qu'on leur accorde, il n'en reste pas moins qu'ils s'appuient obligatoirement sur les fonctions offertes par la boîte noire qu'est Windows. On ne peut donc que s'en méfier, et dans tous les cas, Windows, lui, aura accès à nos données *en clair*, et personne ne sait ce qu'il pourrait bien en faire.

Pour conclure ce petit tour dans la cour des miracles douteux, ajoutons que la seule « solution » possible dans le cas présent serait une approche de type liste noire, dont page 51 l'inefficacité crasse a déjà été expliquée précédement.

Il est maintenant temps de passer aux choses sérieuses.

Seconde étape : enfermer Windows dans un compartiment (presque) étanche

Ce qui commence à ressembler à une solution sérieuse, ce serait de faire fonctionner Windows dans un compartiment étanche, dans lequel on ouvrirait, quand c'est nécessaire et en connaissance de cause, une porte pour lui permettre de communiquer avec l'extérieur de façon strictement limitée.

En d'autres termes, mettre en place une solution basée sur une logique de type *liste blanche* : rien ne pourrait entrer dans Windows ou en sortir *a priori*, et à partir de cette règle générale, on autorise des *exceptions*, au cas par cas, en réfléchissant à leur impact.

La *virtualisation*[2] permet de mettre en place ce type de systèmes. C'est un ensemble de techniques matérielles et logicielles qui permettent de faire fonctionner, sur un seul ordinateur, plusieurs systèmes d'exploitation, séparément les uns des autres, (presque) comme s'ils fonctionnaient sur des machines physiques distinctes.

2. Pour plus d'informations, voir la page de Wikipédia sur le sujet [https://secure.wikimedia.org/wikipedia/fr/wiki/Virtualisation].

Il est ainsi relativement facile, de nos jours, de faire fonctionner Windows **à l'intérieur** d'un système GNU/Linux, en lui coupant, par la même occasion, tout accès au réseau — et en particulier, en l'isolant d'Internet.

⚠ **Attention** : il est conseillé de lire l'intégralité de ce chapitre **avant** de se précipiter sur les recettes pratiques ; la description de l'hypothèse qui suit est assez longue, et ses limites sont étudiées à la fin de ce chapitre, où des contre-mesures sont envisagées. Il serait quelque peu dommage de passer quatre heures à suivre ces recettes, avant de se rendre compte qu'une toute autre solution serait, en fait, plus adéquate.

Commençons par résumer l'hypothèse proposée.

L'idée est donc de faire fonctionner Windows dans un compartiment *a priori* étanche, **à l'intérieur** d'un système Debian chiffré tel que celui qui a pu être mis en place à la suite de la lecture du cas d'usage précédent. Ce qui servira de disque dur à Windows, c'est en fait un gros fichier, rangé à côté de tous nos autres fichiers, sur le disque dur de notre système Debian chiffré. Ce fichier, qui n'a vraiment rien de particulier, nous le nommons une *image de disque virtuel*, parfois abrégé par une *image de disque*.

page 57

Le fait que ce pseudo-disque dur soit un fichier nous simplifiera grandement la vie par la suite, qui décrit plus précisément la procédure envisagée.

Installer VirtualBox

La recette « installer Virtual Box » explique comment installer le logiciel VirtualBox, qui nous servira à lancer Windows dans un compartiment étanche.

page 168

Installer un Windows « propre » dans VirtualBox

Préparons une image de disque virtuel *propre* : la recette « installer un Windows virtualisé » explique comment installer Windows dans VirtualBox en lui coupant, **dès le départ**, tout accès au réseau.

page 170

À partir de ce moment-là, on qualifie Windows de système *invité* par le système Debian chiffré, qui, lui, est le système *hôte*.

Installer les logiciels nécessaires dans le Windows « propre »

Autant installer, dès maintenant, dans le Windows « propre », tout logiciel *non compromettant* nécessaire à la réalisation des œuvres préméditées : ça évitera de le refaire au début de chaque nouveau projet… et ça évitera, souhaitons-le ardemment, d'utiliser une image Windows « sale » pour un nouveau projet, un jour où le temps presse.

Vu que le Windows *invité* n'a pas le droit de sortir de sa boîte pour aller chercher lui-même des fichiers, il est nécessaire de lui faire parvenir depuis « l'extérieur » les fichiers d'installation des logiciels nécessaires.

Une telle opération sera aussi utile, par la suite, pour lui envoyer toutes sortes de fichiers, et nous y reviendrons. Pour l'heure, vu que nous sommes en train de préparer une image de Windows « propre », servant de base à chaque nouveau projet, ne mélangeons pas tout, et contentons-nous de lui envoyer uniquement ce qui est nécessaire à l'installation des logiciels non compromettants souhaités.

Créons, sur le système hôte, un dossier nommé *Logiciels Windows*, et copions-y **uniquement** les fichiers nécessaires à l'installation des logiciels souhaités.

Puis partageons ce dossier avec le Windows *invité*, en cochant la case *Lecture seule*, et sans rendre ce partage permanent ; la recette « envoyer des fichiers au système virtualisé » explique comment procéder pratiquement.

page 176

Et en ce qui concerne l'installation des logiciels à l'intérieur du Windows *invité* : toute personne suffisamment accro à Windows pour lire ces pages est, sans aucun doute, plus compétente que celles qui écrivent ces lignes.

⚠ **Attention** : une fois cette étape effectuée, il est impératif de ne **rien** faire d'autre dans ce Windows virtualisé.

Congeler le Windows « propre »

Congelons maintenant l'image de disque *propre* qui vient d'être préparée, c'est-à-dire : sauvegardons-la dans un coin, telle quelle, et on ne démarrera plus **jamais** dessus. Par la suite, elle ne servira plus que de base de départ.

page 172 La recette sauvegarder une image de disque virtuel, explique comment effectuer cette opération.

Nouveau projet, nouveau départ

Mettons qu'un nouveau projet nécessitant l'utilisation de Windows débute ; voici ce qui se passe :

1. l'image de disque propre est clonée, pour donner naissance à une nouvelle image de disque, en tout point identique ; c'est la *décongélation* ;
2. la nouvelle image de disque, issue de la décongélation, peut maintenant être démarrée dans son compartiment étanche ; elle servira **exclusivement** pour le nouveau projet, et devient désormais une image *sale* ;
3. au sein de cette nouvelle image *sale*, un nouvel utilisateur Windows est créé ; le nom qui lui est attribué doit être différent **à chaque fois** qu'un nouveau projet est ainsi démarré, et cet utilisateur servira **exclusivement** pour ce nouveau projet. Ceci, parce que les logiciels tendent à inscrire le nom de l'utilisateur actif dans les méta-données des fichiers qu'ils enregistrent, et qu'il vaut mieux éviter de rendre possibles de fâcheux recoupements.

page 22

page 174 La recette créer une nouvelle machine virtuelle à partir d'une image propre explique les détails techniques de la chose.

Maintenant que nous avons un compartiment étanche, voyons comment y ouvrir des portes sélectivement, en fonction des besoins.

Comment envoyer des fichiers au Windows embastillé ? Vu que le Windows *invité* n'a pas le droit de sortir de sa boîte pour aller chercher lui-même des fichiers, il peut être nécessaire de lui en faire parvenir depuis « l'extérieur », par exemple :

- de la matière première (*rushes*, images ou textes provenant d'autres sources) ;
- un logiciel nécessaire au nouveau projet, et absent de l'image virtuelle *décongelée*.

Nous avons déjà vu comment procéder, mais c'était dans un cas très particulier : l'installation de nouveaux logiciels dans un Windows « propre » *invité*. Partager des fichiers avec un Windows « sale » requiert davantage de réflexion et de précautions, que nous allons maintenant étudier.

La façon de faire est légèrement différente, en fonction du support sur lequel se trouvent, à l'origine, les fichiers à importer (CD, DVD, clé USB, dossier présent sur le disque dur du système chiffré), mais les précautions d'usage sont les mêmes :

- Windows doit **uniquement** avoir accès aux fichiers qu'on veut y importer, et c'est tout. Il n'est pas question de lui donner accès à un dossier qui contient, pêle-mêle, des fichiers concernant des projets qui ne devraient pas être recoupés entre eux. Si ça implique de commencer par une phase de tri et de rangement, et bien, soit.

- Lorsque Windows a besoin de *lire* (recopier) les fichiers contenus dans un dossier, on lui donne **uniquement** accès en *lecture* à ce dossier. Le moins on donne le droit à Windows d'écrire ici ou là, le moins il laissera de traces gênantes.

À noter que, lorsqu'on décide de partager un dossier du système hôte avec un Windows *invité*, VirtualBox propose de rendre ce partage permanent. Ça évite de refaire la manipulation à chaque fois qu'il est nécessaire d'envoyer un fichier au Windows *invité*, mais ça implique le risque de déposer des fichiers dans ce dossier sans penser qu'ils pourront être lisibles par Windows et ses sbires.

C'est pourquoi, afin d'éviter de se mélanger les pinceaux, nous recommandons de :

- créer **un** dossier d'importation par projet ;
- nommer ce dossier de façon aussi explicite que possible ; par exemple : *Dossier lisible par Windows* ;
- ne jamais partager d'autres dossiers que celui-ci avec le Windows *invité*.

La recette « envoyer des fichiers au système virtualisé » explique comment procéder pratiquement. page 176

Comment faire sortir des fichiers du Windows embastillé ? Le Windows *invité* n'a pas le droit, par défaut, de laisser des traces en dehors de son compartiment étanche. Mais presque inévitablement vient le temps où il est nécessaire d'en faire sortir des fichiers, et à ce moment-là, il nous faut l'autoriser explicitement, par exemple :

- pour emmener à la boîte-à-copies, ou chez l'imprimeur, un fichier PDF exporté ;
- pour projeter, sous forme de DVD, le film fraîchement réalisé.

Lorsqu'on doit récupérer un CD ou DVD non chiffré, et que la machine hôte est munie d'un graveur, il suffit de « prêter » ce périphérique, temporairement, au Windows *invité*, afin de graver depuis ce système.

Dans le cas où rien n'oblige à récupérer les fichiers sur un support non chiffré, il est possible de les exporter vers un dossier vide, dédié à cet usage, et stocké sur un volume chiffré qui peut être :

- une clé USB chiffrée, qu'on active sous Debian en tapant la phrase de passe correspondante ;
- le disque dur de la Debian chiffrée qui fait ici office de système *hôte*.

Ce dossier dédié sera partagé, via VirtualBox, avec le Windows *invité*. Insistons sur les mots **vide** et **dédié** : Windows pourra lire et modifier tout ce que ce dossier contient, et il serait dommageable de lui permettre de lire des fichiers, quand on a seulement besoin d'exporter un fichier.

Afin d'éviter de se mélanger les pinceaux et de limiter la contagion, nous recommandons de :

- créer **un** dossier d'exportation par projet ;
- nommer ce dossier de façon aussi explicite que possible ; par exemple : *Dossier où Windows peut écrire* ;
- ne jamais partager d'autres dossiers que celui-ci avec le Windows *invité*, mis à part le dossier d'importation que le paragraphe précédent préconise.

Les recettes « récupérer des fichiers depuis un système virtualisé » et « chiffrer une clé USB » expliquent comment procéder pratiquement. page 178 page 143

Quand le projet est terminé

Quand ce projet est terminé, il faut faire le ménage, mais avant toute chose :

1. l'œuvre résultante est exportée sur le support approprié (papier, VHS, *etc.*), en s'aidant du paragraphe précédent, qui explique comment faire sortir des fichiers du Windows *invité* ;
2. les fichiers de travail sont, si nécessaire, archivés (le cas d'usage suivant traitant, quelle coïncidence, de la question).

page 79

Puis vient l'heure du grand ménage, qui éliminera du système *hôte* le plus possible de traces du projet achevé :

- l'image de disque sale est retirée de VirtualBox et effacée « pour de vrai » ;
- le dossier d'importation est effacé « pour de vrai » ;
- le dossier d'exportation est effacé « pour de vrai »… après avoir vérifié, une dernière fois, que tout ce qui doit être conservé a bien été archivé ailleurs.

Les recettes « effacer des images de disque » et « effacer des fichiers » expliquent comment effectuer ces opérations.

page 173
page 129

Encore un nouveau projet ?

Si un nouveau projet survient, nécessitant lui aussi d'utiliser Windows, ne réutilisons **pas** le même Windows *sale*. Retournons plutôt à l'étape « nouveau projet, nouveau départ ».

Troisième étape : attaques possibles et contre-mesures

L'hypothèse que nous venons de décrire est basée sur l'utilisation, comme système *hôte*, de la Debian chiffrée mise en place à la première étape du cas d'usage « un nouveau départ ». Toutes les attaques concernant cette Debian chiffrée sont donc applicables à la présente solution. Il est donc maintenant temps d'étudier le cas d'usage correspondant, en particulier à partir de la seconde étape.

page 58
page 69

De retour ? Bien.

Si, malgré ces soucis, l'hypothèse que nous venons de décrire semble être un compromis acceptable, il est maintenant nécessaire de se renseigner sur les limites partagées par toutes les solutions envisagées dans ce cas d'usage.

page 78

Sinon, creusons un peu.

Admettons qu'une des attaques décrites à partir de la troisième étape du cas d'usage « un nouveau départ » semble crédible. Si elle réussissait, le contenu du disque dur chiffré du système *hôte* serait lisible, en clair, par l'attaquant. Or nos fichiers de travail sont, rappelons-le, contenus dans l'image de disque virtuel utilisée par notre Windows *invité*… qui est un bête fichier stocké sur le disque dur du système *hôte*. Ces fichiers de travail, ainsi que toute trace enregistrée par les logiciels utilisés dans Windows, deviennent alors lisibles par l'attaquant.

page 60

Nous allons envisager deux pistes permettant de limiter les dégâts. L'une est de type « liste noire », l'autre est de type « liste blanche ».

Stocker l'image de disque virtuel en dehors du disque du système *hôte*

Une idée est de stocker hors du disque dur du système *hôte* l'image de disque virtuel utilisée par le système Windows *invité*. Par exemple, sur un disque dur externe chiffré. Ainsi, même si le disque du système *hôte* est déchiffré, nos fichiers de travail restent inaccessibles… pourvu que le disque dur externe qui les contient soit, à ce moment-là, convenablement « rangé ».

Cette approche est de type « liste noire », avec tous les problèmes que ça pose. Les

page 53

fichiers de travail et le système Windows sont certes extraits du disque dur du système *hôte*, mais il ne faut pas oublier une chose : ces données seront utilisées par un logiciel animé par le système *hôte*, nommément : VirtualBox. Comme le chapitre « traces à tous les étages » l'explique, diverses traces subsisteront donc, inévitablement, sur le disque dur **interne** de l'ordinateur utilisé.

page 19

Pour suivre cette piste :

- se renseigner sur les limites partagées par toutes les solutions envisagées dans ce cas d'usage ;
- se reporter à la recette permettant de chiffrer un disque dur externe.

page suivante
page 143

Utiliser un système *live* comme système *hôte*

Le pendant de cette approche « liste noire » est une solution de type « liste blanche », conjuguant l'utilisation d'un système *live*, et le stockage de l'image de disque virtuel sur un disque dur externe chiffré.

Pour suivre cette piste :

- se renseigner sur les limites partagées par toutes les solutions envisagées dans ce cas d'usage ;
- se reporter à la recette permettant de chiffrer un disque dur externe, et à celle qui explique comment utiliser un système *live*.

page suivante
page 143
page 101

9.9 Limites communes à ces politiques de sécurité

Toute politique de sécurité étudiée dans ce cas d'usage est vulnérable à un certain nombre d'attaques. Ce, qu'elle soit basée sur un système Debian chiffré, un système *live*, ou sur l'envoûtement de l'infâme Windows.

page 57 Les étapes 4 et 5 du nouveau départ étudient certaines des attaques imaginables, relevant plus ou moins de la science-fiction, selon l'époque, le lieu, les protagonistes et les circonstances. Le moment est venu de les relire d'un œil nouveau.

page 7 Par ailleurs, la partie « problématiques » de ce tome abordait, de façon relativement générale, de nombreux modes de surveillance, qu'il peut être bon de réétudier à la lumière de la situation concrète qui nous occupe ; nommons en particulier les questions d'électricité, champs magnétiques et ondes radios, ainsi que les effets des divers mouchards.

page 14
page 23

Chapitre 10

Cas d'usage : archiver un projet achevé

10.1 Contexte

Un projet sensible touche à sa fin ; par exemple, un livre a été maquetté et imprimé, un film a été monté, compressé, et gravé sur DVD.

page 65

En général, il ne sera dès lors plus nécessaire de pouvoir accéder en permanence aux fichiers de travail (iconographie en haute résolution, *rushes* non compressés). Par contre, il peut être utile de pouvoir les retrouver plus tard, par exemple pour une réédition, une version mise à jour…

Vu qu'un système est d'autant plus susceptible d'être *attaqué* qu'il est fréquemment utilisé, autant extraire les informations rarement utilisées de l'ordinateur utilisé quotidiennement. De surcroît, il est plus facile de nier tout lien avec des fichiers, lorsqu'ils sont stockées sur une clé USB au fond d'un bois, que lorsqu'ils sont rangés sur le disque dur de l'ordinateur familial.

10.2 Est-ce bien nécessaire ?

La première question à se poser avant d'archiver de tels fichiers est la suivante : est-il *vraiment* nécessaire de les conserver ? Lorsqu'on ne dispose plus *du tout* d'une information, quiconque aura beau insister, personne ne sera en mesure de la donner, et c'est parfois la meilleure solution.

10.3 Évaluer les risques

Que veut-on protéger ?

Que donnent les catégories définies lorsque nous parlions d'évaluation des risques, appliquées à ce cas ?

page 49

- confidentialité : éviter qu'un œil indésirable ne tombe trop aisément sur les informations archivées ;
- intégrité : éviter que ces informations ne soient modifiées à notre insu ;
- accessibilité : faire en sorte que ces informations restent accessibles quand on en a besoin.

Ici, l'accessibilité est secondaire par rapport à la confidentialité : toute l'idée de l'archivage est de faire un compromis, en rendant l'accès aux données plus difficile *pour tout le monde*, afin de leur offrir une meilleur confidentialité.

Contre qui veut-on se protéger ?

page 57

Les risques envisagés dans notre « nouveau départ » sont valables ici aussi : un cambriolage, une perquisition ayant des motifs qui ne sont pas directement liés aux informations qu'on veut ici protéger.

Ajoutons, à ces risques, la possibilité que le livre ou le film produit déplaise à quelque commissaire, ministre, P.D.G. ou assimilé. Ça arrive. Admettons que :

- cette autorité a eu vent d'indices lui permettant de soupçonner qui a commis le chef d'œuvre ;
- cette autorité est en mesure de mandater une cohorte de pénibles hommes en armes et uniforme, au petit matin et au domicile des personnes soupçonnées.

Une telle inopportune intrusion débouchera au minimum, de façon tout aussi fâcheuse qu'évidente, sur la saisie de tout matériel informatique qui pourra y être découvert. Ce matériel sera ensuite remis, par les intrus, à un autre homme de main des autorités, qui pratiquera un genre d'autopsie visant à mettre au jour les données stockées sur ce matériel… ou l'ayant été.

page 31

10.4 Méthode

La méthode la plus simple à l'heure actuelle est :

page 143

1. créer une clé USB ou un disque dur externe chiffré ;
2. copier les fichiers à archiver vers ce périphérique ;

page 127

3. écraser le contenu des fichiers de travail.

Une fois ces opérations effectuées, la clé ou le disque dur pourra être entreposé dans un autre lieu que l'ordinateur utilisé couramment.

On pourrait envisager l'utilisation de CD ou de DVD, pour leur faible coût, mais à l'heure actuelle, il est plus complexe de chiffrer correctement des données sur ces supports que sur des clés USB, qui sont désormais monnaie courante et faciles à se procurer.

10.5 Quelle phrase de passe ?

page 93

Vu que les fichiers seront archivés sous forme chiffrée, il sera nécessaire de choisir une phrase de passe. Or, vu que la vocation est l'archivage, cette phrase de passe ne sera pas souvent utilisée. Et une phrase de passe rarement utilisée a toutes les chances d'être oubliée… rendant impossible l'accès aux données.

Face à ce problème, on peut envisager quelques pistes.

Écrire la phrase de passe quelque part

Toute la difficulté étant de savoir où l'écrire, ranger ce document pour pouvoir le retrouver… sans pour autant que d'autres puissent le retrouver et l'identifier comme une phrase de passe.

Utiliser la même phrase de passe que pour son système quotidien

page 105

La phrase de passe de son système quotidien, dans le cas où il est chiffré, est une phrase qu'on tape régulièrement, et dont on a toutes les chances de se souvenir.

Par contre :

- si on est forcé de révèler la phrase de passe commune, l'accès à l'archive devient également possible ;
- il est nécessaire d'avoir **très fortement** confiance dans les ordinateurs avec lequel on accèdera aux archives. Sinon, on peut se faire « piquer », à son insu, la phrase de passe, qui pourra ensuite être utilisée pour lire non seulement les informations archivées, mais aussi toutes les données stockées sur l'ordinateur.

Partager le secret à plusieurs

Il est possible de partager un secret à plusieurs. Cela impose de réunir plusieurs personnes afin de pouvoir accéder au contenu archivé. C'est à peser : ça peut compliquer la tâche aussi bien pour des accès désirés qu'indésirables.

page 159

10.6 Un disque dur ? Une clé ? Plusieurs clés ?

Selon les choix faits précédemment, entre autres sur la phrase de passe, on peut se demander quels supports utiliser. Sachant que sur le plan technique, le plus simple actuellement est d'avoir une seule phrase de passe par support.

Un disque dur externe peut contenir plus de données qu'une clé USB, et est donc parfois nécessaire : pour archiver un projet de vidéo, par exemple.

Archiver plusieurs projets sur un même support permet de se simplifier la tâche, mais il devient alors difficile de séparer les projets selon les niveaux de confidentialité souhaités. Qui plus est, en procédant ainsi, les personnes pouvant accéder aux archives d'un projet ont aussi accès aux autres, ce qui n'est pas forcément souhaitable.

Par ailleurs, si la phrase de passe est un secret partagé, autant faciliter l'accès aux personnes partageant le secret, en ayant un support qu'elles peuvent se transmettre.

Troisième partie

Outils

Dans cette troisième partie, nous expliquerons comment appliquer concrètement quelques-unes des pistes évoquées précédemment.

Cette partie n'est qu'une annexe technique aux précédentes : une fois comprises les problématiques liées à l'intimité dans le monde numérique ; une fois les réponses adaptées choisies, reste la question du « Comment faire ? », à laquelle cette annexe apporte certaines réponses.

Du bon usage des recettes

Les outils et recettes qui suivent sont des solutions extrêmement partielles, qui ne sont d'aucune utilité tant qu'elles ne font pas partie d'un ensemble de pratiques articulées de façon cohérente.

Piocher dans cette boîte à outils sans avoir, au préalable, étudié la partie sur le choix d'une réponse adaptée et défini une *politique de sécurité*, est un moyen remarquable de se tirer une balle dans le pied en croyant, à tort, avoir résolu tel ou tel problème. page 47

On ne peut pas faire plaisir à tout le monde

Partons du principe, pour la plupart des recettes présentées dans ce guide, que l'on utilise GNU/Linux avec le bureau GNOME ; elles ont été écrites et testées sous Debian GNU/Linux version 5.0 (surnommée Lenny) [1] et *The (Amnesic) Incognito Live System* [2].

Pour autant, ces recettes sont généralement concoctables avec d'autres distributions basées sur Debian, telles qu'Ubuntu [3] ou gNewSense [4].

Si l'on n'utilise pas encore GNU/Linux, ou pourra consulter le cas d'usage un nouveau départ ou utiliser un système *live*. page 57 page 101

1. http://www.debian.org/releases/lenny/
2. https://amnesia.boum.org/
3. http://www.ubuntu-fr.org/
4. http://www.gnewsense.org/Main:fr/HomePage

De la bonne interprétation des recettes

Avant de passer aux recettes elles-mêmes, quelques remarques transversales nous ont paru nécessaires.

Dans un certain nombre d'outils, les procédures sont présentées pas à pas, et expliquent, chaque fois que c'est possible, le sens des actions que l'on propose d'effectuer. Une utilisation efficace de ces outils nécessite de s'entendre sur quelques points :

- L'ordre dans lequel chaque recette est développée est d'une importance capitale. Sauf mention contraire, il est simplement inimaginable de sauter une étape pour ensuite revenir en arrière : le résultat, si jamais ces opérations désordonnées en donnaient un, pourrait être soit différent de celui escompté, soit tout bonnement catastrophique.
- Dans le même ordre d'idée, les actions indiquées doivent être effectuées à la lettre. Omettre une option, ouvrir le mauvais dossier, peut avoir pour effet de totalement modifier le sens ou les effets d'une recette.
- De manière générale, la bonne compréhension de ces recettes demande d'y accorder un minimum d'attention et de vivacité d'esprit. On ne peut pas tout réexpliquer à chaque fois : il est implicite d'avoir auparavant « suivi » et intégré les explications des « cas d'usage » dont ces recettes ne sont que la dernière étape.

Chapitre 11

Utiliser un terminal

Souvent, on utilise un ordinateur personnel en cliquant sur des menus et des icônes. Cependant, il existe une autre façon de lui « parler » : en tapant des bouts de texte que l'on appelle des « commandes ». On appelle cette façon d'interagir avec un ordinateur « le terminal », « le *shell* » ou encore « la ligne de commande ».

Ce guide cherche le plus souvent possible à contourner l'utilisation de cet outil, qui est assez déroutant lorsque l'on n'y est pas habitué. Cependant, son usage s'est parfois avéré indispensable.

11.1 Qu'est-ce qu'un terminal ?

Une explication détaillée sur l'usage de lignes de commandes n'est pas l'objet de ce guide, et Internet regorge de tutoriels et de cours assurant très bien ce rôle [1]. Il semblait cependant nécessaire de poser quelques bases sur la manière de s'en servir.

