

guide d'autodéfense numérique

tome 1

hors connexions

00	0010		0110	0010
0000	0010	11	10010	10111
1011			11	01111
10100001	111		101010011	1101
0001110111	100		101110011	
101111011	00000		1011011	01100
0000001	001		10111	001

```

        1011    1001
        1110    0011
        1011    0001
        01101100    1010
        101010001    1111
        0010001    0010
        1000    01
        1100    0010
        1110    0100
        0111    1100
        010100
        001100011    0000
        1000001011010101
        11110110    0001
        00111

```

cinquième édition

automne 2017

```

00110 01000
10010 11001111
110010 001111
0010011001 10100
1110100000 0100
11111110001110
11100 000

```

Guide d'autodéfense numérique

Tome 1 : Hors connexions
cinquième édition

Ouvrage collectif
guide@boum.org

Empreinte OpenPGP :
D487 4FA4 F6B6 88DC 0913
C9FD 326F 9F67 250B 0939

automne 2017



Copyleft : cette œuvre est libre, vous pouvez la copier, la diffuser et la modifier selon les termes de la *Licence Art Libre* — <http://www.artlibre.org/>

Préface

Les revers de la mémoire numérique

De nos jours, les ordinateurs, Internet et les téléphones portables 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 simples 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 se faire cambrioler parce que son adresse a été publiée sur Internet malgré soi et son absence confirmée sur les réseaux sociaux.

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 sont arrêtées sur la base des traces laissées par l'utilisation d'outils numériques dans le cadre d'activités qui ne plaisaient pas à un gouvernement, pas forcément le leur d'ailleurs — et pas seulement en Chine ou en Iran.

Beaucoup de gens, que ce soient les gouvernants, les employeurs, les publicitaires ou les flics¹, ont intérêt à obtenir l'accès à nos données. La place croissante que prend l'information dans l'économie et la politique mondiales ne peut que les encourager. On sait d'ailleurs déjà qu'ils ne se gênent pas pour faire des recouplements entre les individus. Or, que savons-nous des pratiques légales et illégales de nos proches ?

De plus, comment savoir si ce qui est autorisé aujourd'hui le sera demain ? Les gouvernements changent, les lois et les situations aussi. Et cela peut aller extrêmement vite, comme de nombreuses personnes ont pu le constater avec l'application de l'état d'urgence en France, reconduit depuis fin 2015². 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 ? Des traces *auront été laissées* sur l'ordinateur... et pourraient être employées comme éléments à charge.

Mettre en place des pratiques de protection des données lorsqu'on a le sentiment de ne pas directement en avoir besoin permet aussi de les rendre plus « normales », plus acceptables et moins suspectes. Les personnes qui n'ont pas d'autre possibilité pour survivre que de cacher leurs activités numériques nous en seront reconnaissantes, sans aucun doute.

De manière générale, nous bridons nos actions dès que nous savons que d'autres peuvent nous écouter, nous regarder ou nous lire. Chanterions-nous sous la douche si l'on savait que des micros y sont installés ? Apprendrions-nous à danser si des caméras étaient pointées sur nous ? Écririons-nous une lettre intime aussi librement si une personne lisait par dessus notre épaule ? Avoir des choses à cacher n'est pas seulement une question de légalité, mais aussi d'intimité.

Ainsi, à l'époque des sociétés de contrôle de plus en plus paranoïaques, de plus en plus résolues à traquer la subversion et à voir derrière chaque être humain un terroriste en puissance qu'il faut surveiller de près, se cacher devient un enjeu *politique* et de fait *collectif*. Ne serait-ce que pour mettre des bâtons dans les roues de ceux qui nous voudraient transparentes et repérables en permanence.

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.

Alors, *ayons aussi 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 », jamais inoffensives 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 ; eh 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) ? »

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/>].

2. Wikipédia, 2017, *État d'urgence en France* [https://fr.wikipedia.org/wiki/Etat_d%27urgence_en_France].

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 toute 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.

Avoir de la patience, 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.

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 deux tomes. Qu'on se trouve avec uniquement un ordinateur ou que ce dernier soit connecté à un réseau, cela représente des contextes différents, donc des menaces, des envies et des réponses différentes elles aussi.

Préface à la cinquième édition

Moins d'un an après la parution de la dernière édition en ligne du *Guide*, nous devions déjà nous attacher à préparer la suivante, à la fois pour offrir une nouvelle édition papier ainsi que pour suivre l'évolution des outils que nous recommandons ou encore des lois que nous subissons.

Quelques mois après la plus grande fuite³ de documents confidentiels de la *Central Intelligence Agency*⁴, l'installation dans la permanence de l'état d'urgence en France confirme que la tendance est toujours à la normalisation de la surveillance tous azimuts. Une tendance que la fuite des documents secrets de la *National Security Agency*⁵ par Edward Snowden⁶ avait déjà annoncée. En effet, toute la panoplie des outils de surveillance ou d'infiltration électronique extra-légale révélée au fur et à mesure des scandales rentre petit à petit dans l'arsenal législatif. Et quand ce n'est pas le cas, la nécessité devient sa propre justification, permettant ainsi aux agences gouvernementales de les utiliser sans scrupules ni risque de choquer l'opinion publique.

La maigre consolation qu'on peut tirer de ce nouveau contexte, c'est qu'on sait plus clairement contre quoi se protéger. Mais cela relance aussi la course à la sécurité informatique, en forçant les attaquants à recourir à des techniques plus poussées, comme l'utilisation de failles informatiques encore inconnues du public, contre lesquelles personne ne peut donc trouver de correctif, aussi appellées vulnérabilités Zero day⁷. Vulnérabilité par exemple utilisée par le FBI, en 2015, lors de l'opération *Pacifier*⁸, ou encore dans le cas du *ransomware*⁹ *WannaCry*¹⁰ qui a touché plus de 300 000 ordinateurs à travers le monde au printemps 2017.