Alors on va tout simplement commencer par ouvrir un terminal : sur un bureau GNOME standard, il suffit de cliquer sur *Applications* → *Accessoires* → *Terminal*. Apparaît alors une fenêtre qui indique :

```
IDENTIFIANT@LE_NOM_DE_LA_MACHINE:~$
```

À la fin se trouve un carré, appelé « curseur », qui correspond à l'endroit où inscrire le texte de la commande. Concrètement, avec l'identifiant *roger* sur une machine nommée *debian*, on aura sous les yeux :

```
roger@debian:~$ 
```

C'est à partir de cet état, appelé « invite de commande », que l'on peut taper directement les commandes qu'on veut faire exécuter à l'ordinateur.

L'effet final de ces commandes est souvent le même que celui qu'on peut obtenir en cliquant au bon endroit dans une interface graphique.

Par exemple, si dans le terminal qu'on vient d'ouvrir, on écrit juste `gedit` puis qu'on tape sur *Entrée*, le résultat est qu'on ouvre un éditeur de texte. On aurait pu faire exactement la même chose en cliquant sur *Applications* → *Accessoires* → *Éditeur de texte*. Par contre, on ne pourra pas entrer de nouvelle commande dans notre terminal tant que l'on aura pas quitté l'éditeur de texte.

Dans le cadre de ce guide, l'intérêt du terminal est surtout qu'il permet d'effectuer des actions qu'aucune interface graphique ne propose pour le moment.

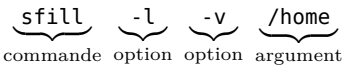
1. Entre autres, une page sur *ubuntu-fr.org* [http://doc.ubuntu-fr.org/console] qui se termine elle-même par d'autres liens.

11.2 À propos des commandes

Les commandes sont comme des ordres qu'on donne à l'ordinateur par le biais du terminal. Ces « lignes de commande » ont leur propre langage, avec leurs mots, leurs lettres, et leur syntaxe. Quelques remarques à ce sujet sont donc utiles.

Syntaxe

page 138 Un exemple, tiré d'un outil que l'on lira plus tard :



Dans cette ligne de commande, on peut voir, dans l'ordre :

- la *commande* que l'on appelle est `sfill`. La commande est en général un programme installé sur le système ;
- deux *options*, `-l` et `-v` qui modifient le comportement du programme `sfill`. Ces dernières peuvent être facultatives selon le programme (et commencent par un ou deux tiret pour qu'on les distingue) ;
- un *argument* `/home` qui précise ce sur quoi va travailler la commande. Il peut y en avoir plusieurs, ou aucun, tout dépend de la commande.

Chacun de ces éléments doit être séparés des autres par un (ou plusieurs) espace. Il y a donc un espace entre la commande et la première option, entre la première option et la suivante, entre la dernière option et le premier argument, entre le premier argument et les suivants, *etc.*

Pour savoir quelles sont les commandes disponibles, leurs options et leurs argumets, pas de mystères : chaque commande dispose normalement d'une page de manuel. Pour y accéder, il suffit d'aller dans *Système* → *Aide*, puis dans *Pages de manuel*. Ces dernières peuvent toutefois être difficile à comprendre par leur aspect techique, et parfois ne sont disponibles qu'en anglais.

Insertion du chemin d'un fichier

Lors de l'utilisation d'un terminal, on a souvent besoin d'indiquer des dossiers et des fichiers. On parle de « chemin » car on décrit généralement dans quel dossier et sous-dossier un fichier se trouve. Pour séparer un dossier de ce qu'il contient, on utilise le caractère `/` (qui se prononce « slash »).

Pour donner un exemple, voici le *chemin* du document `recette.txt` qui se trouve dans le dossier `Documents` du dossier personnel du compte `alligator` :

```
/home/alligator/Documents/recette.txt
```

Comme beaucoup de commandes attendent des noms de fichiers comme arguments, cela devient vite fastidieux de taper leurs chemins complets à la main. Il y a cependant un moyen simple d'insérer un chemin : quand on attrape avec la souris l'icône d'un fichier, et qu'on le déplace pour le lâcher sur le terminal, son chemin s'écrit là où se trouve le curseur.

Cela ne marche cependant qu'avec les vrais fichiers ou dossiers. On obtiendra un nom bizarre qui ne fonctionnera pas, par exemple, pour les fichiers mis à la corbeille, l'icône du *Dossier personnel* sur le bureau ou avec les icônes de clés USB.

Exécution

Une fois que l'on a tapé une commande, on demande à l'ordinateur de l'« exécuter » en appuyant sur la touche *Entrée*.

Fin ou interruption de la commande

L'exécution de la commande prend plus ou moins de temps. Lorsqu'elle est terminée, le terminal retourne toujours à l'état où il était avant qu'on lance la commande, l'« invite de commande » :

```
roger@debian:~$
```

On dit alors que le terminal « rend la main ».

Si on souhaite interrompre l'exécution d'une commande avant qu'elle soit terminée, on peut appuyer la touche Ctrl, et tout en laissant cette touche enfoncée appuyer sur la touche C. On arrête alors la commande immédiatement, un peu comme quand on ferme la fenêtre d'un programme.

Typographie

La plupart des symboles utilisés pour entrer les commandes complètes sont des symboles courants. Lorsqu'une commande emploie le symbole « - », il ne s'agit que du « tiret » qu'on peut obtenir en tapant (sur un clavier français) la touche 6. Pour un « ' » (apostrophe droite), c'est le 4 …

D'autres symboles sont rarement utilisés en-dehors du terminal, mais sont disponibles avec les claviers standards. Ils sont mêmes indiqués sur le clavier, et accessibles à l'aide de la touche Alt de droite, notée AltGr. Voici, en se basant sur un clavier de PC français standard, la correspondance de quelques touches avec les symboles qu'elles écrivent, et leur nom (bien peu seront en fait utilisées dans ce guide) :

Touches	Résultat	Nom du symbole
AltGr + 2	~	tilde
AltGr + 3	#	dièse
AltGr + 4	{	accolade gauche
AltGr + 5	[	crochet gauche
AltGr + 6	|	*pipe*
AltGr + 8	\	antislash
AltGr + 0	@	arobase
AltGr +)	]	crochet droit
AltGr + =	}	accolade droite

Noms à remplacer

Parfois, on précise que l'on va nommer quelque chose que l'on a trouvé pour le réutiliser plus tard. Par exemple, on dira que l'identifiant est `LOGIN`. Mettons qu'on travaille sous l'identifiant `paquerette`. Lorsqu'on écrira « taper `LOGIN` en remplaçant `LOGIN` par l'identifiant de son compte », il faudra taper en réalité `paquerette`. Si l'on tape `LOGIN`, cela ne fonctionnera pas…

11.3 Terminal ? Terminal administrateur ?

Dans le menu *Applications* → *Accessoires* se trouvent deux entrées permettant d'obtenir un terminal : *Terminal* et *Terminal administrateur*.

La première permet d'obtenir un terminal fonctionnant avec les droits d'accès de la session en cours. On ne pourra donc pas l'utiliser pour effectuer des opérations *privilégiées* comme créer une partition chiffrée. Le symbole à la fin de l'« invite de commande » sera un dollar (`$`).

La seconde commande permet d'obtenir un terminal avec les droits d'admnistration. On appelle également cela un *shell root*. À partir de ce dernier, les commandes pourront accéder à l'intégralité du système, sans restriction… avec les risques que cela comporte, donc. Le symbole à la fin de l'« invite de commande » sera un dièse (`#`).

11.4 Encore une mise en garde

Plus encore que pour les recettes dont on parlait plus haut, les commandes doivent être tapées très précisément. Oublier un espace, omettre une option, se tromper de symbole, être imprécis dans un argument, c'est changer le sens de la commande.

Et comme l'ordinateur effectue *exactement* ce qui est demandé, si on change la commande, il fera *exactement autre chose*…

11.5 Un exercice

On va créer un fichier vide nommé « essai », qu'on va ensuite supprimer (sans recouvrir son contenu).

Dans un terminal, entrer la commande :

```
touch essai
```

Et taper sur *Entrée* pour que l'ordinateur l'exécute.

La *commande* `touch` donne l'ordre de créer un fichier vide ; l'*argument* `essai` donne le nom de ce fichier. Aucune option n'est utilisée.

On peut alors vérifier que ce fichier a été créé en lançant la commande `ls` (qui signifie « lister ») :

```
ls
```

Une fois la commande lancée, l'ordinateur répond avec une liste. Sur celui utilisé pour les tests, cela donne :

```
Desktop
essai
```

`Desktop` est le nom d'un dossier qui existait déjà avant, et `essai` le nom du fichier qu'on vient de créer. Un autre ordinateur auraient pu répondre avec de nombreux autres fichiers en plus de `Desktop` et de `essai`.

Ce que répond la commande `ls` n'est qu'une autre manière de voir ce que l'on peut obtenir par ailleurs. En cliquant, sur le bureau, sur l'icône du *Dossier personnel*, on pourra noter dans le navigateur de fichiers l'apparition d'une nouvelle icône représentant le fichier `essai` que l'on vient juste de créer…

On va maintenant supprimer ce fichier. La ligne de commande pour le faire a pour syntaxe générale :

```
rm [options] NOM_DU_FICHIER_A_SUPPRIMER
```

On va utiliser l'option `-v` qui, dans le cadre de *cette* commande, demande à l'ordinateur d'être « bavard » (on parle de « mode verbeux ») sur les actions qu'il va effectuer.

Pour insérer le nom du fichier à supprimer, on va utiliser l'astuce donnée précédemment pour indiquer le chemin du ficher. On va donc :

- taper `rm -v` dans notre terminal,
- taper un espace afin de séparer le fichier l'option `-v` de la suite,
- dans la fenêtre du *Dossier personnel*, on va prendre avec la souris l'icône du fichier `essai` et la déposer dans le terminal.

À la fin de cette opération, on doit obtenir quelque chose comme :

```
rm -v '/home/LOGIN/essai'
```

On peut alors appuyer sur la touche *Entrée* et constater que l'ordinateur répond :

```
détruit 'essai'
```

Cela indique qu'il a bien supprimé le fichier demandé. On peut encore vérifier son absence en lançant un nouveau `ls` :

```
ls
```

On doit constater l'absence de `essai` dans la liste que nous répond la commande. Sur le même ordinateur que tout à l'heure, cela donne :

```
Desktop
```

Et l'icône doit également avoir disparu dans le navigateur de fichiers. Apparemment, il a été supprimé… même si, comme expliqué dans la première partie, son contenu existe encore sur le disque. Comme c'était un fichier vide nommé « essai », on peut se dire que ce n'est pas bien grave.

page 31

11.6 Pour aller plus loin

Cette première expérience avec cette fenêtre pleine de petits caractères pourrait être le début d'une longue passion. Pour l'entretenir, rien de mieux que de prendre le temps de lire le chapitre « Débuter en console [2] » de la *formation Debian* ou celui baptisé « Linux en mode texte : consolez-vous ! [3] » du livre *Linux aux petits oignons*.

2. http://formation-debian.via.ecp.fr/debuter-console.html
3. http://www.microlinux.fr/linux_aux_petits_oignons/chapitre_4/chapo.html

Chapitre 12

Choisir une phrase de passe

Une « phrase de passe » (ou *passphrase* en anglais) est un secret qui sert à protéger des données chiffrées. C'est ce qu'on utilise pour chiffrer un disque dur, des documents… voire, comme nous le verrons dans le second tome de cet ouvrage, des clés cryptographiques.

On parle de *phrase* plutôt que de *mot* de passe car un seul *mot*, aussi bizarre et compliqué soit-il, est beaucoup moins résistant qu'une simple phrase de plusieurs mots. On considère qu'une phrase de passe doit être constituée d'au moins 10 mots. Mais plus il y en a, mieux c'est !

Un critère important, mais parfois négligé : une bonne phrase de passe est une phrase de passe dont on peut *se souvenir*. Mais, et c'est tout aussi important, une bonne phrase de passe doit être *impossible à deviner*.

Une technique simple pour trouver une bonne phrase de passe, difficile à deviner, mais néanmoins facile à retenir, est d'utiliser des paroles de chansons :

1. Choisissons un air qui nous trotte dans la tête, mais que nous ne chantons pas souvent à haute voix.
2. Trouvons dans les paroles, en évitant le refrain, un vers que nous aimons bien.
3. Prenons ce vers et transformons-le quelque peu. Par exemple, nous pouvons mettre de la ponctuation, remplacer des mots par de l'écriture SMS, *etc.*

Chaque fois que nous aurons besoin de taper cette phrase de passe, chantons-nous à nous-même la chanson (mentalement !), et le tour est joué.

Un conseil toutefois : il est préférable d'éviter les caractères accentués ou tout autre symbole n'étant pas directement disponible sur un clavier américain. Cela peut éviter des problèmes de touches manquantes, et surtout de mauvais codage des caractères.

Un exemple, avec un air particulièrement difficile à s'enlever de la tête, *Can't Get You Out of my Head* de Kylie Minogue. On choisit ensuite les vers :

> There's a dark secret in me
> Don't leave me locked in your heart

On peut les transformer ainsi, pour obtenir une phrase de passe :

```
There is a DARK secret in me: do not leave me locked in Ur heart!
```

Une fois vos données chiffrées avec votre nouvelle phrase de passe, c'est une bonne idée de l'utiliser tout de suite une grosse dizaine de fois pour déchiffrer vos données. Cela permettra d'apprendre un peu à vos doigts comment la taper.

Chapitre 13

Démarrer sur un CD ou une clé USB

On va voir ici comment démarrer un ordinateur PC sur un média externe, par exemple un CD d'installation de Debian, ou un système *live* sur une clé USB.

Parfois, en particulier sur les ordinateurs modernes, c'est très simple. D'autres fois, c'est un peu à s'arracher les cheveux…

Cela se joue au tout début du démarrage de l'ordinateur, dans le BIOS. On a vu que page 13 c'est lui qui permet de choisir le périphérique (disque dur, clé USB, CD-ROM, *etc.*) où se trouve le système qu'on veut utiliser.

13.1 Essayer naïvement

Commencer par mettre le CD dans le lecteur, ou par brancher la clé, puis (re)démarrer l'ordinateur. Parfois, ça marche tout seul. Si c'est le cas, c'est gagné, lire la suite est inutile !

13.2 Tenter de choisir le périphérique de démarrage

Sur les BIOS récents, il est souvent possible de choisir un périphérique de démarrage au cas par cas.

(Re)démarrer l'ordinateur en regardant attentivement les tous premiers messages qui s'affichent à l'écran. Chercher des messages en anglais qui ressembleraient à :

- `Press [KEY] to select temporary boot device`
- `[KEY] = Boot menu`
- `[KEY] to enter MultiBoot Selection Menu`

Ces messages disent d'utiliser la touche `KEY` pour choisir un périphérique de démarrage. Cette touche est souvent F12 ou F10 .

Sur les Mac, il existe un équivalent de cette possibilité : immédiatement après l'allumage de l'ordinateur, il faut appuyer et maintenir la touche alt (parfois également marquée option). Au bout d'un moment, on doit normalement voir apparaître le *Gestionnaire de démarrage*[1].

Mais revenons à nos PC. Souvent, le BIOS va trop vite, on n'a pas le temps de lire le message, de le comprendre et d'appuyer sur la touche. Qu'à cela ne tienne, une fois la bonne touche identifiée, redémarrer encore la machine et appuyer sur la touche

1. http://support.apple.com/kb/HT1310?viewlocale=fr_FR

en question (ne pas maintenir la touche enfoncée, mais la presser puis la relâcher plusieurs fois) dès l'allumage de l'ordinateur.

Avec un peu de chance, un message comme celui-ci s'affiche :

```
+---------------------------------+
| Boot Menu                       |
+---------------------------------+
|                                 |
| 1: USB HDD                      |
| 4: IDE HDD0: BDS GH87766319819  |
| 8: Legacy Floppy Drives         |
|                                 |
|    <Enter Setup>                |
|                                 |
+---------------------------------+
```

Si ça marche, c'est gagné. Choisir la bonne entrée dans ce menu, en se déplacant avec les flèches du clavier [↑] et [↓], puis appuyer sur *Entrée*. Par exemple, pour démarrer sur une clé USB, choisir `USB HDD`. L'ordinateur doit démarrer sur le périphérique sélectionné. Lire la suite est inutile !

13.3 Modifier les paramètres du BIOS

Si choisir un périphérique de démarrage temporaire ne fonctionne pas, il va falloir rentrer dans le BIOS pour choisir manuellement l'ordre de démarrage. Pour pimenter un peu la chose, les BIOS sont quasiment tous différents, de telle sorte qu'il est impossible de donner une recette qui marche systématiquement [2].

Entrer dans le BIOS

Encore une fois, il s'agit de (re)démarrer l'ordinateur en regardant attentivement les premiers messages qui s'affichent à l'écran. Chercher des messages en anglais qui ressembleraient à :

- `Press [KEY] to enter setup`
- `Setup: [KEY]`
- `[KEY] = Setup`
- `Enter BIOS by pressing [KEY]`
- `Press [KEY] to enter BIOS setup`
- `Press [KEY] to access BIOS`
- `Press [KEY] to access system configuration`
- `For setup hit [KEY]`

Ces messages disent d'utiliser la touche `[KEY]` pour entrer dans le BIOS. Cette touche est souvent [Suppr] (*Delete*, *DEL*) ou [F2], parfois [F1], [F10], [F12], [Échap], [⇆] (*Tab*), voire autre chose encore.

Voici un tableau qui résume les touches d'accès au BIOS pour quelques fabriquants d'ordinateurs communs [3].

2. Des protocoles illustrés pour quelques BIOS sont disponibles sur http://www.hiren.info/pages/bios-boot-cdrom

3. Sources : http://pcsupport.about.com/od/fixtheproblem/a/biosaccess_pc.htm, ainsi que http://michaelstevenstech.com/bios_manufacturer.htm

Fabriquant	Modèle	Touches observées
Acer	modèles récents	F2 , Suppr
Acer	modèles anciens	Ctrl + Alt + Échap , F1
AST, ARI		Ctrl + Alt + Échap , Ctrl + Alt + Suppr
Compaq	modèles récents	F10
Compaq	modèles anciens	F1 , F2 , Suppr
CompUSA		Suppr
Cybermax		Échap
Dell	modèles récents	F2
Dell	anciens *desktops*	Ctrl + Alt + Entrée , Suppr
Dell	anciens portables	Fn + Échap , Fn + F1
eMachines		↹ (*Tab*), Suppr , F2
Fujitsu		F2
Gateway		F1 , F2
HP		F1 , F2 , Échap
HP	tablet PC	F10 , F12
IBM	modèles récents	F1
IBM	anciens modèles	F2
IBM/Lenovo	modèles récents	F1 , F2
IBM/Lenovo	anciens modèles	Ctrl + Alt + F3 , Ctrl + Alt + Ins , Fn + F1
Intel	Tangent	Suppr
Micron		F1 , F2 , Suppr
NEC		F2
Packard Bell		F1 , F2 , Suppr
Shuttle		F1 , Suppr
Sony		F1 , F2 , F3
Tiger		Suppr
Toshiba		F1 , Échap
Toshiba	Equium	F12

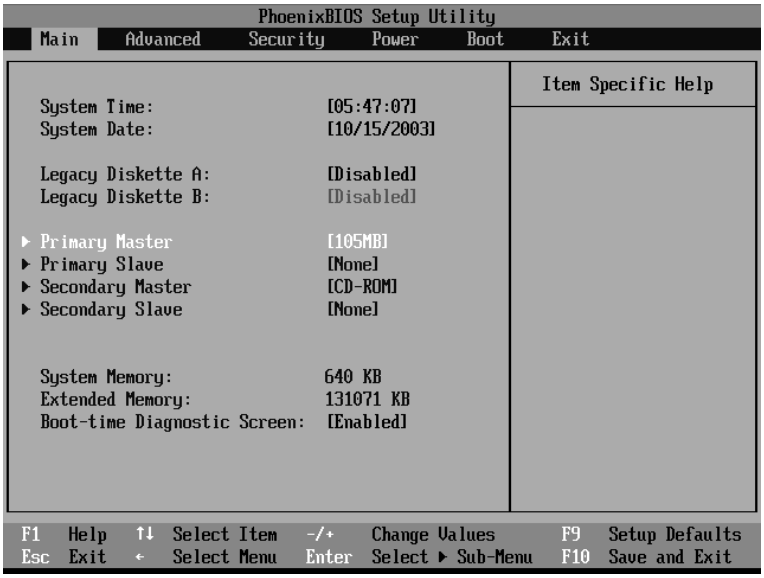
Souvent, le BIOS va trop vite, et on n'a pas le temps de lire le message, de le comprendre et d'appuyer sur la touche. Qu'à cela ne tienne, une fois la bonne touche identifiée, redémarrer encore la machine en appuyant sur la touche en question (ne pas maintenir la touche enfoncée, mais la presser puis la relâcher plusieurs fois). Parfois, l'ordinateur se perd et plante. Dans ce cas, redémarrer et réessayer…

Si une image s'affiche à la place du message que l'on espère voir, il se peut que le BIOS soit configuré pour afficher un logo plutôt que ses messages. Essayer d'appuyer sur Échap ou sur ↹ (*Tab*) pour voir les messages.

Si l'ordinateur démarre trop rapidement pour qu'on ait le temps de lire les messages qu'il affiche, il est parfois possible d'appuyer sur la touche *Pause* (souvent en haut à droite du clavier) pour geler l'écran. Réappuyer sur n'importe quelle touche peut « dégeler » l'écran.

Modifier la séquence de démarrage

Une fois dans le BIOS, l'écran est souvent bleu ou noir, et plein de menus. En général, une zone en bas ou à droite de l'écran explique comment naviguer entre les options, comment changer d'onglet… (aide se dit « *help* », touche se dit « *key* », sélectionner se dit « *select* », valeur « *value* » et modifier « *modify* »). Les touches à utiliser pour se déplacer sont généralement décrites aussi, par exemple `←↑↓→: Move` (en anglais, déplacer se dit « *move* »). Il s'agit des flèches du clavier ↓ et ↑ et/ou ← et → . Parfois, la touche ↹ (*Tab*) est utile aussi.



Un écran de BIOS

L'idée, c'est de fouiller dedans jusqu'à trouver quelque chose qui contient `boot`, et qui ressemble par exemple à :

- `First Boot Device`
- `Boot Order`
- `Boot Management`
- `Boot Sequence`

S'il n'y a pas, tenter quelque chose comme `Advanced BIOS Features` (sur les *Award-BIOS*) ou `Advanced features` (sur les *AMIBIOS*).

Une fois la bonne entrée trouvée, il s'agit de trouver comment on la modifie. Par exemple `Enter: Select` ou `+/-: Value`. L'objectif est alors de mettre le CD ou l'USB en premier, selon sur lequel on veut démarrer.

Parfois, il faut entrer dans un sous-menu. Par exemple s'il y a un menu `Boot order` et qu'il est écrit dans l'aide `Enter: Select`, appuyer sur entrée une fois le menu sélectionné.

D'autres fois, les options se changent directement. Par exemple, s'il y a une option comme `First boot device` et qu'il est écrit dans l'aide `+/-: Value`, appuyer sur la touche [+] ou la touche [−] jusqu'à ce que la bonne valeur, comme par exemple `IDE CDROM`, soit sélectionnée. Parfois, ce sont plutôt les touches *Page suivante* ou [PgUp] et *Page précédente* ou [PgDown] qui sont utilisées. Parfois encore, ce sont des touches comme [F5] et [F6]. D'autres fois encore, ces touches servent à monter et à descendre le périphérique dans une liste correspondant à l'ordre de démarrage.

Bien choisir sa nouvelle configuration

Une fois qu'on a réussi à sélectionner le bon support pour le démarrage, il faut se demander si on veut le laisser pour toujours ou pas. Si on veut le laisser, il peut être utile de placer le disque dur en deuxième position dans la séquence de démarrage. Ainsi, si le support placé en premier est absent, l'ordinateur démarrera sur le disque dur. Si l'on ne met pas le disque dur dans la séquence de démarrage, l'ordinateur ne démarrera pas dessus, même en l'abscence de CD ou de clé USB.

Cependant, le fait de laisser son ordinateur démarrer *a priori* sur un support externe peut avoir des conséquences fâcheuses : il devient un peu plus facile pour un intrus de le faire démarrer en utilisant ce support, par exemple pour effectuer une attaque.

On peut certes mettre en place, avec le BIOS, un mot de passe d'accès à l'ordinateur, qui devra être entré avant tout démarrage. Mais il est inutile de compter sur celui-ci pour protéger quoi que ce soit : cette protection peut, la plupart du temps, être contournée très facilement.

Enregistrer et quitter

Une fois la nouvelle configuration établie, il reste à enregistrer et à quitter. Encore une fois, lire l'aide à l'écran, comme `F10: Save`. Parfois, il faut appuyer une ou plusieurs fois sur [Échap] pour avoir le bon menu. Un message s'affiche alors pour demander (en anglais) si on est sûr de vouloir enregistrer et quitter. Par exemple :

```
+-------------------------------------+
|         Setup Confirmation          |
+-------------------------------------+
|                                     |
| Save configuration and exit now     |
|                                     |
|     <Yes>              <No>         |
|                                     |
+-------------------------------------+
```

On veut effectivement enregistrer, donc on sélectionne `Yes` et on appuye sur *Entrée*.

Chapitre 14

Utiliser un système *live*

14.1 Qu'est-ce qu'un système *live* ?

Un système *live* est un système GNU/Linux qui fonctionne sans être installé sur le disque dur de l'ordinateur.

Attention, cela ne signifie pas qu'il n'y aura pas de traces sur le disque dur : par exemple, nombre de systèmes *live* utilisent l'espace d'échange (*swap*) présent sur le disque dur s'ils en détectent un. De plus, ils utilisent parfois automatiquement les partitions qu'ils y détectent.

page 17

14.2 Des systèmes *live* discrets

Par contre, certains systèmes *live* sont spécialement conçus pour (tenter de) ne laisser aucune trace sur le disque dur de l'ordinateur sur lequel ils sont utilisés, à moins que l'on ne leur demande expressément de le faire. C'est par exemple le cas de *The (Amnesic) Incognito Live System* — le système *live* (amnésique) Incognito.

Il n'y a alors (si les personnes à l'origine du système *live* ne se sont pas trompées) rien d'écrit sur le disque dur. Tout ce qui sera fait à partir du système *live* sera uniquement écrit en mémoire vive, qui s'efface plus ou moins pour de vrai toute seule quand on éteint l'ordinateur, au moins au bout d'un certain temps.

page 11

Utiliser de tels systèmes *live* est donc l'une des meilleures façons d'utiliser un ordinateur sans laisser de traces. Nous verrons ici comment obtenir un système *live*, et comment démarrer dessus.

Le moyen usuel d'utiliser un système *live* est de le graver sur un CD. On parle alors de *Live CD*.

Cependant, il est aussi possible d'utiliser un système *live* qui ne touche pas à l'ordinateur à partir d'une clé USB. Néanmoins, vu qu'il est possible d'écrire des données sur une clé USB alors que ce n'est pas possible sur un CD, on a moins de garanties au cas où les gens qui ont écrit le système *live* auraient commis des erreurs. Cela devient aussi plus simple pour des personnes malveillantes de modifier votre système *live* pour, par exemple, enregistrer vos mots de passe ou vos frappes clavier.

14.3 Télécharger un système *live*

Les images de *The (Amnesic) Incognito Live System* sont distribuées par BitTorrent. C'est un protocole de partage de fichiers de pair-à-pair (« *peer-to-peer* » en anglais)

qui nécessite d'utiliser un logiciel de téléchargement spécial. Il faut tout d'abord télécharger un petit fichier, appelé *torrent*. Ce fichier contient les informations qui seront nécessaires au logiciel de téléchargement pour trouver les sources du fichier que l'on veut obtenir.

Télécharger le *torrent*

Tout d'abord, il faut télécharger un fichier `.torrent`. Pour cela, il est nécessaire de pointer un navigateur vers : https://amnesia.boum.org/torrents/files/

On peut constater qu'il y a sur cette page un certain nombre de fichiers, correspondant à plusieurs versions de ce système *live*. Le nom du fichier contient sa description :

- l'architecture pour laquelle il fonctionne, par exemple `i386` ou `powerpc` ;
- sa version, ainsi que sa date de publication, par exemple `0.4.2-20100207`.

Il y a en outre plusieurs extensions pour les mêmes noms de fichiers :

- les fichiers `.torrent` correspondent au torrent lui-même. C'est lui qui permet de télécharger le système *live*, une fois ouvert avec un client BitTorrent ;
- les fichiers `.asc` contiennent la signature cryptographique du `.torrent` ;
- les fichiers `.package` contiennent la liste des paquets Debian installés dans le système *live* correspondant.

On va choisir la version la plus récente, et télécharger le fichier `.torrent` pour notre architecture — choisir `i386` en cas de doute.

Vérifier l'authenticité du système *live*

Le système *live* que l'on vient de télécharger est signé avec GnuPG, qui utilise du chiffrement asymétrique. Malheureusement, cette technique est au-delà des notions abordées dans ce tome. Il faudra se référer aux sections correspondantes du tome *en ligne*.

Télécharger le système *live*

Sur une Debian standard, il suffit de double-cliquer sur le fichier `.torrent` téléchargé, et le logiciel de téléchargement de *torrents Transmission* s'ouvre tout seul. Il suffit de cliquer sur *Ajouter* pour démarrer le téléchargement.

À ce stade, il peut être intéressant de repérer le dossier dans lequel le logiciel de téléchargement place le fichier téléchargé : on aura besoin de le retrouver par la suite. Pour ce faire, dès que le fichier a été placé dans la fenêtre des téléchargements, on peut faire dessus un clic droit → *Détails* → onglet *Informations*. La ligne *Dossier de destination* indique l'endroit où sera placé le fichier.

Si le client BitTorrent ne s'ouvre pas tout seul, on va l'ouvrir à la main :

page 119

- sous Debian ou Ubuntu, dans le menu *Applications* → *Internet* ouvrir le *Client BitTorrent Transmission*. S'il ne s'y trouve pas, il faut, au préalable, installer le paquet `transmission-gtk` ;
- sous Mac OS X, il est aussi possible d'installer Transmission [1] ;
- sous Windows, il est possible d'installer le client libre Vuze [2].

1. http://www.transmissionbt.com/
2. http://www.vuze.com/

14.4 Installer le système *live* sur le support choisi

Selon qu'on a téléchargé une version pour CD ou pour clé USB, la manière de l'installer sur le support en question diffère.

Avec une image CD : graver l'image

Le fichier téléchargé est une « image ISO », c'est-à-dire un format de fichier que la plupart des logiciels de gravure reconnaissent comme « image CD brute ». En général, si on a inséré un disque vierge dans son lecteur, qu'on fait un clic droit sur le fichier téléchargé et qu'on choisit *Graver un disque*, le logiciel de gravure s'occupe tout seul de transformer cette image en l'écrivant sur un CD vierge ou réinscriptible.

Sous Windows, si on ne dispose pas déjà d'un logiciel capable de graver des images ISO, le logiciel libre InfraRecorder [3] fera parfaitement l'affaire.

Avec une image pour clé USB : copier l'image bit à bit

L'image ISO téléchargée est un peu spéciale, et peut aussi être utilisée pour démarrer le système *live* à partir d'une clé USB. Tout le problème maintenant va être d'effectuer une copie *bit à bit* de cette image sur la clé USB — ce qui n'est pas tout à fait la même chose qu'une copie classique type copier/coller.

Repérer l'emplacement de la clé USB

Se munir d'une clé USB vierge, ou contenant uniquement des données auxquels on ne tient pas [4].

La démarche pour repérer le chemin que le système attribue à cette clé lorsqu'on la branche sur un port USB est la même que celle expliquée plus loin. page 144

Attention : pour cette recette, une fois le nom du périphérique repéré, on ne prendra pas en compte le chiffre final qui le désigne. C'est-à-dire que si on a trouvé comme nom `sdx1`, on ne va noter que `sdx`. On va appeler cet emplacement `EMPLACEMENT_CIBLE`.

Lancer la copie bit à bit

On va maintenant ouvrir un *Terminal administrateur*, tout en gardant à portée de souris l'icône de l'image ISO téléchargée auparavant. page 87

On va commencer la commande en tapant (**sans** faire *Entrée*) :

```
dd oflag=sync if=
```

Afin d'indiquer la *source* de la copie, on va maintenant, avec la souris, attraper l'icône du fichier ISO et l'amener dans le terminal. Après avoir relâché le bouton, ce qui est affiché doit ressembler à :

```
dd oflag=sync if='/home/lea/Desktop/amnesia-i386-gnome-fr-0.4.2-20100207.iso'
```

Ce n'est toujours pas fini, car il faut maintenant indiquer la *destination* de la copie, en ajoutant à la fin de notre commande :

3. http://infrarecorder.org/
4. Les données présentes au début de la clé seront perdues. Par contre, sur le reste de la clé, il serait facile de procéder à une analyse pour retrouver les fichiers dont le contenu n'aurait pas été écrasé auparavant…

```
of=/dev/EMPLACEMENT_CIBLE
```

Une fois cela fait, la commande complète doit ressembler à quelque chose comme :

```
dd oflag=sync if='/home/lea/Desktop/amnesia-i386-gnome-fr-0.4.2-20100207.iso'
↪    of=/dev/sdx
```

La copie se lance dès qu'on a appuyé sur *Entrée*, ne laissant plus apparaître qu'un sobre carré à la ligne suivante.

Dès que l'invite de commande (qui se finit par un `#`) réapparaît, la copie est terminée.

14.5 Démarrer sur un système *live*

Dès que la copie/gravure est terminée, on peut redémarrer l'ordinateur en laissant le support du système *live* dedans, et vérifier que la copie a fonctionné… à condition bien sûr qu'on ait configuré le BIOS pour qu'il démarre sur le bon support : voir la recette expliquant comment démarrer sur un média externe pour les détails.

page 95

Chapitre 15

Installer un système chiffré

15.1 L'idée

On a vu que tout ordinateur — hormis avec certains systèmes *live* — laisse un peu partout des traces des fichiers ouverts, des travaux effectués, des connexions Internet, *etc.* On a vu aussi qu'une façon d'exposer un peu moins les données conservées sur l'ordinateur ainsi que les traces qu'on y laisse est de chiffrer le système sur lequel on travaille dans son ensemble.

page 19
page 37

Il est possible d'installer un système d'exploitation GNU/Linux comme Debian [1], sur une partie chiffrée du disque dur. À chaque démarrage, l'ordinateur va demander une phrase de passe, après quoi il débloque le chiffrement du disque, ce qui donne accès aux données, et permet donc le démarrage du système. Sans cette phrase, toute personne qui voudrait consulter le contenu de ce disque se trouvera face à des données indéchiffrables. C'est ce qu'on se propose de faire dans cette recette.

page 15

15.2 Limites

⚠ **Attention !** Cette simple installation chiffrée ne règle pas tous les problèmes de confidentialité d'un coup de baguette magique. Elle ne protège les données qu'à certaines conditions.

Limites d'un système chiffré

Nous recommandons chaudement les lectures préalables suivantes :

- le chapitre concernant le chiffrement (et ses limites),
- le cas d'usage un nouveau départ, qui étudie, en détails, les limites pratiques d'un tel système et les attaques possibles contre lui.

page 37
page 57

Sans ça, l'installation d'un système chiffré peut procurer un sentiment erroné de sécurité, source de bien des problèmes.

Limites d'une nouvelle installation

Lors de l'installation d'un nouveau système, on part de zéro. Il n'y a aucun moyen simple de vérifier que le CD d'installation qu'on utilise est fiable, et ne contient pas

1. Pour chiffrer le disque dur lors de l'installation d'Ubuntu, il est nécessaire d'utiliser le CD nommé *alternate installer* [http://www.ubuntu.com/getubuntu/downloadmirrors#alternate].

par exemple de logiciels malveillants. On ne pourra éventuellement s'en rendre compte que *par la suite* — et peut-être qu'il sera trop tard…

15.3 Télécharger un CD d'installation

Pour réaliser l'installation du système, le plus simple est d'utiliser un CD ou un DVD. Toutefois, Debian en propose plusieurs, et il est donc nécessaire de commencer par choisir celui qui convient le mieux à notre situation.

Le CD d'installation par le réseau

Le plus rapide est d'utiliser un CD d'installation par le réseau. Le CD contient un système très basique mais assez léger qui fonctionne sur la plupart des ordinateurs personnels. Il télécharge les logiciels à installer par Internet. Il faut donc que l'ordinateur sur lequel on souhaite installer Debian soit connecté à Internet, de préférence par un câble réseau (et non par le *Wi-Fi* qui ne fonctionnera que rarement).

Ces images de CD d'installation de Debian se trouvent sur le site du projet [2]. Télécharger l'image dont le nom termine par `amd64-i386-powerpc-netinst.iso`, elle fonctionnera sur tous les ordinateurs domestiques.

Le CD ou le DVD complet

S'il n'est pas possible de connecter à Internet l'ordinateur sur lequel on souhaite installer Debian, il est possible de télécharger un CD ou un DVD complet.

Si l'ordinateur que l'on souhaite installer est pourvu d'un lecteur de DVD et que l'on a également accès à un graveur de DVD, le mieux est d'utiliser le DVD `multi-arch` [3]. Ce dernier fonctionnera sur tous les PCs et les Macs récents.

Lorsqu'on est contraint au CD, il faut choisir l'une des images parmis les multiples possibilités [4]. Ce que nous cherchons, ce sont les images officielles des cédéroms de la distribution « stable ».

Sur la pages proposant les CDs d'installations, il y a plusieurs liens, correspondant à différents types de processeurs. On parle d'« architectures », comme `amd64`, `i386`, `powerpc` :

- Pour un « vieux » Mac (et non un « Mac Intel »), c'est `powerpc`.
- La plupart des ordinateurs récents (PC et Mac) à l'exception des *netbooks* fonctionnent plus rapidement avec `amd64`. C'est le cas des ordinateurs contenant des processeurs *Athlon64*, *Athlon X2*, *Core 2 Duo*, *Core 2 Quad*.
- Tous les PC (par opposition aux Macs) fonctionnent avec `i386`. C'est le cas des ordinateurs contenant des processeurs *Pentium III*, *Pentium 4*, *Pentium M*, *Athlon*, *Celeron*.

Souvent, le nom du processeur est écrit sur un autocollant collé quelque part sur l'ordinateur. Dans le doute, le mieux est de choisir `i386` qui fonctionnera dans la plupart des cas.

Après avoir choisi son architecture, il suffit de télécharger le premier CD de la liste (par exemple `debian-504-amd64-CD-1.iso` ou `debian-504-i386-CD-1.iso`).

2. Pour les images multi-architectures d'installation par le réseau : http://cdimage.debian.org/debian-cd/current/multi-arch/iso-cd/
3. L'image du DVD *multi-architecture* se télécharge sur http://cdimage.debian.org/debian-cd/current/multi-arch/iso-dvd/
4. La liste de toutes les images se trouve sur http://www.debian.org/CD/http-ftp/

À noter toutefois : l'espace disponible sur un CD est trop petit pour contenir toutes les traductions des logiciels qui seront installés. Si l'on réalise donc une installation avec un CD et sans connexion à Internet, certains logiciels seront en anglais.

Pour la suite de notre installation, on parlera de CD, mais cela s'applique tout autant si l'on a choisi d'utiliser un DVD.

15.4 Vérifier l'empreinte du CD d'installation

Il est bon de s'assurer que le téléchargement de l'image s'est bien déroulé en vérifiant l'empreinte de l'installeur. Cela ne permet malheureusement pas vraiment de s'assurer de l'authenticité de l'installeur téléchargé, car l'empreinte est signée avec GnuPG, qui utilise du chiffrement asymétrique. Malheureusement, cette technique est au-delà des notions abordées dans ce tome. Il faudra se référer aux sections correspondantes du tome *en ligne*. De plus, le logiciel utilisé pour vérifier l'empreinte n'a pas lui-même été vérifié, et il peut être corrompu.

Évoquons tout de même rapidement le processus à suivre, même si il faudra attendre le tome *en ligne* pour les recettes correspondantes :

- à l'adresse où l'on a téléchargé le CD, télécharger les fichiers `SHA1SUMS` et `SHA1SUMS.sign` ;
- démarrer sur un système déjà installé. Si l'on a accès à un ordinateur sous GNU/Linux, tout va bien. Si on ne dispose que d'un système *live*, il est possible de mettre l'image téléchargée sur une clé USB, puis de vérifier l'empreinte à partir du système *live* ; page 101
- vérifier la signature GnuPG de l'empreinte, disponible dans le fichier `SHA1SUMS.sign` ;
- enfin, vérifier que l'empreinte du fichier téléchargé est bien celle attendue. page 163

15.5 Graver le CD d'installation

Le fichier téléchargé est une « image ISO », c'est-à-dire un format de fichier que la plupart des logiciels de gravure reconnaissent comme « image CD brute ». En général, si on a inséré un disque vierge dans son lecteur, qu'on fait un clic droit sur le fichier et qu'on choisit « graver un CD », le logiciel de gravure s'occupe tout seul de transformer cette image en l'écrivant sur le CD — en tout cas, ça marche avec *The (Amnesic) Incognito Live System*, et plus généralement sous Debian ou Ubuntu.

Sous Windows, si on a pas déjà installé de logiciel capable de graver des images ISO, le logiciel libre InfraRecorder [5] fera parfaitement l'affaire.

15.6 L'installation proprement dite

Pour installer la Debian chiffrée depuis le CD d'installation, il faut démarrer sur celui-ci en suivant la recette correspondante. page 95

À partir de là, l'installation proprement dite peut commencer : prévoir une demi-journée — avec du temps libre, car l'ordinateur pourra travailler longtemps sans surveillance particulière.

Vérifier, dans le cas d'un CD d'installation par le réseau, d'avoir bien branché le câble reliant l'ordinateur au réseau.

5. http://infrarecorder.org/

Lancement de l'installeur

On démarre donc sur le CD d'installation. Un premier menu nommé *Installer boot menu* apparaît. Pour avoir de jolis menus graphiques, on peut choisir *Graphical install* en se déplacant avec les touches  et du clavier. Sinon, *Install* marche aussi. Une fois l'entrée sélectionnée, il faut appuyer sur la touche *Entrée*.

Choisir la langue et la disposition du clavier

- Un menu nommé *Choose language* apparaît alors : l'installeur propose de choisir une langue pour la suite de l'installation. Toujours en se déplaçant avec les flèches, sélectionner *Français* et appuyer sur la touche *Entrée*.
- Un menu demande le pays, pour peaufiner l'adaptation du système. Choisir son lieu géographique, et appuyer sur *Entrée*.
- Dans *choisir la disposition du clavier*, le choix par défaut *Français (`fr-latin9`)* convient si l'on a un clavier français « azerty ».
- L'installeur charge ensuite les fichiers dont il a besoin.

Configuration du réseau et baptême de la machine

- L'installeur prend alors un peu de temps pour configurer le réseau, puis demande le *Nom de machine*. Choisir un petit nom pour son ordinateur, en sachant que ce nom sera ensuite visible depuis le réseau, et pourra aussi s'inscrire dans les fichiers créés avec celui-ci.
- L'installeur demande le *Domaine*. On peut en général laisser ce champ vide.

Choix du serveur Debian

Si cette question n'apparaît pas à ce moment, pas d'inquiétudes, c'est simplement que l'installeur utilisé n'est pas celui par le réseau. Dans ce cas, elle arrivera un peu plus tard au cours de l'installation.

- L'installeur demande de *Choisir un miroir de l'archive Debian*. Le choix par défaut *France* est bon si l'on est en France.
- Il demande ensuite le *Miroir de l'archive Debian* à utiliser. Le choix par défaut *ftp.fr.debian.org* est aussi très bien.
- L'installeur demande si on a besoin d'un *Mandataire HTTP*. On laisse vide.
- L'installeur télécharge alors les fichiers dont il a besoin pour continuer.

Partitionner les disques

Le CD démarre ensuite l'outil de partitionnement. Il détecte les partitions présentes, et va proposer de les modifier.

- Dans le menu *Méthode de partitionnement*, choisir *Assisté — utiliser tout un disque avec LVM chiffré*.
- Dans *disque à partitionner* choisir le disque sur lequel installer Debian GNU/Linux. Si l'on veut supprimer votre ancien système, il est en général possible de choisir le premier disque de la liste.
- L'installeur propose ensuite différents *Schémas de partitionnement*. Là, il y a plusieurs possibilités :
  - *Tout dans une seule partition* fonctionne toujours ;
  - Si l'on a un gros disque (mettons, au moins 20 Go), on peut séparer la partition `/home` qui contiendra vos données personnelles dans une partition séparée.

- L'installeur prévient alors qu'il va appliquer le schéma actuel de partitionnement, ce qui sera irréversible. Vu que l'on a bien fait les sauvegardes de ce que l'on voulait garder, répondre *Oui* à *Écrire les modifications sur les disques et configurer LVM ?* L'installeur va alors remplacer l'ancien contenu du disque par des données aléatoires. C'est très long — de nombreuses heures sur un gros disque — il y a le temps de faire autre chose !
- L'installeur demande alors une *Phrase secrète de chiffrement*. Choisir une bonne phrase de passe. page 93
- Confirmer la phrase de passe.
- L'installeur montre une liste de toutes les partitions qu'il va créer sur les disques. Il est possible de lui faire confiance et de *Terminer le partitionnement et appliquer les changements*.
- L'installeur prévient qu'il va détruire toutes les données présentes sur le disque. Tout le disque a déjà été rempli de données aléatoires, donc s'il contenait des données importantes elles ont déjà été effacées. Répondre *Oui* à *Faut-il appliquer les changements sur les disques ?* L'installeur crée alors les partitions, ce qui peut prendre un petit bout de temps.

L'installation du système de base

L'installeur va maintenant installer un système GNU/Linux minimal. Le laisser faire…

Créer les utilisateurs et choisir les mots de passe

Ici, on se propose de tricher pour que ce soit le premier compte créé sur le système qui ait le droit d'administrer l'ordinateur[6]. Si l'on ne fait pas cela, il y aura un mot de passe supplémentaire pour administrer le système. Cela dit, ce choix doit être bien pesé au préalable : souvent, il est plus simple d'utiliser cette méthode, notamment parce qu'il n'y a pas un mot de passe supplémentaire à retenir. Cependant, dans sa configuration par défaut, elle peut permettre à n'importe quel programme lancé dans ce compte, *sans que celui-ci nous demande confirmation au préalable*, d'effectuer des opérations en disposant des privilèges d'administrateur ; et ce, pendant quinze minutes après la saisie du mot de passe.

Si l'on souhaite avoir un mot de passe administrateur à part, il suffit de sauter cette étape et de lire les questions de l'installeur, qui sont assez claires. Sinon, voici comment faire :

- Choisir *Revenir en arrière*, en utilisant la touche  (*Tab*).
- Dans le *Menu principal du programme d'installation*, sélectionner *Changer la priorité des questions de configuration*.
- Choisir d'*Ignorer les questions de priorité inférieure* à *intermédaire*.
- Retourner dans *Créer les utilisateurs et choisir les mots de passe*.
- Répondre *Non* à *Faut-il autoriser les connexions du superutilisateur ?*.
- Dans *Nom complet du nouvel utilisateur* choisir le nom associé au premier compte créé sur le système. Ce nom sera souvent enregistré dans les documents créés dans cette session, il peut donc être interessant de choisir un nouveau pseudonyme.
- Dans *Identifiant pour le compte utilisateur*, choisir un identifiant (*login*) pour ce compte. Il est prérempli, mais peut être modifié. L'installeur prévient pour le cas où l'on voudrait le changer, qu'il doit commencer par une lettre minuscule et être suivi d'un nombre quelconque de chiffres et de lettres minuscules.
- L'installeur demande un mot de passe pour l'utilisateur qui aura le droit d'administrer l'ordinateur.

6. Dans le terminal, on deviendra alors administrateur en mettant `sudo` avant une commande.

- L'installeur revient, à la fin de cette étape, au Menu principal. Pour poursuivre l'installation garder le choix par défaut, *Configurer l'outil de gestion des paquets*, en appuyant sur la touche *Entrée*.

Sélection des logiciels

La prochaine question concerne la *configuration de popularity-contest* et demande *Souhaitez-vous participer à l'étude statistique sur l'utilisation des paquets ?* Il est possible de répondre *Oui* sans risque supplémentaire de divulguer des informations : non seulement cette fonction se veut anonyme, mais vu que les logiciels seront de toute façon téléchargés à partir des serveurs de Debian, ceux-ci pourraient déjà savoir quels paquets on utilise s'ils le voulaient.

L'installeur demande les *Logiciels à installer*. Ses propositions conviennent en général : *Environnement graphique de bureau* et *Système standard*, plus *Ordinateur portable* sur un portable. Puis, pour atteindre le bouton de validation, il faut utiliser la touche  (*Tab*).

L'installeur installe alors tout le reste du système Debian GNU/Linux. C'est long, il y a le temps d'aller faire autre chose.

Installation du programme de démarrage GRUB

L'installeur propose de mettre en place le programme de démarrage, qui permet de démarrer Linux, sur une partie du disque dur appelée « secteur d'amorçage ». Répondre *Oui* et attendre que l'installeur termine l'installation.

Lorsqu'il a terminé, l'installeur demande de sortir le CD d'installation, et propose de redémarrer l'ordinateur. Éjecter le CD et choisir *Continuer*.

Redémarrer sur le nouveau système

L'ordinateur démarre alors sur le nouveau système. À un moment, il demande la phrase de passe sur un écran noir : `Enter passphrase to unlock the disk`. La taper, sans s'inquiéter que rien ne s'affiche, et appuyer sur la touche *Entrée* à la fin.

Quand un écran avec écrit *Bienvenue Identifiant : saisissez votre identifiant* s'affiche, entrer le *login*, puis dans *Mot de passe* entrer le mot de passe.

Voilà un nouveau système Debian chiffré prêt à être utilisé. Pour qui n'en aurait jamais utilisé, se balader dedans peut être une bonne idée pour s'y familiariser.

page 114
Peut-être sera-t-il utile maintenant d'installer de nouveaux logiciels. Dans les premiers paquets que l'on ajoute sur la plupart des systèmes Debian, on compte `ntfs-3g` (pour
page 16
lire les disques externes en NTFS) et, avec les dépôts *contrib* et *non-free*, les paquets `firmware-PILOTE` ou `b43-fwcutter` qui permettront de faire fonctionner la carte *Wi-Fi*.

15.7 Quelques pistes pour continuer

Des outils

page 151
Il peut maintenant être utile d'apprendre à sauvegarder des données… et à en effacer
page 127
« pour de vrai ».

Un peu de documentation

Voici quelques références de documentations sur Debian et GNU/Linux :

- Le guide de référence officiel de Debian [7] ;
- La page d'accueil de la documentation officielle d'utilisation de Debian [8] ;
- La Formation Debian GNU/Linux [9] : une excellente auto-formation sur Debian en français.

On peut trouver beaucoup de documentations sur l'utilisation de GNU/Linux. Si elles sont souvent très utiles, elles sont malheureusement, comme beaucoup de choses sur Internet du reste, de qualité inégale. En particulier, beaucoup d'entre elles arrêteront de fonctionner lorsqu'une partie du système sera modifiée, ou seront peu soucieuses de l'intimité que l'on attend de notre système. Il faut donc faire preuve d'esprit critique et tenter de les comprendre avant de les appliquer.

Ceci dit, voici encore quelques réferences de wikis et des forums :

- Le wiki officiel de Debian [10] (partiellement traduit de l'anglais) ;
- Le forum en français sur Debian `debian-fr.org` [11] ;
- Andesi [12] : un wiki et forum en français sur Debian.