Les scénarios les plus alarmistes sont finalement le tissu du quotidien en matière de surveillance électronique. Malgré la propagation d'un sentiment d'impuissance, ces différentes révélations sur l'état général de la surveillance numérique rendent d'autant plus nécessaire de se donner les moyens d'y faire face.

Du côté des outils, le mois de juin 2017 a vu la sortie de la nouvelle version de Debian, baptisée « *Stretch* » ainsi que la version 3.0 du système *live Tails* dorénavant basée sur *Stretch*. Cette mise à jour a apporté de nombreux changements tant au niveau graphique que dans les logiciels proposés. Cela à également amené à des changements

3. Wikipédia, 2017, *Vault 7* [https://fr.wikipedia.org/wiki/Vault_7].

4. Wikipédia, 2017, *CIA* [https://fr.wikipedia.org/wiki/Central_Intelligence_Agency].

5. Wikipédia, 2017, *NSA* [<https://fr.wikipedia.org/wiki/NSA>].

6. Wikipédia, 2017, *Edward Snowden* [https://fr.wikipedia.org/wiki/Edward_Snowden].

7. Wikipédia, 2017, *Vulnérabilité Zero day* [https://fr.wikipedia.org/wiki/Vuln%C3%A9rabilit%C3%A9_Zero_day].

8. Joseph Cox, 2016, *Pressure Mounts on FBI To Reveal Tor Browser Exploit* [<https://motherboard.vice.com/read/pressure-mounts-on-fbi-to-reveal-tor-browser-exploit-playpen>] (en anglais).

9. Wikipédia, 2017, *Ransomware* [<https://fr.wikipedia.org/wiki/Ransomware>].

10. Wikipédia, 2017, *WannaCry* [<https://fr.wikipedia.org/wiki/WannaCry>].

dans ce présent guide, avec notamment le remplacement de *VirtualBox* par le *Gestionnaire de machine virtuelle*. Il a donc fallu revoir les outils pour que les recettes fonctionnent sur ces nouveaux systèmes.

Et pour les personnes ayant déjà un système installé avec la version précédente de Debian (*Jessie*), un chapitre de cette cinquième édition explique comment procéder à la mise à jour vers Debian *Stretch*.

Grâce à cette révision, nous espérons que les pages suivantes restent d'une compagnie avisée dans la traversée de la jungle numérique... du moins, jusqu'à la suivante.

Tome 1

Hors connexions

Sommaire

Préface	iii
Les revers de la mémoire numérique	iii
Rien à cacher ?	iii
Comprendre pour pouvoir choisir	iv
Prendre le temps de comprendre	v
Un « guide »	v
Préface à la cinquième édition	vii
Sommaire	1
I Comprendre	7
1 Quelques bases sur les ordinateurs	9
1.1 Des machines à traiter les données	9
1.2 Le matériel	9
1.3 Électricité, champs magnétiques, bruits et ondes radios	15
1.4 Les logiciels	15
1.5 Le rangement des données	17
2 Traces à tous les étages	19
2.1 Dans la mémoire vive	19
2.2 Dans la mémoire virtuelle	20
2.3 Veille et hibernation	20
2.4 Les journaux	21
2.5 Sauvegardes automatiques et autres listes	21
2.6 Les métadonnées	22
3 Malware, mouchards et autres espions	23
3.1 Contexte légal	23
3.2 Les logiciels malveillants	24
3.3 Les matériels espions	26
3.4 Les <i>keyloggers</i> , ou enregistreurs de frappe au clavier	27
3.5 Des problèmes d'impression ?	28
4 Quelques illusions de sécurité...	31
4.1 Logiciels propriétaires, <i>open source</i> , libres	31
4.2 Le mot de passe d'un compte ne protège pas ses données	33
4.3 À propos de l'« effacement » des fichiers	34
4.4 Les logiciels portables : une fausse solution	36

5 La cryptographie	39
5.1 Protéger des données des regards indiscrets	39
5.2 S'assurer de l'intégrité de données	44
5.3 Symétrique, asymétrique ?	46
II Choisir des réponses adaptées	49
<hr/>	
6 Évaluation des risques	51
6.1 Que veut-on protéger ?	51
6.2 Contre qui veut-on se protéger ?	51
7 Définir une politique de sécurité	53
7.1 Une affaire de compromis	53
7.2 Comment faire ?	54
7.3 Quelques règles	54
8 Un nouveau départ	59
8.1 Contexte	59
8.2 Évaluer les risques	59
8.3 Définir une politique de sécurité	60
9 Travailler sur un document sensible	67
9.1 Contexte	67
9.2 Évaluer les risques	67
9.3 Accro à Windows ?	68
9.4 Le système <i>live</i> amnésique	68
9.5 Travailler sur un document sensible... sur un système <i>live</i>	70
9.6 Travailler sur un document sensible... sous Windows	70
9.7 Nettoyer les métadonnées du document terminé	77
9.8 Limites communes à ces politiques de sécurité	77
10 Archiver un projet achevé	79
10.1 Contexte	79
10.2 Est-ce bien nécessaire ?	79
10.3 Évaluer les risques	79
10.4 Méthode	80
10.5 Quelle phrase de passe ?	80
10.6 Un disque dur ? Une clé ? Plusieurs clés ?	81
III Outils	83
<hr/>	
11 Utiliser un terminal	87
11.1 Qu'est-ce qu'un terminal ?	87
11.2 À propos des commandes	88
11.3 Priviléges d'administration	90
11.4 Encore une mise en garde	90
11.5 Un exercice	90
11.6 Attention aux traces !	92
11.7 Pour aller plus loin	92
12 Choisir une phrase de passe	93