7. http://www.debian.org/doc/manuals/debian-reference/index.fr.html
8. http://www.debian.org/doc/user-manuals
9. http://formation-debian.via.ecp.fr/
10. http://wiki.debian.org/
11. http://forum.debian-fr.org/
12. http://www.andesi.org/

Chapitre 16

Choisir, vérifier et installer un logiciel

Cette partie propose quelques recettes à propos de la gestion de ses logiciels :

- Comment trouver un paquet Debian ? Lorsqu'on cherche à réaliser de nouvelles tâches avec un ordinateur, on est souvent amené à installer de nouveaux logiciels… quelques conseils pour trouver ce que l'on cherche dans Debian ; page suivante
- Avec quels critères de choix ? On est parfois amené à choisir un logiciel pour effectuer une certaine tâche, et il est alors courant de se sentir perdu dans la multitude de solutions disponibles… quelques critères permettant de prendre une décision adéquate ; page 116
- Comment installer un paquet Debian ? Une fois que l'on sait quel paquet contient le logiciel que l'on veut utiliser, reste à l'installer proprement ; page 119
- Comment modifier ses dépôts Debian ? Les paquets Debian qui contiennent les programmes se trouvent dans ce qu'on appelle des *dépôts*. Si les dépôts fournis avec Debian contiennent quasiment tous les logiciels dont on peut avoir besoin, il est parfois utile d'ajouter de nouveaux dépôts ; page 121
- Comment faire de l'*APT pinning* ? Ce mécanisme permet, lorsqu'un même paquet est accessible à partir de plusieurs dépôts, de choisir quelle version le système doit installer. page 125

16.1 Trouver un logiciel

Parfois, on connaît déjà le nom du logiciel que l'on souhaite installer — parce qu'on nous l'a conseillé, parce qu'on l'a trouvé sur Internet — et l'on veut savoir s'il est dans Debian. D'autres fois, on connaît seulement la tâche que l'on souhaiterait que le logiciel remplisse. Dans tous les cas, la base de données des logiciels disponibles dans Debian répondra certainement à nos questions.

Voici quelques conseils pour y trouver ce que l'on cherche :

- *trouver une application* s'applique pour chercher un programme susceptible d'être ouvert dans le menu *Applications* ; sinon…
- *trouver un paquet Debian* peut s'appliquer dans tous les cas. Il donne davantage de choix, dans lesquels il est cependant facile de se perdre. Par exemple, c'est là que l'on trouvera le dictionnaire allemand pour *OpenOffice.org*, ou des *codecs*, pilotes *etc.*

page 116

Pour faire des choix éclairés, lorsque plusieurs logiciels permettent d'effectuer une même tâche, voir choisir un logiciel.

Trouver une application

- Ouvrir, via le menu, *Système* → *Administration*, *Ajouter/Supprimer des applications*. Puisque le gestionnaire de paquets permet de modifier les logiciels installés sur l'ordinateur, et donc de choisir à quels programmes on fait confiance, on est rassuré qu'il nous demande notre mot de passe pour s'ouvrir.
- Ensuite, il y a deux techniques pour chercher une application :
  - soit entrer des mots-clé ou le nom de l'application dans la case *Rechercher* en haut à droite, puis appuyer sur *Entrée*. Les résultats de la recherche apparaissent dans la liste en-dessous. Les descriptions des applications peu courantes sont rarement traduites en français. Avec quelques bases d'anglais, il est souvent intéressant d'essayer des mots-clé dans cette langue ;
  - soit cliquer, dans la liste de gauche, sur la catégorie correspondant à ce que l'on recherche. Les applications disponibles apparaissent alors dans la liste en haut à droite.
- Dans cette liste, cliquer sur le nom d'une application. Sa description s'affiche alors dans le cadre en bas à droite.
- Reste maintenant à installer le paquet correspondant.

page 119

Trouver n'importe quel paquet Debian

- Dans le menu *Système*, aller dans le sous-menu *Administration*, et ouvrir le *gestionnaire de paquets Synaptic*. Puisque le gestionnaire de paquets permet de modifier les logiciels installés sur l'ordinateur, et donc de choisir à quels programmes on fait confiance, on est rassuré qu'il nous demande notre mot de passe pour s'ouvrir.
- Une fois dans le gestionnaire de paquets, commençons par recharger la liste des paquets disponibles en cliquant sur l'icône *Recharger*. Le gestionnaire de paquets télécharge alors les dernières informations sur les paquets disponibles depuis un serveur Debian.
- Ensuite, il y a deux techniques pour chercher un paquet :
  - soit cliquer sur l'icône *Rechercher* dans la barre d'outils. Là, vérifier que *Description et nom* est bien sélectionné dans *Rechercher dans*. Taper alors un ensemble de mots-clé dans la case *Rechercher* (par exemple « dictionnaire allemand openoffice ») et cliquer sur *Rechercher* ;
  - soit sélectionner une catégorie dans la colonne de gauche.

- Les résultats de la recherche ou les paquets de la catégorie s'affichent alors dans la liste en haut à droite. En cliquant sur le nom d'un paquet, sa description apparaît dans le cadre en bas à droite.
- Reste maintenant à installer le paquet correspondant. page 119

16.2 Critères de choix

On est parfois amené à choisir un logiciel pour effectuer une certaine tâche, et il est alors courant de se sentir perdu dans la multitude de solutions disponibles. Voici donc quelques critères permettant de prendre une décision adéquate.

page 29

L'intérêt qu'il y a à utiliser des logiciels libres par rapport à des logiciels propriétaires ou *open source* a d'ores et déjà été expliqué. La suite du texte s'attachera donc uniquement à départager les logiciels libres disponibles.

Mode d'installation

Il est généralement préférable d'installer des logiciels fournis par sa distribution GNU/Linux (par exemple, Debian). Il y a deux principales raisons à ça.

Tout d'abord, une question pratique : la distribution fournit les outils pour installer et mettre à jour, de façon plus ou moins automatisée, un ensemble de logiciels ; elle nous alerte lorsqu'une faille de sécurité affecte l'un des logiciels que l'on utilise. Mais dès lors qu'on installe un logiciel qui n'est pas fourni par sa distribution, on est livré à soi-même : il faut penser à le mettre à jour, se tenir informé des failles de sécurité qui y sont découvertes, gérer les dépendances entre logiciels. Ça demande des efforts, du temps, des compétences.

D'autre part, une question de politique de sécurité : lorsqu'on a choisi sa distribution GNU/Linux, on a implicitement décidé d'accorder une certaine confiance à un ensemble de gens, à un processus de production. Installer un logiciel qui n'est pas fourni par sa distribution implique de prendre une décision similaire à propos d'un nouvel ensemble de gens, d'un nouveau processus de production. Une telle décision ne se prend pas à la légère : lorsqu'on décide d'installer un logiciel n'appartenant pas à sa distribution, on élargit l'ensemble des personnes et processus à qui on accorde de la confiance, et on augmente donc les risques.

Maturité

L'attrait de la nouveauté qui lave plus blanc que blanc est bien souvent un piège.

Mieux vaut, autant que possible, choisir un logiciel ayant atteint une certaine maturité : dans un logiciel activement développé et utilisé depuis au moins quelques années, il y a des chances que les plus gros problèmes aient déjà été découverts et corrigés… y compris les failles de sécurité.

Pour s'en rendre compte, il faut consulter l'historique de chacun des logiciels, sur leur site web ou dans le fichier nommé `Changelog` (ou approchant), généralement livré avec le logiciel.

Processus de production et « communauté »

page 30

L'étiquette *logiciel libre* est un critère essentiellement juridique, qui ne doit jamais suffire à nous inspirer confiance.

Certes, le fait qu'un logiciel soit placé sous une licence libre ouvre la possibilité de modes de développement inspirant confiance.

Mais les personnes développant ce logiciel peuvent fort bien, intentionnellement ou non, décourager toute coopération et travailler en vase clos. Que nous importe alors que le programme soit *juridiquement* libre, si, de fait, personne d'autre ne lira jamais son code source ?

Il convient donc d'étudier rapidement le processus de production des logiciels en lice, en s'aidant des questions suivantes, qui nous permettront de surcroît de jauger le dynamisme du processus en question :

- Qui développe ? Une personne, des personnes, toute une équipe ?
- Le nombre de personnes qui contribuent au code source va-t-il en augmentant ou en diminuant ?
- Le développement est-il actif ? Il ne s'agit pas ici de vitesse pure, mais de réactivité, de suivi à long terme, de résistance. Le développement logiciel est une course d'endurance et non un *sprint*.

Et à propos des outils de communication collective sur lesquels s'appuie le développement (listes et salons de discussion, par exemple) :

- A-t-on facilement accès aux discussions guidant le développement du logiciel ?
- Rassemblent-ils de nombreuses personnes ?
- Ces personnes participent-elle à son développement, ou ne font-elles que l'utiliser ?
- Quelle atmosphère y règne ? Calme plat, silence de mort, joyeuse cacophonie, sérieux glaçant, bras ouverts, hostilité implicite, tendre complicité, *etc.* ?
- Le volume de discussion, sur les derniers mois/années, va-t-il en diminuant ou en augmentant ? Plus que le volume brut, c'est surtout la proportion de messages obtenant des réponses qui importe : un logiciel mûr, stable et bien documenté ne sera pas forcément source de discussions, mais si plus personne n'est là pour répondre aux questions des néophytes, ça peut être mauvais signe.
- Peut-on trouver des retours d'utilisation, des suggestions d'améliorations ? Si oui, sont-elles prises en compte ?
- Les réponses sont-elles toujours données par un nombre réduit de personnes, ou existe-t-il des pratiques d'entraide plus large ?

Popularité

La popularité est un critère délicat en matière de logiciels. Le fait que la grande majorité des ordinateurs de bureau fonctionnent actuellement sous Windows n'indique en rien que Windows soit le meilleur système d'exploitation disponible.

Pour autant, si ce logiciel n'est pas utilisé par beaucoup de monde, on peut douter de sa viabilité à long terme : si l'équipe de développement venait à cesser de travailler sur ce logiciel, que deviendrait-il ? Qui reprendrait le flambeau ?

On peut donc retenir, comme règle générale, qu'il faut choisir un logiciel utilisé par un nombre suffisamment important de personnes, mais pas forcément *le* logiciel le plus utilisé.

Afin de mesurer la popularité d'un logiciel, il est possible, d'une part, d'utiliser les mêmes critères que ceux décrits ci-dessus au sujet du dynamisme de la « communauté » formée autour de lui. D'autre part, Debian publie les résultats de son concours de popularité [1], qui permet de comparer non seulement le nombre de personnes ayant installé tel ou tel logiciel, mais aussi, voire surtout, l'évolution dans le temps de leur popularité.

Passé de sécurité

Voici de nouveau un critère à double tranchant.

1. http://popcon.debian.org/

On peut commencer par jeter un œil sur le suivi de sécurité [2] proposé par Debian. En y cherchant un logiciel par son nom, on peut avoir la liste des problèmes de sécurité qui y ont été découverts et parfois résolus.

Si ce logiciel a un historique de sécurité parfaitement vierge, ça peut impliquer soit que tout le monde s'en fout, soit que le logiciel est écrit de façon extrêmement rigoureuse.

Si des failles de sécurité ont été découvertes dans le logiciel étudié, il y a plusieurs implications, parfois contradictoires.

1. Ces failles ont été découvertes et corrigées :
   - donc elles n'existent plus, *a priori* ;
   - donc quelqu'un s'est préoccupé de les trouver, et quelqu'un d'autre de les corriger : on peut supposer qu'une attention est donnée à cette question.
2. Ces failles ont existé :
   - le logiciel est peut-être écrit sans que la sécurité soit un souci particulier ;
   - d'autres failles peuvent subsister, non encore découvertes ou pire, non encore publiées.

Afin d'affiner notre intuition par rapport à ce logiciel, il peut être bon de se pencher sur le critère « temps » : par exemple, il n'est pas dramatique que quelques failles aient été découvertes au début du développement d'un logiciel, si aucune n'a été découverte depuis quelques années ; on peut alors mettre ça sur le compte des erreurs de jeunesse. Au contraire, si de nouvelles failles sont découvertes régulièrement, depuis des années, et jusqu'à très récemment, il est fort possible que le logiciel ait encore de nombreux problèmes de sécurité totalement inconnus… ou non publiés.

Pour illustrer le propos, il est possible de comparer l'historique des failles de Claws Mail [3] et celui de Thunderbird [4].

Équipe de développement

Qui a écrit, qui écrit ce logiciel ? Une fois répondu à cette question, divers indices peuvent nous aider à déterminer la confiance qui peut être accordée à l'équipe de développement. Par exemple :

- Les mêmes personnes ont aussi écrit un autre logiciel, que nous utilisons déjà intensivement ; nos impressions sur cet autre logiciel sont tout à fait pertinentes dans le cadre de cette étude.
- Des membres de l'équipe de développement ont des adresses qui finissent par `@debian.org`, et ont donc le droit de modifier les logiciels fournis par Debian GNU/Linux ; si nous utilisons cette distribution, nous accordons déjà, de fait, une certaine confiance à ces personnes.
- Des membres de l'équipe de développement ont des adresses qui finissent par `@google.com`, ce qui montre que Google les paie ; s'il n'y a aucun doute à avoir sur leurs compétences techniques, on peut se demander à quel point leur travail est téléguidé par leur employeur qui, lui, n'est digne d'aucune confiance quant à ses intentions concernant vos données personnelles.

2. L'équipe de sécurité de Debian maintient des informations pour chacun des paquets, visibles sur le *security tracker* [http://security-tracker.debian.net/tracker/].
3. http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=claws+mail
4. http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=thunderbird

16.3 Installer un paquet Debian

Ouvrir le gestionnaire de paquets

Une fois que l'on sait quel paquet contient le logiciel que l'on veut utiliser, reste à l'installer. Pour cela, on va utiliser le *Gestionnaire de paquets Synaptic* que l'on peut ouvrir à partir du menu *Système* → *Administration*.

Puisque le gestionnaire de paquets permet de modifier les logiciels installés sur l'ordinateur, et donc de choisir à quels programmes on fait confiance, on est rassuré qu'il nous demande notre mot de passe pour s'ouvrir.

Recharger la liste des paquets disponibles

Une fois dans le gestionnaire de paquets, commençons par recharger la liste des paquets disponibles en cliquant sur l'icône *Recharger*. Le gestionnaire de paquets télécharge alors les dernières informations sur les paquets disponibles depuis un serveur Debian.

Rechercher le paquet à installer

Ensuite, on va trouver le paquet qu'on veut installer. On clique sur l'icône *Rechercher* dans la barre d'outils. Là, si on connait le nom de ce paquet (par exemple grâce à la section précédente), on l'écrit dans la case *Rechercher*, et on sélectionne *Nom* dans *Rechercher dans*.

page 114

Sélectionner le paquet à installer

Vient alors la phase d'installation proprement dite du paquet trouvé précédemment. Il y a différentes façons de le faire, selon que l'on souhaite utiliser la version disponible dans les dépôts officiels de sa distribution, ou un paquet provenant d'un autre dépôt, par exemple pour avoir une version plus récente.

Pour installer la version par défaut

Normalement, le paquet désiré se trouve maintenant quelque part dans la liste de paquets. Une fois trouvée la ligne correspondante, on clique-droit dessus, et dans le menu contextuel on choisit *Sélectionner pour installation*.

Si ce paquet dépend d'autres paquets, le gestionnaire de paquets ouvre alors une fenêtre où il demande s'il doit *Prévoir d'effectuer d'autres changements ?* En général, ses propositions sont pertinentes, et on peut accepter en cliquant sur *Ajouter à la sélection*.

Pour installer une version particulière

Parfois, on souhaite installer une version particulière d'un paquet parmi celles disponibles. Par exemple, si on a ajouté des dépôts spécifiques. Au lieu de choisir *Sélectionner pour installation* dans le menu contextuel, il faut alors choisir, dans le menu *Paquet*, de *Forcer la version…* La suite ne change pas.

page 121

Attention : pour que les nouvelles versions du paquet soient installées automatiquement lors des mises à jour, il est nécessaire d'ajouter à sa configuration les bonnes règles d'*APT Pinning*.

page 125

Appliquer les modifications

Il est possible de répéter les deux dernières étapes pour installer plusieurs paquets en même temps. Une fois qu'on a préparé cette installation, il ne reste qu'à la lancer en cliquant sur *Appliquer* dans la barre d'outils. Le gestionnaire de paquets ouvre alors une fenêtre *Appliquer les modifications suivantes* où il liste tout ce qu'il va faire. Après avoir jeté un œil pour vérifier qu'on ne s'est pas trompé, on clique sur *Appliquer*.

Le gestionnaire de paquets télécharge alors les paquets depuis Internet, les vérifie, puis les installe. Il peut arriver que le gestionnaire indique que certains paquets n'ont pas pu être vérifiés : cette information n'est pas à prendre à la légère. Dans un tel cas, il vaut mieux annuler le téléchargement, cliquer sur *Recharger* dans le menu principal, et recommencer l'opération de sélection des paquets. Si l'indication apparaît de nouveau, cela peut être le fruit d'une attaque, d'une défaillance technique ou de soucis de configuration. Mais autant s'abstenir d'installer de nouveaux paquets avant d'avoir identifié la source du problème.

Enfin, si tout s'est bien passé, il s'affiche une fenêtre comme quoi *Les modifications ont été appliquées* et on peut donc cliquer sur *Fermer*. C'est alors une bonne idée de fermer le gestionnaire de paquets, pour éviter qu'il tombe entre d'autres mains.

16.4 Comment modifier ses dépôts Debian

Les paquets Debian qui contiennent les programmes se trouvent dans ce qu'on appelle des *dépôts*. Si les dépôts fournis avec Debian contiennent quasiment tous les logiciels dont on peut avoir besoin, il est parfois utile d'installer de nouveaux dépôts, par exemple *backports.org* qui contient des programmes plus récents que ceux inclus dans la distribution stable de Debian, ou *debian-multimedia.org* qui contient des *codecs* et des logiciels multimédia non-libres, ou interdits dans certains pays.

Attention : ajouter un nouveau dépôt Debian sur un ordinateur revient à décider de faire confiance aux gens qui s'en occupent. Si les dépôts de *backports.org* sont maintenus par des membres de Debian, ce n'est pas le cas pour de nombreux autres dépôts. La décision de leur faire confiance ne doit pas se prendre à la légère : si le dépôt en question contient des logiciels malveillants, il serait possible de les installer sur l'ordinateur sans même s'en rendre compte.

page 24

Autenticité du contenu des dépôts

Les dépôts Debian sont signés par des clés GnuPG. Ceci afin de s'assurer que leur contenu n'a pas été altéré par malveillance ou simple problème technique.

Cette section traite rapidement de comment chercher et vérifier une clé GnuPG. Il s'agit de chiffrement asymétrique, technique qui sera traitée plus avant dans le tome 2. On se contentera ici de donner un protocole permettant de vérifier une clé à partir d'une empreinte (ou « *fingerprint* »). On **n'**apprendra **pas** à bien utiliser le chiffrement asymétrique.

Ce protocole simplifié a des limites : en particulier, on vérifie la clé à partir d'une empreinte (une sorte de somme de contrôle). Toute la confiance que nous allons donner à la clé vient uniquement de cette empreinte. Or vérifier cette empreinte à partir de ce qui est écrit dans ce guide, cela signifie faire fortement confiance à la source à partir de laquelle on l'a obtenu. En lisant ce guide sur Internet, c'est encore pire : on fait, en plus, confiance à sa connexion Internet.

page 41

Encore une fois, tout est affaire de compromis entre utilisabilité et sécurité. Pour obtenir des empreintes en toute confiance, le mieux est de les vérifier en tête-à-tête. Malheureusement, ce n'est généralement pas possible en pratique lorsqu'il s'agit de dépôts Debian. Ce n'est cependant pas une raison pour ne rien vérifier du tout.

Dans le cadre du premier tome de ce guide, qui n'aborde pas les problématiques liées à l'utilisation de réseaux, il n'y a pas de solution vraiment satisfaisante. En attendant le tome 2, qui traitera plus avant de ces questions, le mieux qu'on ait trouvé est d'utiliser :

- des empreintes données dans ce guide, qui ont été vérifiées sur Internet à partir de nombreuses connexions différentes à plusieurs moments différents, mais impliquent de faire confiance à la source de ce guide ;
- si possible, les empreintes qui se trouvent sur d'autres ordinateurs sur lesquels les dépôts en question auraient été installés précédemment si l'on peut y avoir accès, chez des proches par exemple.

Ce protocole est vraiment loin d'être sûr. Il met cependant des bâtons dans les roues de l'éventuelle personne qui souhaiterait nous faire installer des logiciels malveillants.

Quelques empreintes vérifiées par nos soins

Trois des empreintes de dépôts parmis les plus utilisés sont reproduites ci-dessous :

Dépôt	Date	Empreinte
`backports.org`	août 2005	`2703 4F81 A2EB 2840 A438` `6C09 EA8E 8B21 16BA 136C`
`debian-multimedia.org`	octobre 1999	`1D7F C53F 80F8 52C1 88F4` `ED0B 07DC 563D 1F41 B907`
`deb.torproject.org`	septembre 2009	`A3C4 F0F9 79CA A22C DBA8` `F512 EE8C BC9E 886D DD89`

Comparer les empreintes avec celles présentes sur d'autres ordinateurs

Si on peut avoir accès à des ordinateurs sur lesquels les dépôts que l'on souhaite utiliser ont déjà été installés, on pourra recouper les empreintes données dans ce guide avec celles présentes sur ces ordinateurs.

Pour ce faire, ouvrir un *Terminal administrateur* à partir du menu *Applications* → *Accessoires* → *Terminal administrateur*.

Taper alors :

```
#> apt-key finger
```

Puis appuyer sur *Entrée*. On obtient alors une liste des clés de dépôts, chacune sous la forme suivante :

```
pub   1024D/16BA136C 2005-08-21
Empreinte de la clé = 2703 4F81 A2EB 2840 A438 6C09 EA8E 8B21 16BA 136C
uid                  Backports.org Archive Key <ftp-master@backports.org>
sub   2048g/5B82CECE 2005-08-21
```

C'est la troisième ligne de chaque entrée qui donne le nom du dépôt. Il s'agit dans cette liste de trouver le nom du dépôt qui nous nous intéresse. Dans l'exemple ci-dessus, on a :

```
uid                  Backports.org Archive Key <ftp-master@backports.org>
```

Il s'agit donc de la clé de *backports.org*. L'empreinte correspondante se trouve sur la ligne juste au dessus :

```
Empreinte de la clé = 2703 4F81 A2EB 2840 A438 6C09 EA8E 8B21 16BA 136C
```

Noter alors cette empreinte pour de futurs comparaisons.

Récupérer la clé d'un dépôt depuis Internet

Il faut tout d'abord ouvrir *Applications* → *Accessoires* → *Mots de passe et clés de chiffrement*.

- Dans le menu *Distant* choisir *Chercher des clés distantes* ;
- Dans *Chercher des clés contenant*, taper une partie du nom de la clé recherchée, par exemple « Backports.org », puis cliquer sur *Chercher* ;
- Une fenêtre *Clés distantes contenant […]* s'ouvre. Ici on a par exemple « Backports.org archive key » qui a pour identifiant `16BA136C` ;
- Il est alors possible d'importer la clé en allant dans le menu *Clé* puis *Importer*.

Afin de s'assurer que la clé que l'on vient d'obtenir est bien celle qu'on attend, il s'agit maintenant de vérifier son empreinte :

- Une fois la clé importée, aller dans l'onglet *Autres clés obtenues* de la fenêtre principale.
- Sélectionner la clé à vérifier, dans notre exemple « Backports.org archive key ».
- Cliquer dessus avec le bouton droit de la souris, et dans le menu contextuel qui apparaît, choisir *Propriétés*.
- Aller dans l'onglet *Détails*.
- Dans *Empreinte* il y a la somme de contrôle de la clé. C'est ce que l'on doit vérifier pour s'assurer qu'on a la bonne clé. Vérifier que l'empreinte correspond bien à ce l'on attend.

Si c'est bien le cas, on peut maintenant exporter la clé dans un fichier, avant de pouvoir l'ajouter au logiciel qui s'en servira pour vérifier le contenu des dépôts. Pour cela, dans le menu *Clé* choisir *Exporter la clé publique*. L'enregistrer, par exemple sur son bureau, en acceptant le nom par défaut.

Ajouter un nouveau dépôt

À partir du menu *Système* → *Administration* ,ouvrir les *Sources de mise à jour*. Puisque ce logiciel permet de choisir à quels programmes on fait confiance, on est rassuré qu'il nous demande notre mot de passe pour s'ouvrir.

Configurer l'emplacement du dépôt

Aller dans l'onglet *Third party software*, ce qui signifie « Tierces parties » (ce qui veut dire venues d'ailleurs que du projet Debian) et cliquer sur le bouton *Add* (Ajouter).

Entrer l'adresse du dépôt à ajouter dans la case *APT Line* de la boîte de dialogue qui s'ouvre. Par exemple, pour ajouter les *backports* (programmes plus récents que ceux inclus dans la distribution Debian stable) il faut entrer :

```
deb http://www.backports.org/debian lenny-backports main
```

Si l'on souhaite également installer des logiciels non-libres, on peut ajouter `contrib` et `non-free` en plus de `main`. À la place de la ligne précédente, on pourrait plutôt entrer :

```
deb http://www.backports.org/debian lenny-backports main contrib non-free
```

Une fois que c'est fait, il suffit de cliquer sur *Add source* (ajouter la source). Lorsqu'un nouveau dépôt a été ajouté, il est nécessaire de faire de l'*APT Pinning* afin de s'assurer que les logiciels qu'on télécharge depuis celui-ci seront maintenus à jour. page 125

On doit disposer d'un fichier qui contient la clé avec laquelle sont signées les listes de paquets du dépôt à ajouter, et avoir vérifié cette clé. La télécharger sur un site web et lui faire confiance aveuglément n'est pas une bonne idée.

Ajouter une nouvelle clé de confiance

- Aller dans l'onglet *Authentification* et cliquer sur le bouton *Import key file* (Importer un fichier de clé) ;
- Sélectionnez le fichier où on a précédemment sauvegardé la clé téléchargée — `Backports.org Archive Key.asc` sur le bureau si l'on a suivi les conseils précédents — et cliquer sur valider. On pourra ensuite supprimer le fichier en question.

Mettre à jour les paquets disponibles

Il est maintenant possible de refermer les *Sources de mises à jour*. Le logiciel propose alors (en anglais) de recharger les listes de paquets. Accepter en cliquant sur *Reload*.

Installer le paquet avec les clés du dépôt

Une fois la clé ajoutée, on a accès au dépôt. Celui-ci fournit souvent un paquet contenant les clés de ce dépôt, et permettant de les mettre aisément à jour. Il est souvent nommé à partir du nom du dépôt, suivi du mot `keyring`. Par exemple, pour *backports.org*, il s'agit de `debian-backports-keyring`. Il faut donc prendre le temps d'installer ce paquet, s'il est disponible.

page 119

16.5 APT Pinning

L'*APT Pinning* permet, lorsqu'un même paquet est disponible dans plusieurs dépôts, de dire au système quelle version installer.

Il est principalement utilisé pour installer un logiciel dans une version plus récente que celle disponible dans la distribution *stable* de Debian — soit à partir de paquets en provenance des *backports*, soit en provenance de la version en cours de développement de Debian. Il permet également de s'assurer que les paquets qu'on a pu choisir de cette manière seront maintenus à jour.

Pour régler ces *préférences*, il faut modifier un fichier texte, car il n'existe pas d'interface graphique pour le faire. La documentation complète de ce fichier se trouve dans la page de manuel `apt_preferences(5)` [5].

Ce fichier n'est accessible que par le super-utilisateur (*root*). On va donc ouvrir un *Terminal administrateur* dans lequel on tape : page 87

```
gedit /etc/apt/preferences
```

S'ouvre ensuite la fenêtre de l'éditeur. Si c'est la première fois que l'on touche aux *préférences*, le fichier sera vide.

Commençons par indiquer au système que la version *stable* est celle que l'on préfère. Pour cela, il faut préciser deux niveaux de priorité, un pour la version stable, et un pour le reste :

```
Explanation: La version stable avant les autres
Package: *
Pin: release a=stable
Pin-Priority: 900

Explanation: Les autres version uniquement si on le décide
Package: *
Pin: release o=Debian
Pin-Priority: 100
```

Si on utilise les *backports*, il est également conseillé d'ajouter :

```
Explanation: Mises à jour automatiques pour les backports
Package: *
Pin: release a=lenny-backports
Pin-Priority: 200
```

Pour installer ensuite un paquet particulier provenant de la version en développement (`testing`), par exemple l'éditeur de sous-titres Gaupol, on ajoute :

```
Explanation: La dernière version de Gaupol gère mieux les traductions
Package: gaupol
Pin: release a=testing
Pin-Priority: 990
```

L'utilisation de paquets en provenance de la version de développement est parfois périlleuse. Plus simplement, il existe des paquets spécialement prévus (et testés) pour la version *stable* et contenant des versions plus récentes. Par exemple, pour installer la version des *backports* (`lenny-backports`) du logiciel de téléphonie Twinkle, il suffit d'ajouter :

5. En plus d'être accessible sur tous systèmes Debian, on peut également lire cette page de manuel en ligne [http://manpages.debian.net/cgi-bin/man.cgi?query=apt_preferences&locale=fr].

```
Explanation: La dernière version de Twinkle en provenance de backports
Package: twinkle
Pin: release a=lenny-backports
Pin-Priority: 990
```

Une fois les modifications terminées, on sauvegarde via l'icône de la barre d'outils, et on peut quitter l'éditeur en fermant la fenêtre. On peut ensuite également fermer le terminal.

Il est maintenant possible de relancer *Synaptic* et d'y recharger la liste des paquets disponibles en cliquant sur l'icône *Recharger*. Pour disposer de la nouvelle version de Gaupol ou de Twinkle, il suffit de lancer une mise à jour, s'il est déjà installé, ou de demander l'installation du paquet, s'il ne l'est pas encore.

Chapitre 17

Effacer des données « pour de vrai »

On a vu dans la première partie que lorsqu'on efface un fichier, son contenu n'est pas vraiment supprimé. Cependant, il existe des programmes qui permettent d'effacer des fichiers *et leur contenu*, ou du moins qui tentent de le faire, avec les limites expliquées auparavant.

page 31
page 32

17.1 Un peu de théorie

Pour la plupart des prochaines recettes, nous allons utiliser les logiciels contenus dans le paquet Debian `secure-delete`.

La méthode de Gutmann

La documentation [1] de ce paquet nous dit (en anglais) :

Le processus d'effacement fonctionne comme suit :

1. *la procédure d'écrasement (en mode sécurisé) remplace le contenu du fichier à 38 reprises. Après chaque passage, le cache du disque est vidé ;*
2. *le fichier est tronqué, de sorte qu'un attaquant ne sache pas quels blocs du disque appartenaient au fichier ;*
3. *le fichier est renommé, de sorte qu'un attaquant ne puisse tirer aucune conclusion sur le contenu du fichier supprimé à partir de son nom ;*
4. *finalement, le fichier est supprimé. […]*

Le protocole décrit ci-dessus est basé sur une publication de Peter Gutmann publiée en 1996 [2].

Le compromis adopté

Les 38 écritures mentionnées ci-dessus proviennent de l'étude de Peter Gutmann. Mais cette dernière porte sur des technologies de disques durs qui n'existent plus de nos jours. Il a depuis ajouté, à la fin de son article, un paragraphe intitulé *Epilogue*

1. Fichier `README.gz` installé sur une Debian dans `/usr/share/doc/secure-delete`.
2. Peter Gutmann, *Secure Deletion of Data from Magnetic and Solid-State Memory* [http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html], *Department of Computer Science*, *University of Auckland*, 1996.

qui nous dit, en substance, que pour un disque dur « récent » [3], il suffit d'écraser les données quelques fois avec des données aléatoires. Mais mise à part la nature et le nombre de réécritures, le processus décrit précédemment reste tout à fait d'actualité.

De surcroît, le NIST (*National Institute of Standards and Techonology*, organisme gouvernemental états-unien définissant les protocoles de sécurité utilisés, entre autres, par les administrations de ce pays) a publié une étude récente [4] de la NSA, qui semble conclure que sur les disques durs modernes, les données sont tellement collées les unes aux autres qu'il devient impossible de se livrer à des analyses magnétiques pour retrouver les traces de données effacées ; en effet, la densité des données des disques durs ne cesse de croître, afin d'augmenter leur capacité de stockage.

Par conséquent, nous nous contenterons de quelques passages aléatoires dans les recettes qui suivent, tout en précisant comment mettre en œuvre la méthode originale de Gutmann.

page 51 Il s'agira ici encore de faire le bon compromis, au cas par cas, entre la rapidité et le niveau de protection souhaité, en fonction de la taille des données à écraser, de l'âge du disque dur, et de la confiance qu'on accorde au NIST.

Les limites de l'effacement « sécurisé »

Attention : il peut encore rester des informations sur le fichier permettant de le retrouver, notamment si l'on utilise un système de fichiers journalisé comme *ext3*, ReiserFS, XFS, JFS, NTFS, un système d'écriture, de compression ou de sauvegarde, sur disque (exemple : RAID) ou via un réseau. Voir à ce sujet la page 33 première partie.

17.2 Sur d'autres systèmes

page 29 On a vu qu'il est illusoire, si l'on utilise un système d'exploitation propriétaire de rechercher une réelle intimité. Bien qu'il existe des logiciels supposés effacer des fichiers avec leur contenu sous Windows et Mac OS X, il ne faut pas leur faire confiance.

17.3 Allons-y

On peut effacer le contenu :

- de fichiers individuels, voir page suivante ;
- de tout un périphérique, voir page 132 ;
- de fichiers déjà supprimées, voir page 138.

3. Utilisant la technologie PRML [https://secure.wikimedia.org/wikipedia/fr/wiki/PRML], apparue en 1990 [http://www.storagereview.com/guide/histFirsts.html].

4. *Special Publication 800–88: Guidelines for Media Sanitization* [http://csrc.nist.gov/publications/nistpubs/800-88/NISTSP800-88_rev1.pdf].

17.4 Supprimer des fichiers… et leur contenu

Voici donc la méthode à suivre pour se débarrasser de fichiers, en prenant soin de rendre illisible ce qu'ils contenaient.

Installer les logiciels nécessaires

Si le paquet `secure-delete` n'est pas encore installé, l'installer. page 119

Supprimer des fichiers et leur contenu à partir du navigateur de fichiers

Il est possible de configurer le navigateur de fichiers du bureau GNOME pour pouvoir effacer des fichiers avec leur contenu, voir page suivante.

En ligne de commande

Si l'on est à l'aise avec l'utilisation d'un terminal, la suppression de fichiers *et de leur contenu* avec `srm` est simple. Il suffit d'exécuter la commande :

```
srm -r -l -v NOM_DU_FICHIER
```

Note : les options `-r`, `-l` et `-v` qu'on propose d'utiliser ici ont, dans le cadre de la commande `srm`, la signification suivante :

- L'option `-r` indique qu'on veut effacer la cible de manière *récursive*, c'est-à-dire en incluant les sous-dossiers s'il y en a.
- L'option `-l` indique qu'on veut que `srm` écrase le contenu des fichiers deux fois de suite, dont une fois avec des données aléatoires. Si l'on préfère utiliser la méthode originale de Gutmann (plus longue, et peut-être plus sûre), il suffit de ne pas utiliser cette option.
- L'option `-v` indique qu'on veut utiliser le mode *verbose* (bavard) lors de l'exécution de la commande : ainsi, le terminal indiquera au fur et à mesure les actions qu'il effectue. Cela permet notamment de suivre la progression de la commande, en ajoutant une étoile (`*`) après chaque passage d'effacement sur le fichier.

Sur ce sujet, on peut jeter un œil à la partie sur la ligne de commande page 87

17.5 Ajouter à Nautilus une commande pour effacer des fichiers et leur contenu

Pour utiliser la commande `srm` depuis le bureau graphique GNOME, on va ajouter un tout petit programme (un *script*) au navigateur de fichiers de GNOME (qui s'appelle Nautilus).

Installer les logiciels nécessaires

page 119

Si le paquet `secure-delete` n'est pas encore installé, l'installer.

Télécharger ou écrire le script

Afin d'ajouter ce petit programme, deux possibilités : le télécharger si on a accès à Internet ou le recopier (en se relisant plusieurs fois).

Première option : télécharger le script

- Télécharger le script `Supprimer_en_ecrasant_les_donnees` à partir de l'adresse : https://guide.boum.org/tomes/1_hors_connexions/3_outils/06_effacer_pour_de_vrai/02_wipe_dans_Nautilus/Supprimer_en_ecrasant_les_donnees

page 163

- Vérifier sa somme de contrôle. Attention cependant : croire ce qui est écrit ici revient à accorder sa confiance en l'ensemble du processus par lequel on a obtenu ce document, ce qui n'est pas forcément une bonne idée. Voici tout de même sa somme de contrôle SHA256 :
  `2fd3abd941d50572602aeea5dc7a6be62eb8ccdf0d29fc4638154122752a4c54`

Deuxième option : écrire le script

Quand il est impossible de télécharger le script, il faut l'écrire soi-même, en suivant les instructions suivantes.

- Ouvrir l'*Éditeur de texte* qui se trouve dans le menu *Applications* puis *Accessoires*.
- Écrire, sur la page blanche qui est apparue :

```
#!/bin/bash
if zenity --question \
    --text "Voulez-vous vraiment supprimer ${*} en écrasant son contenu ?" \
    --title "Supprimer en écrasant les données"; then
  srm -rf "$@" && \
  zenity --info --text "${*} a bien été supprimé." \
      --title "Supprimer en écrasant les données" ||
  zenity --error \
      --text "Une erreur est survenue durant l'effacement de ${*}." \
      --title "Supprimer en écrasant les données"
fi
```

- Enregistrer le fichier en cliquant dans le menu *Fichier* sur *Enregistrer*. Le nommer `Supprimer_en_ecrasant_les_donnees` et le ranger sur le bureau (*Desktop*).
- Quitter l'*Éditeur de texte*.

Copier le script là où le navigateur de fichiers le cherche

- Sélectionner le fichier `Supprimer_en_ecrasant_les_donnees` sur le bureau.
- Cliquer dessus avec le bouton droit de la souris. Dans le menu contextuel qui apparaît, cliquer sur *Couper*.
- Ouvrir le *Navigateur de fichiers* qui se trouve dans le menu *Applications* → *Outils système*.
- Dans le menu *Aller à* → *Emplacement…*, puis taper `~/.gnome2/nautilus-scripts/` et appuyer sur la touche *Entrée*.
- Coller le fichier en cliquant dans le menu *Édition* sur *Coller*.

Rendre le script exécutable

- Sélectionner le fichier `Supprimer_en_ecrasant_les_donnees`.
- Cliquer dessus avec le bouton droit de la souris. Dans le menu contextuel qui apparaît, cliquer sur *Propriétés*.
- Dans la boîte de dialoque qui apparaît, aller dans l'onglet *Permissions*, cocher la case *Exécution*.
- Fermer la boîte en cliquant sur *Fermer*.

Vérifier

- Dans le menu contextuel du navigateur de fichiers, un sous-menu *Scripts* contenant une commande `Supprimer_en_ecrasant_les_donnees` devrait apparaître.

Utiliser le script

- Sélectionner les fichiers et dossiers à supprimer.
- Cliquer dessus avec le bouton droit de la souris. Dans le menu contextuel qui apparaît, cliquer sur *Scripts*, puis sur `Supprimer_en_ecrasant_les_donnees`.

17.6 Effacer pour de vrai tout un disque

page 57

Avant de se débarrasser d'un disque dur, de le recycler ou encore de réinstaller un système propre, il peut être judicieux de mettre des bâtons dans les roues des gens qui voudraient récupérer les données qu'il contenait. Pour cela, la meilleure solution est encore de les remplacer par du charabia.

Avant d'utiliser cette recette, il faut réfléchir à deux fois et sauvegarder soigneusement les données à conserver. Si elle est bien appliquée, elle rend en effet les données très difficiles à récupérer, même en analysant le disque dans un laboratoire.

Nous verrons d'abord comment effacer tout le contenu d'un disque, puis comment rendre le contenu d'une partition chiffrée inaccessible rapidement.

17.7 Effacer tout le contenu d'un disque

Pour effacer un volume complet (disque ou partition), on va utiliser la commande `shred` de façon à ce qu'elle recouvre la totalité des données trois fois avec des données aléatoires. Cette commande permet donc, en plus de l'effacement des fichiers, de recouvrir l'espace effacé de telle manière qu'il devient quasiment impossible de retrouver ce qu'il contenait auparavant.

Pour recouvrir le contenu d'un disque, il est nécessaire de ne pas être en train de l'utiliser… Si ce dernier contient le système d'exploitation habituellement utilisé, il faut donc mettre le disque dur dans un autre ordinateur ou utiliser un système *live*. `shred` étant un outil standard, n'importe quel système *live* devrait faire l'affaire.

page 101

Une fois le système démarré sur un autre support que le disque à effacer (un système *live* par exemple), la commande est très simple. Elle exige seulement de connaître l'emplacement du périphérique que l'on veut effacer, puis de faire preuve de patience car le processus prend plusieurs heures.

Trouver le nom du périphérique

Avant tout, il faut savoir repérer sans se tromper l'emplacement du support de stockage qu'on veut effacer. Nous appellerons ce nom `LE_PÉRIPHÉRIQUE`.

Pour un disque externe

page 144

S'il s'agit d'un média externe, utiliser la méthode pour trouver le nom du périphérique décrite dans la recette sur le chiffrement d'un disque externe.

Attention : pour cette recette, une fois le nom du périphérique repéré :

- on ne prendra pas en compte le chiffre final qui le désigne ;
- on ajoutera `/dev/` au début.

C'est-à-dire que si on a trouvé comme nom `sdx1`, on va noter `/dev/sdx`. On va appeler cet emplacement `LE_PÉRIPHÉRIQUE`.

Pour un disque interne

Repérer un disque branché à l'intérieur de l'ordinateur est un peu plus compliqué, car il est impossible de le brancher « à chaud » pour voir ce qui apparaît.

Nous allons utiliser l'outil de partitionnement fourni avec le bureau GNOME. Là encore, le système *live* fera l'affaire.

- Tout d'abord, débrancher tous les disques durs externes, clés USB, lecteurs de cartes mémoire ou autres périphériques de stockage branchés sur l'ordinateur. D'une part, cela évitera de les effacer par erreur ; d'autre part, cela rendra la recherche du disque interne plus facile.
- On va utiliser *Partition Editor*, que l'on ouvre à partir du menu *Système* puis *Administration*. Cet outil permet de modifier l'organisation des disques et peut supprimer toutes vos données. C'est donc un programme qui demande le mot de passe du super-utilisateur.

Vers le haut de la fenêtre de *Partition Editor*, à droite de la barre d'icônes, on peut lire le nom du disque actuellement affiché, ainsi que sa taille entre parenthèses.

Pour vérifier que c'est le bon disque, il est possible de jeter un œil à l'organisation des partitions que *Partition Editor* montre :

- si le disque à effacer contenait un système GNU/Linux non chiffré, il doit y avoir au moins deux partitions, l'une avec un système de fichiers *swap*, l'autre en général *ext3* ;
- si le disque à effacer contenait un système GNU/Linux chiffré, il doit y avoir au moins deux partitions, l'une avec un système de fichiers *ext2*, l'autre en général *crypt-luks* ou *inconnu* ;
- si le disque à effacer contenait un système Windows, il doit y avoir une ou plusieurs partitions notées *ntfs* ou *fat32*.

Le périphérique correspondant au disque interne est en général le premier que *Partition Editor* trouve.

Si l'organisation du premier disque ne correspond pas à celle du disque à effacer, il faut parcourir les autres disques disponibles à partir de la liste qui apparaît après avoir cliqué sur l'icône représentant un disque dur (en haut à droite).

Reste, à l'aide de leur capacité et de leur organisation, à déterminer le bon.

Une fois cela fait, noter dans un coin le nom du périphérique, quelque chose comme `/dev/sdx` ou `/dev/hdx` (toujours en haut à droite). C'est cette valeur que nous appellerons `LE_PÉRIPHÉRIQUE`.

Ouvrir un terminal administrateur

À partir du menu *Applications* → *Accessoires*, ouvrir un *Terminal administrateur*.

Lancer la commande *shred*

Taper en veillant à remplacer `LE_PÉRIPHÉRIQUE` par le nom de périphérique déterminé précédemment :

```
shred -n 3 -v LE_PÉRIPHÉRIQUE
```

Si l'on préfère utiliser la méthode originale de Gutmann (plus longue, et peut-être plus sûre), il faut remplacer `-n 3` par `-n 25` dans la ligne de commande.

Une fois la commande tapée et vérifiée, appuyer sur la touche *Entrée*. La commande `shred` va alors écrire dans le terminal ce qu'elle fait (ainsi qu'on lui a demandé de le faire en ajoutant à la commande `shred` l'option `-v`, qui signifie, dans le cadre de *cette* commande, que l'ordinateur doit être « verbeux » — c'est-à-dire « bavard ») :

```
shred: /dev/hdb: pass 1/3 (random)...
shred: /dev/hdb: pass 2/3 (random)...
shred: /dev/hdb: pass 3/3 (random)...
```

À la fin de la procédure, on peut fermer le terminal.

Réutiliser le disque

Attention, cette méthode efface non seulement les données d'un volume complet mais, à la fin de l'opération, le disque n'a plus ni de table de partitions ni de système de fichiers. Pour le réutiliser, il est nécessaire de créer entièrement au moins une nouvelle partition et son système de fichiers, avec *Partition Editor* par exemple.

page 16
page 16

17.8 Effacer le contenu d'une partition chiffrée LUKS

Certains logiciels de chiffrement d'un disque complet ont la capacité de détruire la clé de chiffrement, rendant ainsi le contenu chiffré incompréhensible. Vu que la clé contient une part minuscule d'informations et peut être détruite presque instantanément, cette méthode est une alternative bien plus rapide à l'écrasement de l'ensemble des données. Ceci dit, cette option n'est réalisable que si le disque dur a déjà été chiffré. Si les données confidentielles contenues sur le disque ne sont pas déjà chiffrées, il est nécessaire d'effacer le disque entier, comme expliqué précédemment, avant de pouvoir s'en débarrasser ou le réutiliser en toute tranquillité.

page 132

Il est extrêment rapide de rendre inaccessible le contenu d'une partition LUKS, le format de stockage standard des clés de disques chiffrés sous GNU/Linux.

Repérer la partition en question

Comme dans le cas précédent, si l'on souhaite effacer un disque interne, commencer par débrancher tous les disques durs externes, clés USB, lecteurs de cartes mémoire ou autre périphérique de stockage branché sur l'ordinateur. D'une part, cela évitera de les effacer par erreur ; d'autre part, cela rendra la recherche du disque interne plus facile.

Bien sûr, il ne faut pas faire cela si c'est justement le contenu d'un disque externe que l'on souhaite rendre inaccessible.

Ouvrir Partition Editor

On va s'aider de *Partition Editor*, que l'on ouvre à partir du menu *Système* puis *Administration*. Cet outil permet de modifier l'organisation des disques et peut supprimer toutes vos données. C'est donc un programme qui demande le mot de passe du super-utilisateur.

Chercher le périphérique à effacer

En général, le premier périphérique que *Partition Editor* affiche est le disque dur interne. Il est possible de parcourir les disques trouvés à l'aide de la liste déroulante affichée en haut à droite de la fenêtre.

Normalement, les partitions chiffrées sont indiquées avec un type de fichier *crypt-luks…* mais ce n'est pas toujours le cas : *Partition Editor* peut aussi indiquer que le type de partition est *inconnu*, voir parfois, indiquer un type plus courant (*ext3*, NTFS). S'il s'agit d'un support externe, on peut compléter ce repérage en utilisant la recette correspondante.

page 144

Une fois repérée la partition chiffrée à effacer, noter le nom du périphérique correspondant. Nous appellerons cette valeur `LE_PÉRIPHÉRIQUE_CHIFFRÉ`. Ce doit être quelque chose comme `/dev/sdx9` ou `/dev/hdx3`.

Ouvrir un terminal administrateur

À partir du menu *Applications* → *Accessoires*, ouvrir un *Terminal administrateur*.

Vérifier le premier repérage et récupérer la taille de l'en-tête LUKS

Dans le terminal, la commande `cryptsetup luksDump` donne plein d'informations sur l'en-tête LUKS, dont sa taille sur le disque (en secteurs de 512 octets). Taper donc, en remplacant `LE_PÉRIPHÉRIQUE_CHIFFRÉ` par la valeur déterminée ci-dessus :

```
cryptsetup luksDump LE_PÉRIPHÉRIQUE_CHIFFRÉ
```

Dans le cas où on se serait trompé de périphérique, le terminal ne renvoie soit aucune réponse, soit :

```
Command failed: LE_PÉRIPHÉRIQUE_CHIFFRÉ is not a LUKS partition.
```

Si l'on ne s'est pas trompé, on doit plutôt se voir répondre quelque chose comme :

```
LUKS header information for /dev/sdb2

Version:        1
Cipher name:    aes
Cipher mode:    cbc-essiv:sha256
Hash spec:      sha1
Payload offset: 2056
MK bits:        256
MK digest:      a4 79 85 49 1f 3f 71 e5 1e c6 07 14 88 0c 02 27
                59 80 25 58
MK salt:        b7 b1 2a 5d 6d c5 b5 d2 06 55 a3 85 5d 07 af 9b
                c9 03 46 c6 e6 2f 29 1a 9d b7 58 05 44 cc 68 f9
MK iterations:  10
UUID:           d73cbb8a-058f-469e-935a-7f71debd8193

Key Slot 0: ENABLED
    Iterations:         170901
    Salt:               ec 1e 63 b7 13 fb 20 21 18 5d 86 44 42 d0
                        f2 af 52 a4 74 54 22 3f d8 0b ad 69 8c 46
                        f2 d3 79 4d
    Key material offset:8
    AF stripes:         4000
```

On va avoir besoin de la taille de l'en-tête (en secteurs), écrite sur la ligne `Payload offset` : la noter quelque part. On l'utilisera plus loin sous le nom d'`OFFSET`.

Recouvrir l'en-tête LUKS de données aléatoires

Comme dans la recette précédente, on va utiliser la commande `shred` pour écraser les données, mais cette fois on écrasera uniquement l'en-tête LUKS (cet en-tête est la clé qui permet de déchiffrer le reste des données). Cela ira donc beaucoup plus vite.

Dans le terminal administrateur, taper, en prenant bien soin de remplacer `OFFSET` et `LE_PÉRIPHÉRIQUE_CHIFFRÉ` par les valeurs qu'on a trouvées :

```
shred -n 3 -s $((OFFSET * 512)) -v LE_PÉRIPHÉRIQUE_CHIFFRÉ
```

page 87

Note : l'option `-s` utilisée ici sert, dans le cadre de *cette* commande, à indiquer la taille (*size*) de l'espace qui doit être effacé de manière sécurisée.

Les données chiffrées devraient maintenant être illisibles. Pour s'en assurer, il est possible de chercher un en-tête LUKS qui n'aurait pas été bien effacé en tapant à nouveau :

```
cryptsetup luksDump LE_PÉRIPHÉRIQUE_CHIFFRÉ
```

Si l'en-tête a bien été effacé, le terminal renvoie soit aucune réponse, soit :

```
Command failed: LE_PÉRIPHÉRIQUE_CHIFFRÉ is not a LUKS partition
```

Enfin il est possible, voire même conseillé, d'effacer quand même l'ensemble de la partition, en suivant la recette précédente. page 132

17.9 Rendre irrécupérables des données déjà supprimées

Lorsque des fichiers ont *déjà* été effacés sans précautions particulières, les données qu'ils contenaient se trouvent toujours sur le disque. La commande `sfill` qui est fournie par le paquet `secure-delete` s'occupe de recouvrir les données qui restent sur l'espace libre d'un disque dur.