13 Démarrer sur un CD, un DVD ou une clé USB	95
13.1 Essayer naïvement	95
13.2 Tenter de choisir le périphérique de démarrage	95
13.3 Modifier les paramètres du micrologiciel	96
14 Utiliser un système <i>live</i>	101
14.1 Des systèmes <i>live</i> discrets	101
14.2 Télécharger et installer <i>Tails</i>	102
14.3 Cloner ou mettre à jour une clé <i>Tails</i>	103
14.4 Démarrer sur un système <i>live</i>	103
14.5 Utiliser la persistance de <i>Tails</i>	103
15 Installer un système chiffré	107
15.1 Limites	107
15.2 Télécharger un support d'installation	108
15.3 Vérifier l'empreinte du support d'installation	109
15.4 Préparer les supports d'installation	110
15.5 L'installation proprement dite	111
15.6 Quelques pistes pour continuer	114
15.7 Un peu de documentation sur <i>Debian</i> et GNU/Linux	114
16 Choisir, vérifier et installer un logiciel	117
16.1 Trouver un logiciel	118
16.2 Critères de choix	119
16.3 Installer un paquet <i>Debian</i>	122
16.4 Utiliser des logiciels rétroportés	124
17 Effacer des données « pour de vrai »	127
17.1 Un peu de théorie	127
17.2 Sur d'autres systèmes	128
17.3 Allons-y	128
17.4 Supprimer des fichiers... et leur contenu	129
17.5 Effacer « pour de vrai » tout un disque	129
17.6 Effacer tout le contenu d'un disque	130
17.7 Rendre irrécupérables des données déjà supprimées	132
18 Partitionner et chiffrer un disque dur	133
18.1 Chiffrer un disque avec LUKS et <i>dm-crypt</i>	133
18.2 D'autres logiciels que l'on déconseille	134
18.3 En pratique	134
18.4 Préparer un disque à chiffrer	135
18.5 Créer une partition non chiffrée	136
18.6 Créer une partition chiffrée	136
18.7 Utiliser un disque dur chiffré	137
19 Sauvegarder des données	139
19.1 Gestionnaire de fichiers et stockage chiffré	139
19.2 En utilisant <i>Déjà Dup</i>	141
20 Partager un secret	145
20.1 Partager une phrase de passe	145
20.2 Reconstituer la phrase de passe	147
21 Utiliser les sommes de contrôle	149
21.1 Obtenir la somme de contrôle d'un fichier	149
21.2 Vérifier l'intégrité d'un fichier	150

22 Installer et utiliser un système virtualisé	151
22.1 Installer le <i>Gestionnaire de machine virtuelle</i>	152
22.2 Installer un <i>Windows</i> virtualisé	153
22.3 Prendre un instantané d'une machine virtuelle	156
22.4 Restaurer l'état d'une machine virtuelle à partir d'un instantané	157
22.5 Partager un CD ou un DVD avec un système virtualisé	158
22.6 Partager un dossier avec un système virtualisé	158
23 Garder un système à jour	161
23.1 Garder <i>Tails</i> à jour	161
23.2 Garder à jour un système chiffré	161
23.3 Les mises à jour quotidiennes d'un système chiffré	162
23.4 Passage à une nouvelle version stable	162
24 Nettoyer les métadonnées d'un document	167
24.1 Installer les logiciels nécessaires	167
24.2 Ouvrir le <i>Metadata Anonymisation Toolkit</i>	167
24.3 Ajouter des fichiers à nettoyer	167
24.4 Nettoyer les fichiers	168
Qui parle ?	169
Index	171
Crédits	173

00	0010		0110	0010
0000	0001	011	00010	0111
101		01010		111
101000	011		101010011	1101
000111011	100		101110011	1
101111101	00000		1011011	1100
000001	01		0111	001

```

00110 01000
00101 011001111
00010 0011111
0010011000 010100
1110100000 0100
11111110000111
11000 0001

```

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 / intimité¹. 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.

1. 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 aimeraient mettre derrière cette notion. Ailleurs, on rencontrera souvent le terme « sécurité », mais l'usage qui en est couramment fait nous donne envie de l'éviter.

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 précisément 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.

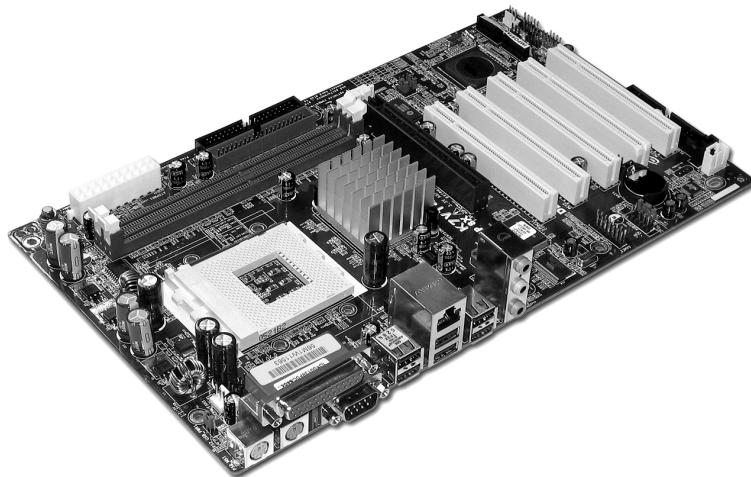
1.2 Le matériel

Somme de composants reliés 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 : téléphone portable, « box » de connexion à Internet, tablette, lecteur MP3, caisse enregistreuse, compteur Linky¹, ordinateur de bord de voiture, *etc.*

1.2.1 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 ou autre mémoire), de quoi démarrer l'ordinateur (un micrologiciel) 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.

1.2.2 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'opérations 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.

1. Les compteurs Linky sont les remplaçants des compteurs électriques historiques - Page Wikipedia [<https://fr.wikipedia.org/wiki/Linky>].



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

Cette série de chiffres permet de représenter les lettres qui composent notre mot. Mais le processus de numérisation perdra toujours de l'information. Pour cet exemple, on perd au passage la spécificité de l'écriture manuscrite alors que pourtant, une rature, des lettres hésitantes constituent tout autant de « l'information ». Lorsque des choses passent dans le tamis du monde numérique, on perd forcément toujours des morceaux au passage.

Au-delà des données, 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.

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*, dont l'unité de mesure est le *bit*². 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.

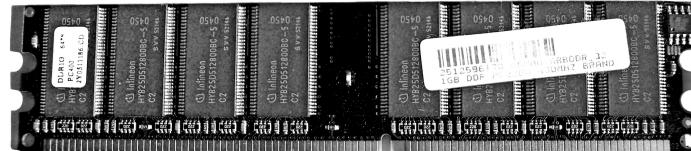
Tous les processeurs ne fonctionnent pas de la même manière. Certains ont été conçus pour être plus efficaces pour certains types de calcul, d'autres pour consommer le moins d'énergie, etc. Par ailleurs, tous les processeurs ne disposent pas exactement des

2. Pour en savoir plus, voir Wikipédia, 2014, *Bit* [<https://fr.wikipedia.org/wiki/Bit>].

mêmes instructions. Il en existe de grandes familles, que l'on appelle des *architectures*. Cela a son importance, car un programme prévu pour fonctionner sur une architecture donnée ne fonctionnera en général pas sur une autre.

1.2.3 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.

1.2.4 Le disque dur



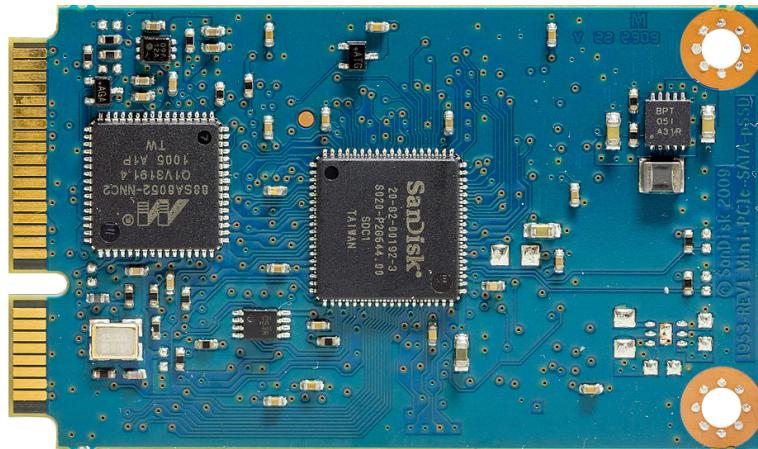
Un disque dur 3 pouces ½

[page 10]

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 utilisait 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 informations à stocker. Ces informations sont stockées sous forme de *bits*, dont plusieurs unités de mesure existent³, permettant de quantifier plus simplement la capacité d'un disque dur, notamment, en termes de mégaoctet (Mo), gigaoctet (Go), etc.

Du fait des mouvements mécaniques, les disques durs rotationnels sont lents. C'est pourquoi, en 2016, plus d'un tiers des ordinateurs portables neufs contiennent un disque SSD à la place d'un disque dur⁴. Un disque SSD est en fait une mémoire flash, la même qui est présente dans les clés USB et les cartes SD. Cette mémoire entièrement électronique est beaucoup plus rapide que les disques durs magnétiques (environ 25 fois plus rapide qu'un disque dur rotationnel).



Un disque SSD avec connexion mSATA

Les disques durs comme les disques SSD permettent de stocker *beaucoup plus d'informations* que la mémoire vive.

Les informations que l'on met donc généralement sur un disque (dur ou SSD) 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 conserve donc une mémoire quasi-permanente et quasi-exhaustive de toutes sortes de traces qui parlent de nous, de ce que nous faisons, avec qui et comment, dès qu'on utilise un ordinateur.

3. Wikipédia, 2017, *Octet* [<https://fr.wikipedia.org/wiki/Octet>].

4. TrendForce, 2016, *SSD Adoption in Notebooks May Hit 40% by Year's End*, I-Connect 007 [<http://ein.iconnect007.com/index.php/article/97843/ssd-adoption-in-notebooks-may-hit-40-by-years-end/>] (en anglais).