Il est intéressant de la lancer en tant que super-utilisateur, pour que les parties du disque réservées à celui-ci (appelées « blocs réservés ») soient aussi effacés.

Attention : comme les autres façons d'effacer un fichier « pour de vrai », cela ne marche pas avec certains systèmes de fichiers « intelligents » qui, pour être plus efficaces, ne vont pas donner à `sfill` tout l'espace libre. Voir à ce sujet la première partie.

page 33

Installer les logiciels nécessaires

Si le paquet `secure-delete` n'est pas encore installé, l'installer.

page 119

Rendre irrécupérables des données déjà supprimées à partir du navigateur de fichiers

Il est possible de configurer le navigateur de fichiers du bureau GNOME pour pouvoir rendre irrécupérables des données déjà supprimées, voir page 140.

En ligne de commande

Attention : la méthode décrite ci-dessous ne fonctionne pas correctement sur les systèmes de fichiers FAT32.

Pour vérifier le système de fichiers d'une partition, on peut faire un clic droit sur l'icône du disque sur le bureau. Puis une fois la fenêtre *Propriétés* ouverte, à la fin de l'onglet *Général*, on peut lire *Type de système de fichiers*. Si l'ordinateur indique *vfat* ou *fat*, alors `sfill` ne recouvrira l'espace libre que si ce dernier fait moins de 4 Go ! Dans ce cas, mieux vaut utiliser la méthode basée sur le navigateur de fichiers, qui a l'avantage de fonctionner correctement sur un système de fichiers FAT32.

page 16

page 140

Ouvrir un terminal administrateur

Ouvrir un terminal, en cliquant sur le menu *Applications*, puis *Accessoires* et enfin *Terminal administrateur*.

Repérer l'emplacement à nettoyer et lancer `sfill`

Avant de lancer la commande, il faut indiquer à sfill un dossier qui se trouve sur la partition (partie de disque) à l'intérieur de laquelle on souhaite rendre plus difficile la récupération des fichiers déjà supprimés. Choisir donc n'importe quel dossier situé sur cette partition : on l'appellera `DOSSIER`.

page 16

Dans le terminal, taper alors :

```
sfill -l -v DOSSIER
```

Et valider la commande en appuyant sur la touche *Entrée*.

L'option `-l` demande à `sfill` de recouvrir l'espace libre à deux reprises. Si l'on préfère utiliser la méthode originale de Gutmann (plus longue, et peut-être plus sûre), il faut ôter cette option de la ligne de commande.

Un exemple

Prenons un exemple typique : on souhaite recouvrir l'espace libre de la partition sur laquelle se trouve son dossier personnel. Pour cela, il faut trouver son identifiant de connexion, ou *login* — celui que l'on tape avant son mot de passe lorsqu'on se connecte à sa session. C'est aussi lui qui s'affiche dans le début de la barre de titre du navigateur de fichiers quand on ouvre son dossier personnel. Nous l'appellerons `LOGIN`. Taper alors dans le terminal administrateur, en remplaçant `LOGIN` par son identifiant de connexion :

```
#> sfill -l -v '/home/LOGIN'
```

Pour l'utilisatrice `lucienne`, cela donnerait :

```
#> sfill -l -v '/home/lucienne'
```

Ensuite, patienter très longtemps (de nombreuses heures), surtout si l'on a un gros disque.

Un compromis possible

Si après avoir essayé `sfill`, on constate qu'il est vraiment trop lent pour l'usage que l'on souhaite en faire, il est intéressant de savoir qu'on peut donner l'option `-l` une seconde fois à `sfill`, pour effacer de façon moins sûre mais plus rapide : ainsi, au lieu de faire deux réécritures, `sfill` n'en fera qu'une — avec des données aléatoires. C'est moins sûr que la méthode précédente, mais c'est mieux que de ne pas lancer `sfill` du tout.

Pour ce faire, il faut lancer `sfill` de la façon suivante :

```
#> sfill -l -l -v DOSSIER
```

17.10 Ajouter à Nautilus une commande pour rendre irrécupérables des données déjà supprimées

page 138 Pour pouvoir effectuer le processus décrit précédemment à partir du navigateur de fichiers de GNOME, on peut lui ajouter un petit programme supplémentaire (un *script*).

Ce programme a l'avantage d'effectuer le recouvrement du contenu de l'espace libre en créant plusieurs fichiers. Ce mécanisme lui permet donc de fonctionner correctement page 16 sur un système de fichiers FAT32.

Installer les paquets nécessaires

page 119 Il est nécessaire d'ajouter le paquet le paquet `secure-delete` au système si ce n'est pas déjà fait.

Télécharger ou écrire le script

Afin d'ajouter ce petit programme, deux possibilités : le télécharger si on a accès à Internet ou le recopier (en se relisant plusieurs fois).

Première option : télécharger le script

- Télécharger le script `Ecraser_l_espace_libre_de_cette_partition` à partir de l'adresse : https://guide.boum.org/tomes/1_hors_connexions/3_outils/06_effacer_pour_de_vrai/07_sfill_dans_nautilus/Ecraser_l_espace_libre_de_cette_partition
- page 163 Vérifier sa somme de contrôle. Attention cependant : croire ce qui est écrit ici revient à accorder sa confiance en l'ensemble du processus par lequel on a obtenu ce document, ce qui n'est pas forcément une bonne idée. Voici tout de même sa somme de contrôle SHA256 :
  `c907691c03d12ad2eadc2ca9758615580d663695b503f6579bef6afa111ccff9`

Deuxième option : écrire le script

Quand il est impossible de télécharger le script, il faut l'écrire soi-même :

- Ouvrir l'*Éditeur de texte* dans le menu *Applications* → *Accessoires*.
- Écrire, sur la page blanche qui est apparue :

```
#!/bin/sh

test -z "$PWD" && exit 1
mkdir -p "$PWD/ECRASEMENT"
trap "rm -rf $PWD/ECRASEMENT" EXIT

{ (echo 0
MAX=4000000
FREE="$(df -P "$PWD" | awk '/\// { print $4 }')"
if [ "$FREE" -gt "$MAX" ]; then
    for n in $(seq 0 $((90 / ($FREE / $MAX))) 90); do
        echo "$n"
        FILE="$PWD/ECRASEMENT/$FREE.$n.$$"
        echo "# Écrasement de $FILE"
        dd if=/dev/zero of="$FILE" seek="$MAX" bs=1k count=1
        shred -n 3 "$FILE"
```

```
    done
    echo 90
fi
echo "# Écrasement de l'espace libre restant"
RESULT=$(gksu --description "sfill" "sh -c '
sfill -l -l \"$PWD/ECRASEMENT\" &&
sfill -l -l \"$PWD/ECRASEMENT\" &&
sfill -l -l \"$PWD/ECRASEMENT\" || echo ERROR")
test "$RESULT" = "ERROR" && exit 1
rm -rf "$PWD/ECRASEMENT"
echo 100
echo "# Écrasement de l'espace libre terminé avec succès"
) || {
echo "# Une erreur est survenue."
zenity --error \
  --text "Une erreur est survenue pendant l'écrasement de l'espace libre." \
  --title "Écrasement de l'espace libre"
} ; } | zenity --progress --title "Écrasement de l'espace libre"
```

- Enregistrer le fichier via *Fichier* → *Enregister*. Le nommer `Ecraser_l_espace_libre_de_cette_partition` et le ranger sur le bureau (*Desktop*).
- Quitter l'*Éditeur de texte*.

Copier le script là où le navigateur de fichiers le cherche

- Sélectionner le fichier `Ecraser_l_espace_libre_de_cette_partition` sur le bureau.
- Cliquer dessus avec le bouton droit de la souris. Dans le menu contextuel qui apparaît, cliquer sur *Couper*.
- Ouvrir le *Navigateur de fichiers* via le menu *Applications* → *Outils systèmes*.
- Dans le menu *Aller à* → *Emplacement…*, entrer `~/.gnome2/nautilus-scripts/` et appuyer sur la touche *Entrée*.
- Coller le fichier à partir du menu *Édition* → *Coller*.

Rendre le script exécutable

- Sélectionner le fichier `Ecraser_l_espace_libre_de_cette_partition`.
- Cliquer dessus avec le bouton droit de la souris. Dans le menu contextuel qui apparaît, sélectionner *Propriétés*.
- Dans la boîte de dialogue qui s'affiche, aller dans l'onglet *Permissions*, cocher la case *Exécution*.
- Fermer la boîte en cliquant sur *Fermer*.

Vérifier

- Dans le menu contextuel du navigateur de fichiers, un sous-menu *Scripts* contenant une commande `Ecraser_l_espace_libre_de_cette_partition` devrait apparaître.

Utiliser le script

- Ouvrir un dossier qui se trouve dans la partition dont l'espace libre devra être écrasé. Cela peut être, par exemple, une clé USB ou un disque externe qui n'est pas en train d'être utilisé.
- Cliquer sur le fond de la fenêtre (sans sélectionner de fichiers ou de dossiers) avec le bouton droit de la souris. Dans le menu contextuel qui apparaît, cliquer sur *Scripts*, puis sur `Ecraser_l_espace_libre_de_cette_partition`.

Chapitre 18

Partitionner et chiffrer un disque dur

Nous allons voir comment chiffrer un disque, et non comment installer un système GNU/Linux chiffré. Il peut s'agir d'un disque dur externe, d'une clé USB ou encore d'une partie seulement d'un disque dur ou d'une clé USB. On peut en effet découper un disque dur ou une clé USB en plusieurs morceaux indépendants, qu'on appelle des partitions.

page 105
page 16

Ci-dessous, on parlera de disque dur, sachant que ça vaut aussi bien pour un disque dur externe que pour une clé USB, sauf mention contraire.

Une fois un disque dur chiffré, les données qu'il contient ne sont accessibles que lorsqu'on a tapé une phrase de passe permettant de le déchiffrer. Pour plus d'informations là-dessus, voir la partie sur la cryptographie. Une fois la phrase de passe tapée, le système a accès aux données du disque dur en question. Il ne faut donc pas taper cette phrase de passe n'importe où, mais seulement sur les ordinateurs et les systèmes en qui on a une confiance suffisante.

page 37
page 51

Il faut également partir du principe que, sauf avec un système *live*, des traces de la présence du disque dur seront gardées par l'ordinateur utilisé.

page 101

Si on veut avoir un endroit sur le disque dur où mettre des données non confidentielles auxquelles on veut accéder sur des ordinateurs indignes de confiance, il est possible de découper le disque dur en deux partitions :

- une partition non chiffrée, où l'on ne met que des données non confidentielles, comme de la musique, que l'on peut utiliser depuis tous les ordinateurs sans taper la phrase de passe ;
- une partition chiffrée, avec les données confidentielles, qu'on n'ouvre que sur les ordinateurs auxquels on fait confiance.

18.1 Chiffrer un disque dur avec LUKS et `dm-crypt`

On va expliquer comment chiffrer un disque avec les méthodes standard sous GNU/Linux, appelées `dm-crypt` et LUKS. Pour l'instant, il n'est pas possible de réaliser cette opération à partir d'une interface graphique. On va donc utiliser un terminal.

page 87

18.2 D'autres logiciels que l'on déconseille

Il existe d'autres logiciels de chiffrement comme FileVault, qui est intégré dans Mac OS X — mais il s'agit d'un logiciel propriétaire — ou TrueCrypt — mais on

page 23

a moins de raisons de lui faire confiance que le chiffrement standard de GNU/Linux, car ce n'est pas vraiment un logiciel libre [1]. De plus, si l'on utilise un logiciel, même libre, sur un système d'exploitation propriétaire, on fait implicitement confiance à ce dernier car il a forcément accès aux données déchiffrées.

page 15

18.3 En pratique

page 127

Si nécessaire, commencer par recouvrir les données que le disque dur a pu contenir dans le passé.

plus bas

Après quoi, il faut trouver le nom du périphérique correspondant au disque dur.

cf. ci-contre

Si l'on souhaite chiffrer une partie seulement du disque dur, il faut maintenant le partitionner.

page 146

À la suite de quoi, qu'on souhaite chiffrer un disque dur entier ou qu'on l'ait partitionné à l'instant, il ne reste plus qu'à l'initialiser pour contenir des données chiffrées.

page 148

Enfin, on peut l'utiliser.

18.4 Trouver le nom d'un disque dur

Le disque dur doit être débranché matériellement. S'il est déjà branché, le débrancher proprement, en le démontant si nécessaire (pour démonter le disque dur externe, aller sur le bureau et cliquer sur l'icône du disque dur avec le bouton droit. Dans le menu contextuel qui apparaît, cliquer sur *Démonter le volume*).

Ouvrir le *Visionneur de journaux système* qui se trouve dans le menu *Applications* → *Outils système*.

Dans la fenêtre de gauche, cliquer sur `messages`.

Brancher le disque dur externe. Quelques lignes devraient apparaître à la toute fin du texte. Par exemple :

```
...057] usb 3-2: new full speed USB device using uhci_hcd and address 2
...279] usb 3-2: New USB device found, idVendor=066f, idProduct=8252
...293] usb 3-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
...304] usb 3-2: Product: Portable MP3 Player
...312] usb 3-2: Manufacturer: Packard
...320] usb 3-2: SerialNumber: 7CF3838EA87B6109
...537] usb 3-2: configuration #1 chosen from 1 choice
...639] Initializing USB Mass Storage driver...
...756] scsi2 : SCSI emulation for USB Mass Storage devices
...352] usbcore: registered new interface driver usb-storage
...358] USB Mass Storage support registered.
...544] scsi 2:0:0:0: Direct-Access Portable Player 0100 PQ: 0 ANSI: 4
...499] sd 2:0:0:0: [sdx] 2041344 512-byte hardware sectors: (996 MiB)
...509] sd 2:0:0:0: [sdx] Write Protect is off
...504]  sdx: sdx1
...762] sd 2:0:0:0: [sdx] Attached SCSI removable disk
```

Ce qu'on veut trouver, c'est le nom du périphérique qui contient la partition du disque dur externe. C'est un nom à trois lettres suivi d'un chiffre, qui commence par `sd` ou

1. TrueCrypt est distribué sous une licence particulière, la « *TrueCrypt Collective License* » : le développement n'est pas ouvert, et seules les sources de la dernière version sont disponibles, ce qui rend plus difficile la vérification des modifications apportées. De plus, le logiciel n'est pas considéré comme libre par nombre de distributions GNU/Linux, notamment Debian et ne correspond pas à la définition de l'*open source* de l'Open Source Initiative [http://www.opensource.org/docs/osd].

`hd`. Dans l'exemple ci-dessus, c'est écrit sur l'avant-dernière ligne, et c'est `sdx1`. Noter cela quelque part : pour l'ordinateur, il s'agit du nom du disque dur externe, qu'on devra écrire tout à l'heure à la place de `LE_PÉRIPHÉRIQUE`.

18.5 Partitionner un disque dur

Si l'on désire chifrer un disque dur en entier, passer directement au point suivant. L'idée de cette partie est d'apprendre à séparer la clé USB ou le disque dur en deux partitions : l'une non chiffrée, où l'on ne met que des données non confidentielles que l'on peut utiliser depuis tous les ordinateurs sans taper la phrase de passe ; l'autre chiffrée, avec les données confidentielles, qu'on ouvre uniquement sur les ordinateurs auxquels on fait confiance.

page suivante

Ci-dessous, on parlera de disque dur, sachant que ça vaut aussi bien pour un disque dur externe que pour une clé USB, sauf si on précise le contraire.

⚠ **Attention**, pour chiffrer le disque dur on va effacer toutes les données qu'il y a dessus !

Partitionner le disque dur

Installer les paquets nécessaires

Pour commencer, on va installer les paquets `dosfstools`, `ntfsprogs`, `cryptsetup` et `secure-delete`.

page 119

Ouvrir Partition Editor

On va utiliser *Partition Editor*, que l'on ouvre à partir du menu *Système* → *Administration*. Cet outil permet de modifier non seulement les clés USB et les disques durs externes, mais aussi le disque interne de l'ordinateur, et peut supprimer toutes vos données. C'est donc un programme qui demande le mot de passe du super-utilisateur.

Choisir le périphérique

Une fois le mot de passe entré, il faut bien faire attention à travailler sur le disque dur externe et non pas sur le disque dur interne de l'ordinateur. Encore une fois : en travaillant sur le disque dur de l'ordinateur, il est possible de perdre toutes ses données !

Pour choisir le disque dur, il faut utiliser le nom du périphérique trouvé tout à l'heure. Dans la boîte déroulante en haut à droite de la fenêtre de *Partition Editor*, il faut choisir `/dev/LE_PÉRIPHÉRIQUE` en remplacant `LE_PÉRIPHÉRIQUE` par ce qu'on a trouvé. *Partition Editor* met entre parenthèses la taille du disque ou de la clé USB sélectionné, ce qui permet de vérifier qu'on a choisi le bon.

page précédente

Démonter la partition

Ensuite, il faut « démonter » le disque dur, c'est-à-dire demander au système de ne plus l'utiliser, sans pour autant le débrancher physiquement. Pour ça, on sélectionne, via le menu, *Partition* → *Démonter*. Si l'entrée est grisée, ce n'est pas grave, ça veut dire qu'il est déjà démonté.

Redimensionner la partition

On va ensuite redimentionner la partition existante, afin de laisser de la place pour la partition chiffrée. À partir du menu *Partition* → *Redimensionner/Déplacer*, on peut choisir avec le curseur la taille de la partition pour les données non confidentielles. L'espace libre après cette partition sera utilisé pour la partition chiffrée.

Un fois satisfait de la taille de cette partition, valider puis aller dans le menu *Édition* → *Appliquer toutes les opérations*. Il s'agit maintenant de s'assurer que le système d'exploitation de l'ordinateur s'aperçoive bien lui aussi de son côté des changements dans le partitionnement de ce périphérique. Pour ce faire, fermer *Partition Editor* et débrancher le périphérique en question, attendre un peu et le rebrancher. Relancer ensuite *Partition Editor* pour continuer la suite du partitionnement.

Créer la deuxième partition où l'on va mettre les données chiffrées

Cliquer sur l'espace gris marqué « non alloué » qui est apparu après l'étape précédente, et aller dans le menu *Partition* → *Nouveau*. Accepter les valeurs par défaut en cliquant sur *Ajouter* : *Partition Editor* choisit d'utiliser tout l'espace qu'on n'a pas alloué à l'étape précédente.

Terminer le partitionnement

Une fois que l'on est satisfait par le partitionnement, revérifier une dernière fois que l'on est sur le bon périphérique (dans la boîte en haut à droite) puis aller dans le menu *Édition* → *Appliquer toutes les opérations*. Un avertissement s'affiche, prévenant que les données présentes sur le disque dur vont être perdues. Lorsqu'on a bien vérifié que l'on ne faisait pas de bêtise, on peut cliquer sur *Appliquer*.

Si, à la fin, *Partition Editor* se plaint de ne pas avoir les droits suffisants pour monter le volume, ce n'est pas bien grave.

Chiffrer la deuxième partition

Une fois la deuxième partition créée, *Partition Editor* affiche les deux partitions que contient désormais notre support de stockage, et il indique également le nom qu'elles portent, comme `sdx1`, `sdx2`, `sdx3`, *etc.* **Repérer le nom complet qui a été attribué à la nouvelle partition**, s'assurer que cette partition est démontée, et suivre les instructions pour chiffrer un disque dur **en utilisant ce nom-là** à la place de `LE_PÉRIPHÉRIQUE`.

plus bas

18.6 Chiffrer un disque dur

Ci-dessous, on parlera toujours de disque dur, sachant que ça vaut aussi bien pour un disque dur externe que pour une clé USB, sauf si on précise le contraire.

Attention, pour chiffrer le disque on va effacer toutes les données qu'il y a dessus !

Installer les paquets nécessaires

page 119

Pour chiffrer notre disque dur, on a besoin d'avoir installé les paquets `secure-delete`, `dosfstools` et `cryptsetup`.

Démonter le disque dur

Aller sur le bureau et cliquer sur l'icône du disque dur externe avec le bouton droit. Dans le menu contextuel qui apparaît, cliquer sur *Démonter le volume*. Il **ne** faut **pas** le débrancher.

Chiffrer le disque dur

Pour ça, on n'a pas trouvé de façon plus simple qu'utiliser un terminal.

On va donc ouvrir un *Terminal administrateur*. page 87

Il faut ensuite entrer, avec le nom trouvé tout à l'heure, qui ressemble à `sdx1`, à la place de `LE_PÉRIPHÉRIQUE` :

```
luksformat /dev/LE_PÉRIPHÉRIQUE
```

Puis presser la touche *Entrée*. `luksformat` prévient (en anglais) qu'il va supprimer toutes les données présentes sur `LE_PÉRIPHÉRIQUE` de façon définitive. Vérifier que l'on ne s'est pas trompé dans `LE_PÉRIPHÉRIQUE` puis taper `YES` (en majuscules, oui).

Pour la suite :

- Choisir une bonne phrase de passe. page 93
- Entrer la phrase de passe trois fois, en appuyant à chaque fois sur la touche *Entrée* à la fin (si l'on ne tape pas la même à chaque fois, il faudra lancer de nouveau la commande précédente).
- Une fois que `luksformat` a fini, on voit à nouveau l'invite du terminal : une ligne qui se termine par `#`, et dans laquelle on peut taper des commandes. Débrancher ensuite physiquement le disque externe de l'ordinateur.
- Garder le terminal ouvert, ce n'est pas tout à fait fini…

Monter le disque dur

Rebrancher le disque dur externe. Une fenêtre demandant la phrase de passe du disque doit s'afficher. L'entrer, et cliquer sur *Valider*.

Remplir le disque dur de données aléatoires

Pour finir, on va remplir l'espace vide du disque dur de données aléatoires. Cela permet de cacher l'endroit où vont se trouver les vraies données et complique la vie de la personne qui voudrait tenter de les déchiffrer.

Il faut tout d'abord trouver où le système a monté le disque dur. Pour cela, aller sur le bureau, et cliquer sur l'icône du disque dur avec le bouton droit. Dans le menu contextuel, sélectionner *Propriétés*, et aller dans l'onglet *Volume*. Là, noter ce qu'il est écrit après *Point de montage*. Nous appelerons cette valeur `POINT_DE_MONTAGE`. C'est l'endroit par lequel les programmes peuvent accéder au contenu déchiffré du disque dur.

Ensuite, dans le terminal, taper — avec le point de montage trouvé tout à l'heure à la place de `POINT_DE_MONTAGE` :

```
sfill -l -l -v POINT_DE_MONTAGE
```

… puis presser la touche *Entrée*.

Le processus dure quelques minutes à quelques heures, selon la taille du disque dur et sa vitesse (par exemple, 2 heures pour une clé USB de 4 Go). Une fois que l'invite de commande s'affiche, c'est fini.

Le disque dur chiffré est maintenant utilisable.

18.7 Utiliser un disque dur chiffré

Afin de permettre au système d'accéder aux données qui se trouvent sur un disque chiffré, il est heureusement nécessaire d'indiquer la phrase de passe. Une opération plus ou moins simple selon les environnements…

Avec GNOME

Sur un système GNU/Linux avec GNOME, comme *The (Amnesic) Incognito Live System* ou Debian, après le branchement d'un disque externe contenant des données chiffrées, une fenêtre apparaît pour demander la phrase de passe.

Pour fermer la partition chiffrée, il suffit de démonter le disque dur comme on le fait habituellement.

Sous GNU/Linux sans GNOME

Avec tous les systèmes GNU/Linux sur lesquels `cryptsetup` est installé, il est possible page 87 à partir d'un *Terminal administrateur* de rendre accessibles les données chiffrées du disque. Cependant, il faut pour cela le mot de passe super-utilisateur de l'ordinateur utilisé.

Pour accéder au disque

page 144 Une fois qu'on a trouvé le nom du périphérique en question, il faut taper la ligne :

```
#> cryptsetup luksOpen /dev/LE_PÉRIPHÉRIQUE disque_chiffre
```

Puis appuyer sur *Entrée*, et entrer la phrase de passe lorsque s'affiche :

```
Enter passphrase for /dev/LE_PÉRIPHÉRIQUE:
```

Cela crée une image déchiffrée du contenu du disque dur dans `/dev/mapper/disque_chiffre`. À noter que `disque_chiffre` est un nom choisi arbitrairement, on aurait aussi bien pu choisir `machin` à la place.

Si le contenu du disque dur n'apparaît toujours pas tout seul, taper :

```
#> mount /dev/mapper/disque_chiffre /mnt
```

Cette opération permet d'accéder aux fichiers de `/dev/mapper/disque_chiffre` en allant dans `/mnt` et s'appelle le « montage ».

Pour fermer le disque dur

Si le contenu du disque dur est apparu tout seul après la première étape ci-dessus, le démonter comme de coutume. Sinon, dans un *Terminal administrateur*, taper :

```
#> umount /mnt
```

Dans tous les cas, taper ensuite dans le terminal :

```
cryptsetup luksClose disque_chiffre
```

Et appuyer sur *Entrée*. Vérifier qu'aucun message d'erreur ne s'affiche. Cette commande permet en effet de fermer réellement la partition chiffrée et de la rendre inaccessible à qui n'a pas la phrase de passe.

Avec d'autres systèmes

Il est possible d'accéder à la partition chiffrée du disque dur sous Windows grâce à FreeOTFE [2]. Pour Mac OS X, rien n'est disponible à l'heure où nous écrivons ces lignes. [3]

Cependant, en faisant cela, on donne sa phrase de passe à une machine qui utilise des logiciels propriétaires, en qui il n'y a aucune raison d'avoir confiance. page 29

Alors le mieux à faire, pour mettre sur son disque dur des données auxquelles on veut accéder sur des ordinateurs en lesquels on n'a pas confiance, c'est de mettre une deuxième partition, non chiffrée, sur son disque dur, comme expliqué dans la section précédente. page 145

2. Disponible sur le site de FreeOFTE [http://www.freeotfe.org/].
3. On peut toutefois jeter un œil au site OSXCrypt.org [http://www.osxcrypt.org/] qui vise à écrire les logiciels permettant d'accéder à des partitions LUKS depuis Mac OS X.

Chapitre 19

Sauvegarder des données

Réaliser des sauvegardes est une opération relativement simple dans son principe : faire une copie des fichiers qu'on ne voudrait pas perdre.

Il existe des logiciels spécialement prévus pour réaliser des sauvegardes, mais le gestionnaire de fichiers suffit amplement dans les cas simples. Il suffit de copier les fichiers comme on le fait habituellement vers un support de stockage externe (clé USB ou disque dur) que l'on aura préalablement chiffré.

page 143

Réaliser des sauvegardes est surtout un travail de discipline : ne pas oublier de fichiers dans l'opération, et ne pas oublier de *faire* les sauvegardes.

Pour cela, deux conseils qui peuvent porter leurs fruits :

- avoir quelque part une liste des fichiers et dossiers à sauvegarder ;
- réaliser un petit calendrier des jours ou semaines où l'on fera ses sauvegardes, avec des cases à cocher lorsque c'est fait.

Il est bon également de stocker les sauvegardes à un endroit différent, pour éviter qu'elles soient détruites ou perdues en même temps que les données originelles.

Chapitre 20

Créer un compte « utilisateur » sur un système Debian

Le but de cette recette est de créer un nouveau compte « utilisateur », et de l'isoler un peu des autres.

Créer le nouveau compte

Ouvrir *Système* → *Administration* → *Utilisateurs et groupes*. Le système demande le mot de passe d'administration. Cliquer ensuite sur *Ajouter un utilisateur*.

Dans la boîte de dialogue qui s'ouvre, remplir le nom (*login*) du nouveau compte dans *Utilisateur*.

⚠ **Attention**, ce nom restera dans de nombreuses traces : il s'agit donc de ne pas choisir un nom trop évocateur. Il faut aussi choisir un *Mot de passe utilisateur* et le *Confirmer*.

Dans l'onglet *Privilèges utilisateurs*, cocher notamment *Accéder à des périphériques de stockage externes automatiquement*, *Connect to wireless and ethernet networks*, *Utiliser des lecteurs de CD-ROM* et *Utiliser des périphériques audio*.

C'est alors le moment de *Valider* puis de *Fermer* les *Réglages utilisateurs*.

Ouvrir une session avec le nouveau compte

Ouvrir *Applications* → *Outils système* → *Nouvelle connexion*. Entrer le nom du compte (*login*) et le mot de passe choisi précédemment.

Rendre les nouveaux fichiers illisibles pour les autres comptes

Pour cela, on va éditer un fichier de configuration dans l'éditeur de texte.

Ouvrir *Applications* → *Accessoires* → *Éditeur de texte*. Choisir alors *Fichier* → *Ouvrir…* Dans la boîte de dialogue d'ouverture de fichiers, cliquer avec le bouton droit sur la liste des dossiers. Dans le menu contextuel, cocher *Afficher les fichiers cachés*. Sélectionner alors le fichier *.profile* et l'*Ouvrir*.

Dans ce fichier, chercher la ligne :

```
# umask 022
```

Et la remplacer (bien penser à enlever le #) par :

```
umask 077
```

Fermer alors le fichier en enregistrant les modifications.

Interdire la lecture du dossier personnel aux autres comptes

Sur le bureau, sélectionner le *Dossier personnel*, et cliquer dessus avec le bouton droit. Dans le menu contextuel, choisir *Propriétés*. Aller alors dans l'onglet *Permissions*. Dans la section pour le *Groupe* (la seconde), choisir *Accès au dossier* : *Aucun*. Faire de même dans la section *Autres*.

Cliquer alors sur *Appliquer les permissions aux fichiers inclus* et *Fermer*.

Fermer la session

Les modifications ne prennent effet qu'à l'ouverture d'une session. Avant de travailler réellement, il faut donc fermer la session avec *Système* → *Fermer la session*, quitte à en ouvrir une nouvelle immédiatement après.

Chapitre 21

Supprimer un compte « utilisateur » sur un système Debian

L'objectif de cette recette est de supprimer un compte « utilisateur » de l'ordinateur et d'effacer un certain nombre de ses traces.

Fermer les éventuelles sessions du compte

Si une session est ouverte avec le compte à supprimer, fermer sa session.

Cela ne suffit en général pas à quitter tous ses programmes. On va donc le faire à la main : ouvrir un terminal, cliquant sur le menu *Applications*, puis *Accessoires* et enfin *Terminal administrateur*. Taper alors, en remplaçant `LOGIN` par le login du compte à effacer :

```
killall -u LOGIN
```

Garder le terminal ouvert, on en aura de nouveau besoin.

Supprimer les fichiers personnels du compte

On effectue une suppression « normale » (sans effacer le contenu), car on va de toute façon recouvrir l'espace libre juste après.

Dans le terminal, taper, en remplaçant `LOGIN` par le nom du compte à supprimer, puis appuyer sur la touche *Entrée* :

```
find / -user LOGIN -delete
```

C'est un peu long…

Supprimer le compte

Depuis une session ayant le droit d'administrer le système, ouvrir *Système* → *Administration* → *Utilisateurs et groupes*. Sélectionner le compte à supprimer, et cliquer sur *Supprimer*.

Une boîte de dialogue informe que le dossier personnel du compte ne sera pas supprimé. Qu'à cela ne tienne, on l'a déjà supprimé. Après avoir vérifié que l'on supprime bien le bon compte, confirmer.

Supprimer le groupe du compte

Toujours dans les *Réglages utilisateurs*, cliquer sur *Gérer les groupes*. Sélectionner le groupe du nom du compte à supprimer, et choisir *Supprimer*. Après vérification, confirmer malgré l'avertissement.

Il est alors possible de fermer les boîtes de dialogue.

Supprimer les dernières traces existantes

Sur les systèmes GNU/Linux, il existe un programme qui indexe les noms des fichiers et permet de les retrouver facilement : *locate*. Il faut mettre à jour sa base de données de noms de fichiers pour lui dire d'oublier les fichiers que l'on vient de supprimer.

Pour ce faire, il faut lancer, dans notre *Terminal administrateur* :

```
updatedb
```

Recouvrir les traces de fichiers effacés

On veut effacer les traces de fichiers effacés de divers endroits où le compte avait le droit d'en écrire :

- `/home` : le dossier où se trouvent les dossiers personnels ;
- `/tmp` et `/var/tmp` : les dossiers temporaires ;
- `/var` : un dossier de données des applications ;
- `/var/log` : le dossier de journaux système.

Cependant, si tous ces fichiers se trouvent sur la même partition, on ne veut pas effectuer le recouvrement de l'espace libre plusieurs fois : ce serait très long.

Installer le logiciel nécessaire

page 119

Si le paquet `secure-delete` n'est pas encore installé, le faire.

Lancer le recouvrement

La commande suivante (assez compliquée, on avoue) efface juste ce qu'il faut.

Aller dans le terminal ouvert précédemment, et taper :

```
df -P /home /tmp /var /var/log /var/tmp \
| tail -n +2 | awk '{ print $6 }' | sort -u \
| xargs --max-args=1 sfill -l -v
```

Puis patienter, surtout si l'on a un gros disque. Lorsque c'est terminé, il est possible de fermer le terminal.

Des traces resteront

Une fois tout ceci fait, les données devraient avoir été supprimées. Mais à vrai dire, les traces que peut laisser la présence d'un compte sur un système GNU/Linux sont assez nombreuses et assez imprévisibles, car dépendantes aussi des programmes qui auront été installés ou utilisés. Un programme pourra par exemple avoir été prévu

page 22 pour sauvegarder lui même quelques fichiers où seront écrits l'emplacement du dossier personnel (qui contient donc le nom du compte), ou bien écrire dans les méta-données de ses fichiers en format illisible pour un humain le *login* qui aura créé ces documents. Trouver toutes ces traces de manière exhaustive relève du travail de longue haleine, voire de la tâche insurmontable, et c'est dans un cas comme celui-ci que l'on touche les limites de la stratégie de la liste noire. page 53

Malgrès tout, les différents nettoyages fait précédemment dans cet outil devraient en avoir effacé une bonne partie, et si l'on a le temps et la nécessité de s'atteler à cette recherche, il y a bien quelques outils qui peuvent aider.

Pour obtenir la liste de tous les fichiers et dossiers dont le nom contient « pierrine » (le *login* du compte à supprimer), on peut taper la commande suivante :

```
find / -mount -name '*pierrine*'
```

Et pour obtenir tous les fichiers qui contiennent le mot « pierrine » et qui se trouvent dans `/var` ou un de ces sous-dossiers, on utilisera :

```
rgrep /var pierrine
```

Il faut cependant s'attendre à un certain nombre de faux positifs pour cette dernière commande.

Dans la plupart des cas, s'il est impératif d'effacer toute traces de l'existence d'un compte, la réinstallation d'un système crypté restera la solution la plus simple et la plus rapide. page 105

Chapitre 22

Partager un secret

Parfois, on souhaite être plusieurs à partager un secret, sans pour autant que chaque personne ne dispose de la totalité du secret.

Cela tombe bien, plusieurs techniques cryptographiques [1] ont été inventées pour cela. Elles permettent toutes, mais avec des calculs mathématiques un peu différents, de découper un secret en plusieurs morceaux, que l'on pourra reconstituer en en réunissant quelques-uns.

22.1 Partager une phrase de passe

L'usage le plus pratique est de partager comme secret la phrase de passe d'un support chiffré. page 143

Cette étape doit idéalement être faite à partir d'un système *live* afin de ne pas laisser des traces du secret que l'on va partager. page 101

Installer le paquet nécessaire

Pour réaliser le partage du secret, on utilisera le programme `ssss-split`. Pour en disposer, il est nécessaire d'installer le paquet Debian *ssss*. page 119

Les outils contenus dans le paquet *ssss* sont à utiliser en ligne de commande. Toutes les opérations devront donc être effectuées dans un *Terminal*, sans les pouvoirs d'administration. page 87

Générer une phrase aléatoire

Dans notre cas, personne ne doit pouvoir ni se souvenir ni deviner la phrase de passe qui sera utilisée pour le chiffrement. On va donc générer une phrase de passe complètement aléatoire :

```
head -c 32 /dev/random | base64
```

L'ordinateur va répondre quelque chose comme :

```
7rZwOOu+8v1stea98OuyU1efwNzHaKX9CuZ/TK0bRWY=
```

Sélectionner cette ligne à l'aide de la souris et la copier dans le presse-papiers (via le menu *Édition* → *Copier*).

1. Pour plus de détails, voir l'article de Wikipédia sur les secrets répartis [https://secure.wikimedia.org/wikipedia/fr/wiki/Secret_réparti].

Découper le secret

Avant de découper le secret, il faut décider en combien de morceaux il sera découpé, et combien de morceaux seront nécessaires pour le reconstituer.

Ensuite, toujours à l'aide de notre terminal, il faut utiliser `ssss-split` de la façon suivante :

```
ssss-split -t NOMBRE_DE_MORCEAUX_NECESSAIRES -n NOMBRE_DE_MORCEAUX_TOTAL
```

Le message `WARNING: couldn't get memory lock` peut être ignoré sans problème si on utilise bien un système *live*.

Lorsqu'il demande le secret, on peut coller le contenu du presse-papier, à l'aide du menu *Édition* → *Coller*. Appuyer ensuite sur la touche *Entrée* pour valider la commande.

Chaque personne partageant le secret devra conserver l'une des lignes affichées ensuite. Cela dans leur **intégralité**, en prenant également bien en note le premier chiffre suivi du tiret.

Voici un exemple avec la clé aléatoire générée précédemment, partagée entre 6 personnes et qui nécessitera que 3 d'entre elles se réunissent pour la retrouver :

```
$ ssss-split -t 3 -n 6
Generating shares using a (3,6) scheme with dynamic security level.
Enter the secret, at most 128 ASCII characters: Using a 352 bit security level.
1-b8d576a1a8091760b18f125e12bb6f2b1f2dd9d93f7072ec69b129b27bb8e97536ea85c7f6dcee7b43
↪    99ea49
2-af83f0af05fc207e3b466caef30ec4d39c060800371feab93594350b7699a8db9594bfc71ed9cd2bf3
↪    14b738
3-4718cb58873dab22d24e526931b061a6ac331613d8fe79b2172213fa767caa57d29a6243ec0e6cf77b
↪    6cbb64
4-143a1efcde7f4f5658415a150fcac6da04f697ebfeb9427b59dca57b50ec755510b0e57ccc594e6b1a
↪    1eeb04
5-fca1250b5cbec40ab14964d2cd7463af34c389f81158d1707b6a838a500977d957be38f83e8eefb792
↪    66e74a
6-ebf7a305f14bf3143b801a222cc1c857b7e8582119374925274f9f335d283677f4c002f8d68bcce722
↪    ebba1f
```

Créer le support chiffré

page 143

On pourra ensuite créer le support chiffré. Au moment d'indiquer la phrase de passe, on pourra copier le contenu du presse-papier, comme précédemment, ou alors la retranscrire en l'ayant sous les yeux.

22.2 Reconstituer la phrase de passe

Afin de reconstituer la phrase de passe, il est nécessaire de disposer d'au moins autant de morceaux que le nombre minimal décidé lors du découpage.

page 101

Cette étape doit idéalement être faite à partir d'un système *live* afin de ne pas laisser de traces du secret partagé.

Installer les paquets nécessaires

page 119

Comme précédemment, on aura besoin d'avoir installé le paquet *ssss* et d'avoir ouvert un terminal.

Recombiner le secret

Afin de recombiner le secret, on utilisera le programme `ssss-combine`. Il est nécessaire de lui indiquer le nombre de morceaux qu'on a à notre disposition :

```
ssss-combine -t NOMBRE_DE_MORCEAUX_A_DISPOSITION
```

Le programme demande ensuite de saisir les morceaux à notre disposition. Il faut taper *Entrée* après chacun d'entre eux. Si tout se passe bien, le programme affichera ensuite la phrase de passe complète.

Pour reprendre l'exemple précédent, cela donne :

```
$ ssss-combine -t 3
Enter 3 shares separated by newlines:
Share [1/3]: 4-143a1efcde7f4f5658415a150fcac6da04f697ebfeb9427b59dca57b50ec755510b0e5
 ↪    7ccc594e6b1a1eeb04
Share [2/3]: 2-af83f0af05fc207e3b466caef30ec4d39c060800371feab93594350b7699a8db9594bf
 ↪    c71ed9cd2bf314b738
Share [3/3]: 6-ebf7a305f14bf3143b801a222cc1c857b7e8582119374925274f9f335d283677f4c002
 ↪    f8d68bcce722ebba1f
Resulting secret: 7rZwOOu+8v1stea98OuyU1efwNzHaKX9CuZ/TK0bRWY=
```

Attention, si un des morceaux a mal été tapé, l'erreur qui s'affiche n'est pas forcément très explicite :

```
$ ssss-combine -t 3
Enter 3 shares separated by newlines:
Share [1/3]: 4-143a1efcde7f4f5658415a150fcac6da04f697ebfeb9427b59dca57b50ec755510b0e5
 ↪    7ccc594e6b1a1eeb04
Share [2/3]: 2-af83f0af05fc207e3b466caef30ec4d39c060800371feab93594350b7699a8db9594bf
 ↪    c71ed9cd2bf31ab738
Share [3/3]: 6-ebf7a305f14bf3143b801a222cc1c857b7e8582119374925274f9f335d283677f4c002
 ↪    f8d68bcce722ebba1f
Resulting secret: ......L.fm.....6 _....v..w.a....[....zS.....
WARNING: binary data detected, use -x mode instead.
```

Ouvrir le support chiffré

Une fois la phrase de passe obtenue, on peut utiliser un copier/coller afin de déverrouiller le support chiffré, ou alors la retranscrire en l'ayant sous les yeux.

Chapitre 23

Utiliser les sommes de contrôle

Dans la première partie, on a évoqué les *sommes de contrôle*, des « nombres » qui permettent de vérifier l'intégrité d'un fichier (ou de toutes autres données). Le principe est qu'il est quasiment impossible d'avoir une somme de contrôle identique pour deux fichiers différents. Si Alice dit à Bob dans une lettre que le programme qu'il peut télécharger sur son site a pour somme de contrôle SHA256 `171a0233a4112858db23621dd5ffa31d269cbdb4e75bc206ada58ddab444651f` et que le fichier qu'il a reçu a la même somme de contrôle, il est quasiment sûr que personne n'a falsifié le programme en chemin, et il peut exécuter le programme sans trop de craintes.

page 41

Il existe plusieurs algorithmes pour faire des sommes de contrôles. Parmi eux :

- MD5 n'est plus sûr de nos jours et est à proscrire ;
- SHA1 est très utilisé, mais est en voie d'être cassé. Il faut l'abandonner ;
- SHA224, SHA256, SHA384 et SHA512 sont pour l'instant toujours sûrs. Nous allons utiliser SHA256, mais les mêmes méthodes fonctionnent avec les autres algorithmes.

23.1 Obtenir la somme de contrôle d'un fichier

Que l'on souhaite vérifier l'intégrité d'un fichier, ou permettre à son correspondant de le faire, il faut calculer la somme de contrôle de ce fichier.

Il est d'une part possible de configurer Nautilus, le navigateur de fichiers du bureau GNOME, pour effectuer des sommes de contrôle.

page suivante

D'autre part, si l'on est à l'aise avec l'utilisation d'un terminal, on obtient le SHA256 en exécutant la commande :

page 87

```
sha256sum NOM_DU_FICHIER
```

Pour obtenir le SHA1, ce sera :

```
sha1sum NOM_DU_FICHIER
```

Et ainsi de suite pour MD5 (`md5sum`) ou les autres SHA (`sha224sum`, `sha384sum` par exemple).

23.2 Vérifier l'intégrité d'un fichier

- Il faut obtenir la somme de contrôle du fichier original par un moyen sûr, autre que celui par lequel on a reçu le fichier. Par exemple, si l'on a téléchargé le fichier, on peut avoir reçu sa somme de contrôle dans une lettre, ou par téléphone — le mieux étant bien sûr de vive voix.
- Grâce à l'une des méthodes ci-dessus, obtenir la somme de contrôle de sa copie du fichier. Prendre garde à utiliser le même algorithme que celui qui a été utilisé par son correspondant. Si l'on utilise SHA1 et qu'il utilise SHA256, on n'aura bien sûr pas la même somme de contrôle. Si notre correspondant nous propose plusieurs sommes de contrôle, préférer l'algorithme le plus dur à casser (voir ci-dessus).
- Vérifier que les deux sommes de contrôle sont les mêmes — c'est un peu long et fastidieux, mais c'est souvent plus simple à deux, ou en les collant l'une en-dessous de l'autre dans un fichier texte.

23.3 Permettre à d'autres de vérifier l'intégrité d'un fichier

- Grâce à l'une des méthodes ci-dessus, obtenir la somme de contrôle de sa copie du fichier. Préférer l'algorithme le plus dur à casser (voir ci-dessus).
- Faire parvenir cette somme de contrôle à son correspondant par un moyen sûr, autre que celui par lequel on envoie le fichier. Par exemple, si le fichier est envoyé par email, on peut envoyer sa somme de contrôle dans une lettre, ou par téléphone — le mieux étant bien sûr de vive voix.

23.4 Faire une somme de contrôle en mode graphique

Pour faire une somme de contrôle depuis le bureau graphique GNOME, on va ajouter un tout petit programme (un *script*) au navigateur de fichiers de GNOME (qui s'appelle Nautilus).

Télécharger ou écrire le script

Afin d'ajouter ce petit programme, deux possibilités : le télécharger si on a accès à Internet ou le recopier (en se relisant plusieurs fois).

Première option : télécharger le script

Télécharger le script `Calculer_une_somme_de_controle` à partir de l'adresse : https://guide.boum.org/tomes/1_hors_connexions/3_outils/12_utiliser_les_sommes_de_controle/hash_dans_Nautilus/Calculer_une_somme_de_controle

Deuxième option : écrire le script

- Ouvrir l'*Éditeur de texte* qui se trouve dans le menu *Applications* puis *Accessoires*.
- Écrire, sur la page blanche qui est apparue :

```
#!/bin/bash

ALGO=$(zenity --list --title="Calculer une somme de contrôle" \
    --text="Choisir le type de somme de contrôle" \
    --width=400 --height=300 --radiolist \
    --column="" --column="Algorithme" \
    False MD5 \
    False SHA1 \
```

```
    False SHA224 \
    True  SHA256 \
    False SHA384 \
    False SHA512) || exit

COMMAND="$(echo "${ALGO}" | tr A-Z a-z)sum"

RESULT=$(${COMMAND} "$@" | sed -e 's, \+,\n,' |
    tee >(zenity --progress --auto-kill --auto-close --pulsate))

echo "$RESULT" | zenity --list --title="${ALGO}" \
    --text="Sommes de contrôle ${ALGO} des fichiers sélectionnés" \
    --editable --width=800 --height=300 \
    --column="${ALGO}" --column="Fichier" \
    --separator="$(printf '\n')"
```

- Enregistrer le fichier en cliquant dans le menu *Fichier* sur *Enregistrer*. L'appeler `Calculer_une_somme_de_controle` et le ranger sur le bureau (*Desktop*).
- Quitter l'*Éditeur de texte*.

Copier le script là où Nautilus le cherche

- Aller sur le bureau, sélectionner le fichier `Calculer_une_somme_de_controle`.
- Cliquer dessus avec le bouton droit de la souris. Dans le menu contextuel qui apparaît, cliquer sur *Couper*.
- Ouvrir le *Navigateur de fichiers*, qui se trouve dans le menu *Applications* → *Outils systèmes*.
- Dans le menu *Aller à* cliquer sur *Emplacement…*, puis taper `~/.gnome2/nautilus-scripts/` et appuyer sur la touche *Entrée*.
- Coller le fichier en cliquant dans le menu *Édition* sur *Coller*.

Rendre le script exécutable

- Sélectionner le fichier `Calculer_une_somme_de_controle`.
- Cliquer dessus avec le bouton droit de la souris. Dans le menu contextuel qui apparaît, cliquer sur *Propriétés*.
- Dans la boîte de dialogue qui apparaît, aller dans l'onglet *Permissions*, cocher la case *Exécution*.
- Fermer la boîte en cliquant sur *Fermer*.

Vérifier

- Dans le menu contextuel du navigateur de fichiers, un sous-menu *Scripts* contenant la commande `Calculer_une_somme_de_controle` devrait apparaître.

Utiliser le script

- Sélectionner les fichiers dont la somme de contrôle doit être calculée.
- Cliquer dessus avec le bouton droit de la souris. Dans le menu contextuel qui apparaît, cliquer sur *Scripts*, puis sur `Calculer_une_somme_de_controle`.
- Choisir l'algorithme souhaité et valider.
- Les sommes de contrôle des fichiers sélectionnés s'affichent.

Chapitre 24

Installer et utiliser un système virtualisé

Cet ensemble de recettes traite de l'utilisation d'un système d'exploitation virtuel à l'intérieur d'un système GNU/Linux. Elles sont utilisées par le cas d'usage qui parle de travailler sur un document sensible sous Windows. page 72

24.1 Installer VirtualBox

Principe

L'objectif de cette recette est d'installer VirtualBox, un logiciel qui permet de faire tourner un système d'exploitation (appelé invité) à l'intérieur d'un autre (appelé hôte) : on appelle cela de la *virtualisation*. Cette technologie, ainsi qu'une politique de sécurité l'utilisant, est décrite plus avant dans le cas d'usage « Travailler sur un document sensible sous Windows ».

page 72

Ajouter les *backports* à ses dépôts Debian

Afin de pouvoir installer une version plus facilement utilisable de VirtualBox, il est nécessaire d'ajouter les *backports* aux sources de paquets pour son système.

page 121

Installer VirtualBox à partir des *backports*

L'étape suivante est donc d'installer les versions lenny-backports des paquets `virtualbox-ose-dkms` puis `virtualbox-ose` (en mars 2010, il s'agissait de la version 3.1.4).

page 119

Attention, l'ordre indiqué est important. Il faut d'abord demander l'installation de celle de `virtualbox-ose-dkms` **puis** celle de `virtualbox-ose` (via *Paquet* → *Forcer la version…*) ; sinon, il sera nécessaire de redémarrer Debian avant de pouvoir utiliser VirtualBox. Il est également nécessaire de mettre en place l'*APT pinning* sur les paquets `virtualbox-ose`, `virtualbox-ose-qt` et `virtualbox-ose-dkms` pour s'assurer que VirtualBox soit maintenu à jour.

page 125

Vérifier l'installation

Lancer VirtualBox à partir du menu *Applications* → *Outils système* → *VirtualBox OSE*. Si l'installation des paquets s'est bien passée, une fenêtre s'ouvre et nous souhaite la bienvenue dans VirtualBox. Refermons-la, car nous avons encore quelques préparatifs à faire avant de nous servir de ce logiciel.

Ajouter un lien vers le dossier des disques virtuels

On aura besoin plus tard d'accéder au dossier dans lequel VirtualBox range les disques virtuels qu'il utilise. Cependant, il est un peu fastidieux à trouver. On va donc créer une bonne fois pour toutes un lien vers ce dossier.

Ouvrir le dossier personnel à partir du menu *Raccourcis* → *Dossier personnel*.

Le dossier où VirtualBox stocke ses images de disque est caché. Il faut donc afficher les fichiers cachés à partir du menu *Affichage* → *Afficher les fichiers cachés*.

Trouver le dossier `.Virtualbox` et aller dedans (en double cliquant dessus).

À l'intérieur de ce dossier, les disques virtuels sont rangés dans le dossier `HardDisks`. Il n'existe pas encore, alors il faut commencer par le créer : clic droit → *Créer un dossier*, et taper `HardDisks`, en faisant attention aux majuscules et minuscules, comme nom du dossier.

Puis, pour y accéder simplement par la suite, on va ajouter un lien vers ce dossier.

Cliquer dessus avec le bouton droit de la souris et choisir *Créer un lien* dans le menu contextuel qui apparaît.