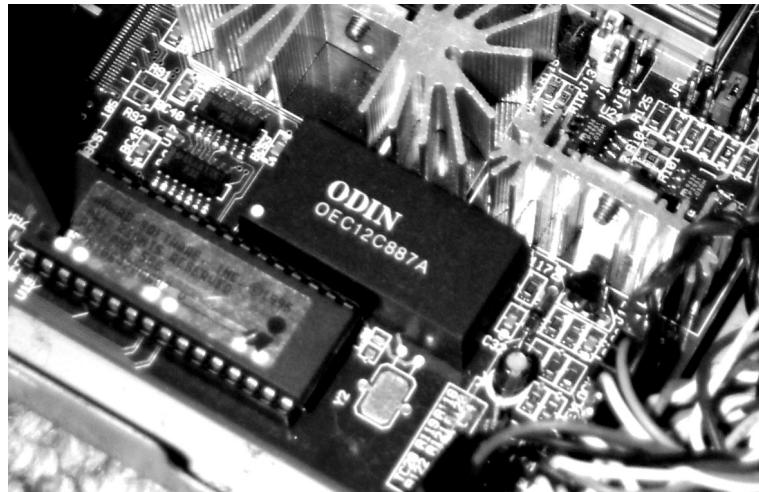
1.2.5 Les autres périphériques

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 être 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, le standard USB (pour *Universal Serial Bus*) est devenu à peu près la norme pour connecter imprimantes, claviers, souris, disques durs supplémentaires, adaptateurs réseaux ou ce qu'on appelle couramment des « clés USB ».

1.2.6 Le micrologiciel de la carte mère



Une puce de micrologiciel Award BIOS sur une carte-mère

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

Ce petit logiciel, appelé micrologiciel (*firmware* en anglais) de la carte mère est contenu dans une puce mémoire sur celle-ci. Cette mémoire fait partie d'un troisième type : la mémoire *flash*. C'est aussi ce type de mémoire qu'on trouve dans les « clés USB » ou les « disques durs » dits *Solid State Drive* (ou SSD). C'est une puce mémoire qui garde les informations lorsqu'elle est éteinte, mais dont on peut remplacer le contenu lors d'une opération qu'on appelle *flashage*.

Le micrologiciel historique de la plupart des ordinateurs personnels était appelé BIOS (*Basic Input/Output System*, ou système d'entrée/sortie de base). Depuis 2012, de plus en plus d'ordinateurs utilisent un nouveau standard appelé UEFI (*Unified Extended Firmware Interface*).

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 ou d'un DVD, voire à partir du réseau).

1.3 Électricité, champs magnétiques, bruits 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 des 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, voire plus⁵. 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 mètres⁶.

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 branché⁷.

D'autres expériences consistant à écouter avec un microphone le bruit des composants électroniques de l'ordinateur ainsi que de son alimentation électrique, ont permis dans certaines conditions de déchiffrer des clés de chiffrement contenues sur l'ordinateur cible⁸. Des corrections des logiciels impliqués ont depuis été publiées afin de compliquer ce type d'attaque.

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 tout de même possible pour des personnes bien équipées de réaliser une « écoute » de ce qui se passe à l'intérieur.

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 tout premiers ordinateurs, chaque fois qu'il fallait exécuter des traitements différents, il fallait intervenir physiquement pour changer la disposition des câbles 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. Ces données, qu'on appelle « programmes », sont chargées, modifiées, et manipulées par d'autres programmes.

5. Berke Durak a réussi en 1995 à capter les ondes électromagnétiques [<http://lambda-diode.com/electronics/tempst/>] émises par la plupart des composants de son ordinateur avec un simple *walkman* capable de recevoir la radio.

6. 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.

7. Paul Kocher, Joshua Jaffe et Benjamin Jun ont publié en 1998, en anglais, un rapport [<http://www.cryptography.com/public/pdf/DPATechInfo.pdf>] expliquant les différentes techniques d'analyse de consommation électrique.

8. Clément Bohic, 2013, *Chiffrement : il suffirait d'écouter le processeur pour décoder les clefs*, silicon.fr [<http://www.silicon.fr/chiffrement-ecouter-processeur-decoder-clefs-91686.html>].

Ceux-ci sont généralement écrits pour essayer de ne faire qu'une seule chose, et de la faire bien, principalement dans le but de 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'événements, d'une somme impressionnante de calculs, qui aboutissent à des impulsions électriques venant à la fin modifier un objet physique (comme un DVD qu'on 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*).

1.4.1 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, imprimante, etc.), de stockage (DVD, 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 DVD. Dans ce dernier cas, on parle de système *live* (vu qu'aucune modification ne pourra être faite sur le DVD).

1.4.2 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 exemples Mozilla Firefox comme navigateur web, LibreOffice pour la bureautique ou encore VLC pour la lecture de musique et de vidéo.

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 être conçues pour le système d'exploitation de l'ordinateur sur lequel on veut s'en servir.

1.4.3 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...

1.5.1 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 : cela s'appelle un système de fichiers.

1.5.2 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.

Notons toutefois 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 34]

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 fichiers est souvent fortement lié à un système d'exploitation particulier.

Pour en citer quelques-un : les types NTFS et 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.

Il est néanmoins possible de lire un système de fichiers « étranger » au système qu'on utilise, moyennant l'usage du logiciel adéquat. Windows est par exemple capable de lire une partition *ext3*, si on installe le logiciel approprié.

Une des conséquences de cela est qu'il peut exister sur un ordinateur donné des espaces de stockage non reconnus par le système d'exploitation, auxquels on ne pourra donc pas accéder aisément.

1.5.3 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 du nom d'un fichier un code, qu'on appelle parfois *extension*, permettant d'indiquer le format du fichier. On peut choisir une extension ou une autre, la modifier, mais cela est surtout à titre indicatif, et ne signifie pas, qu'en la changeant, on change le format de fichier.

Quelques exemples d'extensions : pour la musique, on utilisera souvent les formats MP3 ou Ogg, pour un document texte de LibreOffice ce sera OpenDocument Text (ODT), pour des images, on aura le choix entre JPEG, PNG ou d'autres, *etc.*

[page 31] Comme les logiciels, les formats peuvent être *ouverts* ou *propriétaires*. Les formats *ouverts* sont donc définis publiquement, afin, entre autre, de ne pas restreindre leur utilisation à un seul logiciel.