Une icône *Lien vers HardDisks* apparaît. La sélectionner, et cliquer dessus avec le bouton droit de la souris. Dans le menu contextuel, choisir *Renommer…* et donner un nom un peu plus clair, par exemple *Disques virtuels de VirtualBox*.

Déplacer alors ce lien vers son *Dossier personnel* (accessible depuis le menu *Raccourcis*).

On peut alors à nouveau cacher les fichiers cachés en décochant *Affichage* → *Afficher les fichiers cachés*, puis fermer les dossiers qu'on a ouverts.

Créer un dossier pour sauvegarder les images propres

Comme expliqué dans le cas d'usage, on aura plus tard envie de sauvegarder des images de systèmes propres. Créons dès maintenant un dossier pour cela, par exemple en ajoutant dans le *Dossier personnel* (accessible depuis le menu *Raccourcis*) un dossier *Disques virtuels propres*.

page 72

24.2 Installer un Windows virtualisé

Avant tout chose, se munir d'un CD d'installation de la version de Windows appropriée, et le charger dans le lecteur CD/DVD. Refermer, ou ignorer, la fenêtre qui s'ouvre alors automatiquement.

Préparer l'installation sur VirtualBox

Depuis le bureau, aller dans *Applications* → *Outils système* → *VirtualBox OSE*.

Le programme démarre. Cliquer sur *Nouveau* et suivre l'assistant :

- Choisir un nom pour la machine virtuelle.
- Choisir le type de système correspondant parmi les versions de Windows proposées.
- Valider la taille de mémoire vive dédiée à la machine virtuelle.
- Créer un disque dur virtuel pour accueillir le système d'exploitation virtuel :
  - cliquer sur *Nouveau* et sélectionner *Image dynamique* (l'image disque s'agrandira au besoin, jusqu'à atteindre au maximum la taille indiquée) ;
  - donner un nom au fichier image disque (on peut aussi choisir son emplacement en cliquant sur le petit dossier à droite de cette ligne, mais c'est bien de le laisser à l'endroit suggéré) ;
  - choisir la taille de l'image virtuelle : sachant qu'on veut accueillir tout un Windows, elle doit être conséquente ! 10 Go, c'est bien si on a assez de place ; en cas de petit disque dur, essayer moins…
  - terminer : le logiciel a créé et sélectionné un disque dur virtuel.
- Cliquer sur suivant et terminer.

On peut maintenant sélectionner la machine Windows nouvellement créée dans le menu principal de VirtualBox, et il reste à installer le système…

⚠ **Mais avant tout**, on va cliquer sur le bouton *Préférences* : ce menu permettra par la suite de la configurer. Pour l'instant on va juste lui dire deux choses :

1. Il faut couper l'accès au réseau (pour des raisons de sécurité déjà mentionnées) :
   - aller dans le sous-menu *Réseau* ;
   - décocher la case *Activer la carte réseau* dans tous les onglets où elle est déjà cochée par défaut (généralement, dans un seul : le premier).
2. Il faut lancer le système sur le CD d'installation de Windows qu'on a mis dans le lecteur CD/DVD :
   - aller dans le sous-menu CD/DVD ;
   - cocher la case *Insérer un CD/DVD-ROM* ;
   - sélectionner *Connecter le lecteur hôte*, ce qui correpond à votre lecteur CD/DVD habituel.

Cliquer ensuite sur *OK* pour enregistrer les paramètres.

Lancer la machine virtuelle

Double-cliquer sur le système virtuel *Windows* dans le menu VirtualBox, et le système virtuel démarre… c'est le moment de découvrir l'utilisation de la machine virtuelle.

Lorsqu'elle est lancée, la machine virtuelle fonctionne dans une fenêtre qui permet de gérer son utilisation :

- en haut à gauche : un menu contenant *machine*, *périphériques*, *aide* ;
- en bas à droite : des icônes indiquant comment la machine virtuelle utilise le matériel. On peut par exemple vérifier, en passant la souris dessus, que toutes les connexions réseau sont désactivées.

Au premier clic dans la fenêtre, le logiciel explique qu'il va capturer la souris ; à la première touche tapée, il explique qu'il capture le clavier. Il faut bien prendre en compte ce qu'il indique, c'est ce qui permet de sortir de la machine virtuelle !

Enfin, tout ça est expliqué par le logiciel. Il nous reste donc à installer le Windows virtuel.

Installer Windows

Le système virtuel démarre sur le lecteur CD/DVD qu'on lui a indiqué et commence l'installation.

On ne rentrera pas dans les détails du processus. On peut toutefois préciser :

- Au moment de formater la partition, mieux vaut choisir *Formater avec NTFS (rapide)*.
- Ne mettez pas d'informations personnelles lorsque le *nom* et l'*organisation* sont demandés. Mettre un simple point (« . ») dans les cases permet, la plupart du temps, de continuer l'installation.
- Lors de la configuration du réseau, un message d'erreur peut être affiché. C'est bon signe : nous avons désactivé le réseau de la machine virtuelle.

Démarrer sur le système invité

Une fois l'installation terminée, aller dans la fenêtre de VirtualBox, et cliquer dans la liste de gauche sur la macine virtuelle qu'on vient d'installer. Cliquer sur l'icône *Préférences*. Dans la liste de gauche, choisir *Supports* et enlever le CD d'installation de Windows. Fermer alors les préférences.

Démarrer alors la machine virtuelle en cliquant sur *Lancer*.

Installer les logiciels bonus pour système invité

Dans la fenêtre qui accueille Windows, ouvrir le menu *Périphérique* qui propose *Installer les additions invité*. Si ce n'a pas été fait auparavant, VirtualBox proposera de télécharger l'image ISO qui les contient. Un nouveau CD-ROM sera ensuite ajouté à l'environnement de Windows, et cela devrait lancer le programme d'installation. Il suffit ensuite d'accepter les choix par défaut pour installer les « Additions invité ».

Une nouvelle icône à l'aspect de cube transparent sera alors apparue en bas à droite du bureau Windows. Elle signifie que les « Additions » on été installées.

Éteindre le Windows virtuel. L'installation du Windows virtuel est maintenant terminée. On peut retourner au cas d'usage. page 72

24.3 Sauvegarder une image de disque virtuel propre

page 72 Comme indiquer dans la méthode permettant de travailler sur un document sensible sous Windows, on peut avoir besoin de sauvegarder (ou congeler) l'image disque d'une machine virtuelle.

Éteindre la machine virtuelle

Si la machine virtuelle propre, qui doit être sauvegardée, est en cours d'utilisation, il faut commencer par l'éteindre (par exemple via le menu *Machine* → *Fermer…* → *Envoyer le signal d'extinction* de VirtualBox).

Ouvrir le dossier des disques virtuels de VirtualBox

Dans le *Dossier personnel*, ouvrir le raccourci *Disques virtuels pour VirtualBox* créé précédemment.

Effectuer la sauvegarde

- Sélectionner le disque virtuel dont le nom correspond à celui de la machine virtuelle, par exemple `Windows 2000.vdi`.
- Dans le menu *Édition* choisir *Copier*.
- Aller dans le dossier de sauvegarde des images propres. Si l'on a suivi les conseils de la recette « installer VirtualBox » il s'agit du dossier *Disques virtuels propres* du *Dossier personnel* du compte utilisé.
page 168
- Dans le menu *Édition* choisir *Coller* pour obtenir une copie du fichier.
- Sélectionner la copie, et la renommer à partir du menu *Édition* → *Renommer…*. Entrer un nouveau nom, par exemple `Sauvegarde propre de Windows 2000.vdi`.

Effacer la machine virtuelle

cf. ci-contre On va va plus se servir de cette machine propre. C'est donc le moment de suivre la recette sur l'effacement d'une machine virtuelle.

24.4 Effacer « pour de vrai » une machine virtuelle

Cette recette vise à effacer proprement une machine virtuelle.

Supprimer la machine virtuelle de VirtualBox

Ouvrir la fenêtre principale de VirtualBox, accessible depuis le menu *Applications* → *Outils système*.

Sélectionner la machine virtuelle à effacer.

Dans le menu *Machine* choisir *Supprimer*, puis confirmer la suppression.

Effacer le disque dur virtuel et son contenu

Ouvrir le dossier des disques virtuels de VirtualBox.

Utiliser l'outil effacer des fichiers avec leur contenu pour effacer le disque virtuel de la machine virtuelle en question. page 129

Prévenir VirtualBox que le disque virtuel n'existe plus

Dans le menu *Fichier* de VirtualBox, ouvrir le *Gestionnaire de supports virtuels*. À ce moment-là, VirtualBox se plaint de l'absence du fichier qu'on vient d'effacer « pour de vrai », ce qui n'est pas franchement étonnant. Ignorer ces récriminations, puis, dans l'onglet *Disques durs*, sélectionner la ligne correspondant au fichier effacé (précédée d'un panneau jaune quelque peu alarmiste), et cliquer sur le bouton *Enlever*.

Le gestionnaire de supports virtuels, ainsi que la fenêtre de VirtualBox, peuvent maintenant être refermés.

24.5 Créer une nouvelle machine virtuelle à partir d'une image propre

page 172
page 72

L'objectif de cette recette est de « décongeler » une image de disque virtuel propre préalablement sauvegardée, afin de l'utiliser pour un nouveau projet, comme le recommande la méthode préconisée pour travailler sur un document sensible sous Windows.

Choix du nom

Il faudra choisir un nom pour cette nouvelle machine virtuelle et les fichiers qui lui correspondent. Ces fichiers étant situés sur le système hôte, ce nom laissera quasi inévitablement des traces dessus, même une fois la machine virtuelle supprimée. Il s'agit donc de choisir ce nom en connaissance de cause.

Copier l'image de disque virtuel

On ne peut pas simplement copier le fichier congelé, car VirtualBox se plaindrait qu'on a deux disques virtuels identiques. Il existe cependant une commande pour recopier un disque virtuel, mais elle n'est accessible que depuis la ligne de commande.

Commençons donc par ouvrir un terminal (*Applications* → *Accessoires* → *Terminal*).

Ensuite, recopions l'image précédemment décongelée avec la commande :

```
VBoxManage clonehd SAUVEGARDE NOUVEAU_DISQUE
```

dans laquelle il faudra remplacer `SAUVEGARDE` par le chemin d'accès à la sauvegarde du disque virtuel, et `NOUVEAU_DISQUE` par le chemin du nouveau disque. Nous allons à présent voir comment construire cette ligne de commande.

Attention Si l'on souhaite taper les noms de fichiers à la main, il faut savoir que les chemins sont relatifs au dossier de VirtualBox — si l'on ne change pas les options, il s'agit de `.VirtualBox`. Pour corriger cela, on pourra par exemple mettre des chemins absolus.

La façon la plus simple de faire est de commencer par taper :

```
VBoxManage clonehd
```

Ensuite, après avoir ajouté un espace supplémentaire, on prend avec la souris l'icône du disque virtuel à décongeler dans le navigateur de fichiers et on relâche au-dessus du terminal.

Pour ajouter le nouveau disque, on recommence tout d'abord l'opération avec l'icône du dossier *Disques virtuels de VirtualBox* créé plus tôt.

L'affichage devrait à présent ressembler à :

```
VBoxManage clonehd '/home/LOGIN/Disques virtuels propres/Windows XP.vdi'
↪    '/home/LOGIN/Disques virtuels de VirtualBox'
```

Un espace a été ajouté automatiquement avec l'insertion du chemin. On va le supprimer, pour ajouter ensuite le nom du nouveau disque, en écrivant par exemple `/Projet1.vdi`.

Au final, cela doit ressembler d'assez prêt à :

```
VBoxManage clonehd '/home/camille/Disques virtuels propres/Windows XP.vdi'
↪    '/home/camille/Disques virtuels de VirtualBox'/Projet1.vdi
```

Après toutes ces étapes, la ligne de commande est complète, et on peut lancer son exécution en tapant sur la touche *Entrée*.

Créer une nouvelle machine virtuelle

Dans le bureau Debian, aller dans *Applications* → *Outils système* → *VirtualBox OSE*.

Le programme démarre. Cliquer sur *Nouveau* et suivre l'assistant :

- choisir un nom pour la machine virtuelle ;
- choisir le type de système correspondant parmi les Windows proposés ;
- choisir la taille de mémoire vive dédiée à la machine virtuelle, en fonction de la quantité dont on a besoin pour le projet prémédité : si on veut utiliser un gros logiciel comme Photoshop, il faut en prévoir le plus possible (au moins 512 Mo), en sachant que VirtualBox refusera qu'on attribue plus de la moitié de la mémoire totale à la machine virtuelle ;
- choisir d'*Utiliser un disque dur existant* comme *Disque dur d'amorçage*, et choisir l'image précédemment décongelée ;
- cliquer sur suivant et terminer.

Il faut maintenant **avant tout** configurer la machine virtuelle. Cliquer sur le bouton *Préférences* en prenant soin de la sélectionner dans la liste auparavant.

Il faut couper l'accès au réseau (pour des raisons de sécurité déjà mentionnées) :

- Aller dans le sous-menu *Réseau* ;
- Décocher la case *Activer l'adaptateur réseau* dans tous les onglets où elle est déjà cochée par défaut (généralement, dans un seul : le premier).

Cliquer ensuite sur *OK* pour enregistrer les paramètres.

Créer un compte « utilisateur » pour le nouveau projet

Comme expliqué dans le cas d'usage, on souhaite travailler sur un compte « utilisateur » différent pour chaque projet. Voici comment le faire avec Windows XP — ça ne doit pas être trop différent avec d'autres versions.

page 72

Démarrer la nouvelle machine virtuelle en cliquant sur *Lancer*.

Une fois dans le Windows virtualisé, ouvrir *Démarrer* → *Panneau de configuration* puis choisir *Comptes d'utilisateurs* et *Créer un nouveau compte*.

Choisir alors un nom pour le nouveau compte, tout en gardant à l'esprit que ce nom sera probablement enregistré dans les documents créés. Choisir ensuite de créer un compte *Administrateur de l'ordinateur*[1] et cliquer sur *Créer un compte*.

Fermer alors la session à partir du menu *Démarrer*. On veillera à ne plus utiliser pour ce projet que le compte nouvellement créé.

1. Étant donné que l'on utilise un disque virtuel propre pour chaque projet et que l'on a pas accès au réseau, cela ne constitue pas un grand risque, mais nous simplifiera la vie.

24.6 Envoyer des fichiers à un système virtualisé

Vu que le Windows *invité* n'a pas le droit de sortir de sa boîte pour aller chercher lui-même des fichiers, il peut être nécessaire de lui en faire parvenir depuis « l'extérieur ». Voyons donc comment procéder.

Depuis un CD ou DVD

C'est nécessaire si on veut installer d'autres logiciels sous Windows virtuel :

- Insérer le CD à lire dans le lecteur, attendre quelques secondes, puis reprendre le contrôle avec le système hôte (rappel : Ctrl + ⇖ ou Home), et dans la fenêtre accueillant Windows, cliquer sur *Périphérique* → *Insérer un disque optique*, et sélectionner le lecteur CD/DVD.
- Windows devrait alors détecter le CD inséré. Si ce n'est pas le cas, on peut aller le chercher dans *Menu Démarrer* → *Poste de travail*. Si ça ne marche pas du premier coup, recommencer l'opération.

On peut ainsi charger les logiciels depuis le lecteur CD de l'ordinateur : il seront installés durablement sur le disque dur virtuel.

Depuis un dossier

Il est possible de rendre un dossier du système hôte lisible par Windows. Mais veillons à ce que ce ne soit pas n'importe quel dossier…

Créer un dossier réservé à cet effet dans le système hôte

- Réduire la fenêtre accueillant le système client.
- Choisir l'emplacement où on veut mettre ce dossier d'échange. Par exemple, sur le bureau Debian, faire un clic droit puis *Créer un dossier* et lui donner un nom évocateur (« Dossier lisible par Windows », par exemple).

Indiquer au gestionnaire de la machine virtuelle où se trouve ce dossier

- Aller dans la fenêtre de VirtualBox dans laquelle est lancée la machine virtuelle Windows et ouvrir le menu *Périphériques* → *Répertoires partagés…*
- Ajouter un dossier en cliquant sur l'icône avec un « + » en haut à droite. Une boîte de dialogue s'ouvre :
  - dans *Chemin du répertoire* cliquer sur *Autre…* et indiquer l'emplacement du dossier à partager ;
  - dans *Nom du répertoire*, le nom que le dossier aura à l'intérieur du système virtuel s'affiche. Il est possible de le modifier, mais ce nom doit être court, et ne doit pas contenir d'espaces ;
  - cocher la case *Lecture seule*. Ainsi, le système virtuel ne pourra que lire le contenu du dossier, mais rien y écrire ;
  - si, et seulement si, le partage de ce dossier doit être permanent, sélectionner *Mise en place permanente* ; sinon, le partage ne sera activé que pour cette session.

⚠ **Attention** : avant de valider, il faut être bien sûr que l'on veut laisser le système Windows lire tout le contenu du dossier qu'on a demandé de partager. Si c'est bon, cliquer sur *OK*, et refermer la fenêtre avec *OK*.

Indiquer à Windows où se connecter pour trouver ce dossier partagé

- Dans le menu *Démarrer*, ouvrir le *Poste de travail*.
- Dans le menu *Outil*, cliquer sur *Connecter un lecteur réseau*.
- Windows propose un nom de lecteur (par exemple `Z:`) et demande d'indiquer le dossier : cliquer sur parcourir (à droite) → *VirtualBox Shared Folders* → *\\Vboxsvr* → *Nom_du_repertoire*, puis *OK*. On peut choisir au passage si on veut que ce « lecteur » ne soit accessible que pour la durée de la session en cours, ou à chaque nouvelle session.

Attention : en apprenant à utiliser ce système de partage, on pourrait être tenté de le configurer pour donner accès directement aux périphériques branchés sur le système hôte : c'est bien **la pire idée qu'on puisse avoir**, qui anéantirait à elle seule toute la politique de sécurité.

On peut retourner au cas d'usage. page 72

24.7 Faire sortir des fichiers d'un système virtualisé

Le Windows *invité* n'a pas le droit, par défaut, de laisser des traces en dehors de son compartiment étanche. Mais presque inévitablement vient le temps où il est nécessaire d'en faire sortir des fichiers. Voyons donc comment procéder.

En gravant un CD ou DVD

Avant tout, sortir les CD ou les DVD qui pourraient être dans les lecteurs et auxquels on ne veut pas donner accès à la machine virtuelle.

Si la machine virtuelle est en fonction, l'éteindre.

Aller alors dans la fenêtre principale de VirtualBox et sélectionner dans la liste de gauche la machine virtuelle sur laquelle se trouvent les données à graver. Cliquer alors sur l'icône *Préférences*.

Dans la boîte de dialogue des préférences, sélectionner *Supports* dans la liste de gauche, et cliquer sur le lecteur CD. Dans *Lecteur optique* choisir *Lecteur de l'hôte* et cocher *Mode direct*.

Il est alors possible de relancer la machine virtuelle, et de graver les données depuis l'intérieur.

Dans un dossier vide

Il est possible de permettre à Windows d'écrire dans un dossier du système hôte. Mais veillons à ce que ce ne soit pas n'importe quel dossier…

⚠ **Attention** : en apprenant à utiliser ce système de partage, on pourrait être tenté de le configurer pour donner accès directement aux périphériques branchés sur le système hôte : c'est bien **la pire idée qu'on puisse avoir**, qui anéantirait à elle seule toute la politique de sécurité.

Créer un dossier réservé à cet effet dans le système hôte

- Réduire la fenêtre accueillant le système client.
- Choisir l'emplacement où on veut mettre ce dossier d'échange. Par exemple, sur le bureau Debian, faire un clic droit puis *Créer un dossier* et lui donner un nom évocateur, comme « Dossier où Windows peut écrire ».

Indiquer au gestionnaire de la machine virtuelle où se trouve ce dossier

- Aller dans la fenêtre de VirtualBox dans laquelle est lancée la machine virtuelle Windows et ouvrir le menu *Périphériques* → *Répertoires partagés…*
- Ajouter un dossier en cliquant sur l'icône avec un « + » en haut à gauche. Une boîte de dialogue s'ouvre :
  - dans *Chemin du répertoire* cliquer sur *Autre…* et indiquer l'emplacement du dossier à partager ;
  - dans *Nom du répertoire*, le nom que le dossier aura à l'intérieur du système virtuel s'affiche. Il est possible de le modifier ;
  - Si on veut exporter un dossier de façon permanente (et non pas pour cette session uniquement) cocher la case *Mise en place permanente*
  - ne **pas** cocher la case *Lecture seule*.

⚠ **Attention** : avant de valider, il faut être bien sûr que le dossier en question est vide. Windows pourra en effet non seulement y écrire, mais aussi y lire. Si c'est bon, cliquer sur *OK*, et refermer la fenêtre avec *OK*.

Indiquer à Windows où se connecter pour trouver ce dossier partagé

- Dans le menu *Démarrer*, ouvrir le *Poste de travail*.
- Dans le menu *Outils*, cliquer sur *Connecter un lecteur réseau*.
- Windows propose un nom de lecteur (par exemple `Z:`) et demande d'indiquer le dossier : cliquer sur parcourir (à droite) → *VirtualBox Shared Folders* → *\\Vboxsvr* → *Nom_du_dossier*, puis *OK*. On peut choisir au passage si on veut que ce « lecteur » ne soit accessible que pour la durée de la session en cours, ou à chaque nouvelle session.

On peut retourner au cas d'usage. page 72

Qui parle ?

D'où vient cet ouvrage ? Qui parle, en ses lignes ?

Nous pourrions nous contenter de dire qu'il nous semble parfaitement inintéressant de chercher des réponses à de telles interrogations ; que nous laissons aux flics, spécialistes de la question, le privilège de s'y consacrer ; que nous avons mieux à faire.

Le fait que telle ou telle personne couche des mots sur le papier n'est pas, croyons-nous, particulièrement déterminant dans le contenu d'un texte, dans son existence même.

Nous croyons plutôt qu'il s'écrit lorsque des désirs s'entremêlent, lorsque des nécessités se confrontent, lorsque des questions appellent des réponses. Des façons de se rapporter à ce qui nous entoure se rencontrent, se partagent, se transforment alors. Elles se lient, et des manières *communes* de s'y rapporter se construisent, qui interagissent avec d'autres : cela va des conflits aux complicités, en passant par l'alliance et le clin d'œil entendu ; sont alors en jeu sensibilités, critères éthiques, calculs stratégiques…

Bien plus que la « pensée » de X ou Y, un livre exprime l'état de ces interactions, à un certain moment.

*
* *

Deux caractéristiques de cet ouvrage nous obligent néanmoins à faire face, sous certains angles, aux interrogations relatives à sa provenance. Cet ouvrage prétend d'une part transmettre des savoirs et savoirs-faire techniques, réservés d'ordinaire à de rares spécialistes. D'autre part, la justesse des indications fournies peut avoir de larges implications sur la sérénité des personnes qui les mettraient en œuvre. Les petites erreurs qui nous auront échappé peuvent donc avoir de graves conséquences.

Il importe donc de dire quelques mots sur les bouches qui ont prêté leurs voix à ce guide. Mettre au clair l'étendue de nos savoirs(-faire) — et leurs limites — permet de trouver un rapport d'apprentissage plus adéquat à cet écrit, mais aussi de décider du niveau de confiance *technique* qu'il mérite. Disons donc que, collectivement :

- les questions brassées par ce guide nous traversent, techniquement et politiquement, depuis une dizaine d'années ;
- nous connaissons très bien le fonctionnement des systèmes d'exploitation, et particulièrement celui de Debian GNU/Linux ;
- nous avons des bases solides en cryptographie, mais sommes très loin de pouvoir prétendre à une quelconque expertise en la matière.

Et pour finir, affirmons une dernière fois que la parole portée par cet ouvrage, comme toute parole de *guide*, se doit d'être prise avec des pincettes d'autant plus longues que ses implications sont importantes.

Index

Photo page 10 de Darkone, licence CC BY-SA 2.5, trouvée sur :
https://secure.wikimedia.org/wikipedia/fr/wiki/Fichier:ASRock_K7VT4A_Pro_Mainboard.jpg.

Photo page 11, domaine public, trouvée sur :
https://secure.wikimedia.org/wikipedia/fr/wiki/Fichier:Pentium-60-back.jpg

Photo page 11, domaine public, trouvée sur :
https://secure.wikimedia.org/wikipedia/fr/wiki/Fichier:DDR_RAM-3.jpg

Photo page 12, domaine public, trouvée sur :
https://secure.wikimedia.org/wikipedia/fr/wiki/Fichier:Hdd-wscsi.jpg

Photo page 13 de Zac Luzader Codeczero, licence CC BY 3.0, trouvée sur :
https://secure.wikimedia.org/wikipedia/fr/wiki/Fichier:AT_Motherboard_RTC_and_BIOS.jpg.

guide d'autodéfense numérique

tome 1

hors connexions

[…] *nous n'avons pas envie d'être contrôlables par quelque « Big Brother » que ce soit. Qu'il existe déjà ou que l'on anticipe son émergence, le mieux est sans doute de faire en sorte qu'il ne puisse pas utiliser, contre nous, tous ces merveilleux outils que nous offrent — ou que lui offrent — les technologies numériques.* […]

Même si l'on choisit de ne pas les utiliser directement, d'autres le font pour nous. Alors, autant essayer de comprendre ce que ça implique.

Face à ces constats, la seule voie praticable semble être de devenir capables d'imaginer et de mettre en place des politiques de sécurité adéquates.
Tout l'enjeu de ce guide est de fournir cartes, sextant et boussole à quiconque veut cheminer sur cette route.

Ce premier tome se concentre sur l'utilisation d'un ordinateur « hors connexion » — on pourrait aussi bien dire préalablement à toute connexion *: les connaissances générales qu'il apporte valent* que l'ordinateur soit connecté ou non *à un réseau.*

Un livre à lire, relire, pratiquer, en solitaire ou à plusieurs, à faire découvrir et à partager… ou comment affiner l'art de la navigation dans les eaux troubles du monde numérique.