Certains formats propriétaires ont été observés à la loupe pour être utilisables par d'autres logiciels, mais leur compréhension reste souvent imparfaite. C'est typiquement le cas pour l'ancien format de Microsoft Word (DOC) ou celui d'Adobe Photoshop (PSD).

1.5.4 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 auxquelles il veut réellement accéder. La *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, parfois avec Linux), 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.

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 12]

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, événements 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 que d'y brancher un *iPod* quand on a le dos tourné¹. Une fois récupérées, les nombreuses informations que contient la mémoire vive, sur les personnes qui utilisent cet ordinateur par exemple, pourront alors être exploitées...

Par ailleurs, ces données deviennent illisibles lors de la mise hors tension. Cela prend néanmoins du temps, 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 jours².

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. Maximillian Dornseif, 2004, *Owned by an iPod* [<https://pacsec.jp/psj04/psj04-dornseife.ppt>]. Bruce Schneier, 2006, *Hacking Computers Over USB* [https://www.schneier.com/blog/archives/2006/06/hacking_compute.html] (en anglais).

2. J. Alex Halderman et Al., 2008, *Least We Remember : Cold Boot Attacks on Encryption Keys* [<https://citp.princeton.edu/memory/>] (en anglais).

2.2 Dans la mémoire virtuelle

[page 18] 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 36] 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.

[page 39] Pour éviter de laisser n'importe qui accéder à ces données, il est possible d'utiliser un système d'exploitation configuré pour chiffrer la mémoire virtuelle.

2.3 Veille et hibernation

La plupart des systèmes d'exploitation permettent 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.

2.3.1 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.

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

2.3.2 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 recommencera à travailler à partir de là.

[page 18] Sur les systèmes GNU/Linux, la copie de la mémoire se fait généralement dans la *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 39] 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 support amovible (disque externe, clé USB...) 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 LibreOffice 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 (clé USB, disque dur externe, CD ou DVD), 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'érasement 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 développement porte une attention particulière à cette question, garantit que ces traces ne seront pas laissées ailleurs que dans la mémoire vive. Nous y reviendrons.

2.6 Les méta-données

En plus des informations contenues dans un fichier, il existe des informations accompagnant celui-ci, qui ne sont pas forcément visibles de prime abord : date de création, nom du logiciel, de l'ordinateur, *etc.* Ces « données sur les données » s'appellent communément des « méta-données ».

page 17 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. Ces traces sont laissées sur l'ordinateur (ce qui peut déjà être un problème en soi), mais celles-ci ne sont la plupart du temps pas inscrites dans le fichier.

page 18 En revanche, de nombreux formats de fichiers conservent également des méta-données à l'intérieur du fichier. Elles seront donc diffusées lors d'une éventuelle copie sur une clé USB, ou lors de l'envoi d'un email ou d'une publication en ligne. Ces informations pourront 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 complet des dernières modifications³, et donc, potentiellement, des informations que l'on pensait avoir supprimé...

La palme revient probablement aux formats d'images comme TIFF ou JPEG : ces fichiers de photos créés par un appareil numérique ou un téléphone portable contiennent des méta-données au format EXIF. Ce dernier peut contenir la marque, le modèle et le numéro de série de l'appareil utilisé, mais aussi la date, l'heure et parfois les coordonnées géographiques de la prise de vue, sans oublier une version miniature de l'image. Ce sont d'ailleurs ces méta-données qui mettront fin à la cavale John McAfee, fondateur et ancien patron de la société de sécurité informatique du même nom⁴. Et toutes ces informations ont tendance à rester après être passées par un logiciel de retouche photo. 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 ».⁵

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

3. Deblock Fabrice, 2006, *Quand les documents Word trahissent la confidentialité* [<http://www.journaldunet.com/solutions/0603/060327-indiscretions-word.shtml>].

4. Big Brower, 2012, *Vice de forme – La bourde qui a mené à l'arrestation de John McAfee* [<http://bigbrowser.blog.lemonde.fr/2012/12/12/vice-de-forme-la-bourde-qui-a-mene-a-larrestation-de-john-mcafee/>].

5. Maximilian Dornseif et Steven J. Murdoch, 2004, *Hidden Data in Internet Published Documents* [<http://events.ccc.de/congress/2004/fahrplan/files/316-hidden-data-slides.pdf>] (en anglais).

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.

[page 31]

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 auteurs de violences domestiques¹, 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 personnes ou 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 légale², et, malgré les apparences³, il ne faut pas croire que Microsoft donne toutes les astuces de Windows à la police française.

3.1 Contexte légal

Cependant, les flics et les services de sécurité français disposent maintenant des moyens de mettre en place une surveillance informatique très complète en toute légalité, en s'appuyant sur plusieurs « mouchards » présentés par la suite.

1. Catherine Armitage, 2014, *Spyware's role in domestic violence* [<http://www.theage.com.au/technology/technology-news/spywares-role-in-domestic-violence-20140321-358sj.html>] parle de l'utilisation des malwares et autres outils technologiques par des auteurs de violences domestiques (en anglais).

2. Pour ce faire une idée des problématiques liées à l'espionnage industriel : Wikipédia, 2014, *Espionnage industriel* [https://fr.wikipedia.org/wiki/Espionnage_industriel].

3. Microsoft, en partenariat avec Interpol, a fabriqué une boîte à outils appelée COFEE (Computer Online Forensic Evidence Extractor) mise à disposition des polices d'une quinzaine de pays. Korben, 2009, *Cofee – La clé sécurité de Microsoft vient d'apparaître sur la toile* [<https://korben.info/cofee-la-cle-securite-de-microsoft-vient-dapparaitre-sur-la-toile.html>].

La loi « renforçant la lutte contre le crime organisé, le terrorisme et leur financement, et améliorant l'efficacité et les garanties de la procédure pénale » de 2016⁴ inclut des dispositions légales qui permettent d'installer des mouchards pour enregistrer et communiquer ce qui s'affiche à l'écran ou ce que les différents périphériques (clavier, webcam, scanner, téléphone portable,...) transmettent à l'ordinateur. La « pose » des ces mouchards est autorisée à distance ou en pénétrant dans le domicile de la personne surveillée pour y installer les outils nécessaires⁵. Ces mesures ne s'appliquent pas qu'aux actes relevant du « terrorisme », (comme la « prolifération des armes de destruction massive »), mais aussi à nombre de crimes et délits dès lors qu'ils sont commis à plusieurs (en « bande organisée »). Cela peut aller de l'aide « à la circulation et au séjour irréguliers d'un étranger en France » en passant par la « destruction, dégradation et détérioration d'un bien »⁶, mais aussi être sur simple demande du Procureur de la République en « cas d'urgence résultant d'un risque imminent de déteriorissement des preuves ou d'atteinte grave aux personnes ou aux biens ».

La loi relative au renseignement de 2015⁷ donne à peu près les mêmes pouvoirs⁸ aux « services spécialisés de renseignement » pour « la recherche, la collecte, l'exploitation et la mise à disposition du Gouvernement des renseignements relatifs aux enjeux géopolitiques et stratégiques ainsi qu'aux menaces et aux risques susceptibles d'affecter la vie de la Nation »⁹.

3.2 Les logiciels malveillants

Les logiciels malveillants¹⁰ (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* font partie de cette famille. Certains programmes peuvent appartenir à plusieurs de ces catégories simultanément.

Afin de s'installer sur un ordinateur, certains logiciels malveillants exploitent les vulnérabilités du système d'exploitation¹¹ 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.

4. République française, 2016, *loi n° 2016-731 du 3 juin 2016 renforçant la lutte contre le crime organisé, le terrorisme et leur financement, et améliorant l'efficacité et les garanties de la procédure pénale* [<https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000032627231>].

5. République française, *Code de procédure pénale*, articles 706-95-4 et 706-95-5 [<https://www.legifrance.gouv.fr/affichCode.do?idSectionTA=LEGISCTA000032653938&cidTexte=LEGITEXT000006071154#LEGIARTI000032631879>].

6. République française, *Code de procédure pénale*, articles 706-73 et 706-73-1 [<https://www.legifrance.gouv.fr/affichCode.do?idSectionTA=LEGISCTA00006138138&cidTexte=LEGITEXT000006071154>].

7. République française, 2015, *loi n° 2015-912 du 24 juillet 2015 relative au renseignement* [<https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000030931899>].

8. République française, *Code de la Sécurité Intérieure*, article L853-2 [<https://www.legifrance.gouv.fr/affichCodeArticle.do?idArticle=LEGIARTI000030935977&cidTexte=LEGITEXT000025503132>].

9. République française, *Code de la Sécurité Intérieure*, article L811-2 [<https://www.legifrance.gouv.fr/affichCodeArticle.do?idArticle=LEGIARTI000030939233&cidTexte=LEGITEXT000025503132>].

10. Toute cette partie est grandement inspirée du passage consacré à la question dans le *Surveillance Self-Defense Guide* [<https://ssd.eff.org/fr/module/comment-puis-je-me-prot%C3%A9ger-contre-les-logiciels-malveillants>] de l'*Electronic Frontier Foundation*.

11. D'après l'*Internet Storm Center* [<https://isc.sans.edu/survivaltime.html>], une installation de Microsoft Windows sur laquelle les mises à jour de sécurité n'ont pas été faites se fait compromettre en environ 7 heures si elle est connectée directement à Internet en 2016.

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. C'est ainsi qu'un simple lien vers une vidéo posté sur un réseau social lié à la révolution syrienne amenait en fait les internautes à télécharger un virus contenant un keylogger¹². 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. Le cas sans doute le plus compliqué est celui de failles jusqu'alors inconnues, appelées « exploits zero day »¹³, et que les logiciels antivirus seraient bien en mal de reconnaître, car pas encore répertoriées. C'est exactement ce genre d'exploitation de failles « zero day » que la compagnie VUPEN a vendu à la NSA en 2012¹⁴.

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. Mais ils peuvent également utiliser le micro, la webcam ou d'autres périphériques de l'ordinateur. Il existe un vrai marché spécialisé où l'on peut acheter de tels programmes, personnalisés pour différents objectifs.

[page 39]

Ces logiciels permettent d'effectuer de nombreuses opérations : obtenir des numéros de cartes bancaires, des mots de passe de compte *PayPal*, à envoyer des pourriels, à participer à attaquer un serveur en le saturant de demandes, *etc.* Mais ils servent tout aussi bien à espionner des organisations ou des individus spécifiques¹⁵.

12. Eva Galperin et Al., 2014, *Quantum of Surveillance : Familiar Actors and Possible False Flags in Syrian Malware Campaigns* [https://www.eff.org/files/2013/12/28/quantum_of_surveillance4d.pdf] (en anglais).

13. Wikipédia, 2014, *Vulnérabilité jour zéro* [https://fr.wikipedia.org/wiki/Vuln%C3%A9rabilit%C3%A9_zero-day].

14. Grégoire Fleurot, 2013, *Espionnage : Vupen, l'entreprise française qui bosse pour la NSA* [<https://www.slate.fr/france/77866/vupen-nsa-espiionage-exploits>].

15. Ministry of Justice of Georgia et Al., 2012, *Cyber Espionage Against Georgian Government* [<http://dea.gov.ge/uploads/CERT%20DOCS/Cyber%20Espionage.pdf>] (en anglais).

Pour donner un exemple venu des Émirats Arabes Unis, un militant des droits humains, Ahmed Mansour, a été victime d'une attaque ciblée sur son smartphone¹⁶. Un SMS contenant un lien vers un virus lui a été envoyé. Ce virus permettait à la personne le contrôlant d'utiliser la caméra, le micro et de surveiller les activités du téléphone de la victime à tout instant. L'attaque a été déjouée et disséquée grâce à Citizen Lab.

[page 23] Les services de renseignement et les flics français ont le droit d'utiliser légalement de tels logiciels, ce qui veut très certainement dire qu'ils en disposent. Une suite de logiciels espions attribuée aux services de renseignement français a d'ailleurs été découverte notamment en Iran¹⁷.

Personne ne sait combien d'ordinateurs sont infectés par des logiciels malveillants, mais certaines personnes 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 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 venu¹⁸ ;
- prendre au sérieux, c'est-à-dire considérer un tant soit peu 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.

3.3 Les matériels espions

[page 24] Les adversaires voulant mettre la main sur les secrets contenus par nos ordinateurs peuvent utiliser des logiciels malveillants comme on vient de le voir, mais peuvent tout aussi bien utiliser du matériel espion. Et celui-ci n'a rien à envier au bon vieux James Bond !

Il existe toute une gamme de matériel plus ou moins facilement disponible permettant intrusion ou exfiltration d'information à quasiment tous les niveaux d'un ordinateur. Suite à la publication de documents confidentiels de la NSA via Edward Snowden, un véritable catalogue d'espionnage informatique a été publié sur le journal allemand Spiegel¹⁹.

16. Andréa Fradin, 2016, « *Pegasus* », l'arme d'une firme israélienne fantôme qui fait trembler Apple [<http://tempsreel.nouvelobs.com/rue89/rue89-surveillance/20160826.RUE3689/pegasus-1-arme-d-une-firme-israelienne-fantome-qui-fait-trembler-apple.html>].

17. Martin Untersinger, 2015, *Dino*, le nouveau programme-espion développé par des francophones, Le Monde.fr [https://www.lemonde.fr/pixels/article/2015/06/30/dino-le-nouveau-programme-espion-developpe-par-des-francophones_4664675_4408996.html].

18. Ce conseil vaut tout autant pour les personnes utilisant GNU/Linux. En décembre 2009, le site gnome-look.org a diffusé un *malware* [<https://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.

19. Spiegel, 2013, *Interactive Graphic : The NSA's Spy Catalog* [<https://www.spiegel.de/international/world/a-941262.html>] (en anglais).

Sans en faire un tour exhaustif, on peut découvrir pêle-mêle dans ce catalogue de faux connecteurs USB, permettant de retransmettre sous forme d'ondes radio ce qui transite par ces mêmes connecteurs, des minuscules puces installées dans les câbles reliant écran ou clavier à l'ordinateur et faisant de même, pour qu'un adversaire puisse capter ce qu'on tape ou voit tout en étant à bonne distance. Enfin, pléthore de matériel espion installé dans l'ordinateur, que ce soit au niveau du disque dur, du BIOS, etc.

Le tableau n'est pas très encourageant, un véritable audit de son ordinateur demanderait de démonter celui-ci avec très peu de chance de le remonter de telle manière qu'il puisse fonctionner de nouveau. Cela dit, ce matériel n'est pas à la disposition de tous types d'adversaires. De plus, rien n'indique que l'usage d'un tel matériel est devenu monnaie courante, que ce soit pour des raisons de coût, d'installation, ou autres paramètres.

Nous allons quand même nous attarder un peu sur le cas des *keyloggers* qui peuvent entrer à la fois dans la catégorie du matériel espion et des logiciels malveillants.

3.4 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 installés²⁰.

Une fois mis en place, leur capacité à enregistrer touche par touche ce qui est tapé sur un clavier contourne donc tout dispositif de chiffrement, et permet d'avoir directement accès aux phrases, mots de passe et autres données sensibles entrées.

[page 39]

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 clavier²¹. 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 cas²². 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 15]

Les *keyloggers* logiciels sont beaucoup plus répandus, parce qu'ils peuvent être installés à distance (*via* un réseau, par le biais d'un logiciel malveillant, ou autre), 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* logiciels²³.

20. security.resist.ca, 2014, *Keystroke Loggers & Backdoors* [<http://security.resist.ca/keylog.shtml>] (en anglais).

21. Pour ce faire une idée, nombreux modèles sont en vente libre [<http://www.google.com/products?q=keyloggers>] pour une somme allant de 40 à 100 \$.

22. ZDNet Australia, 2007, *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>] (en anglais).

23. En 2000, l'usage d'un *keylogger* a permis au FBI [https://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 (en anglais).

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. Même si le catalogue de la NSA publié fin 2013 rend compte de la difficulté de trouver soi-même des dispositifs d'enregistrement de frappe à peine plus gros qu'un ongle. Pour les *keyloggers* logiciels, les pistes sont les mêmes que pour les autres *logiciels malveillants*.

3.5 Des problèmes d'impression ?

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

3.5.1 Un peu de stéganographie

Première chose à savoir : de nombreuses imprimantes haut de gamme signent leur travail²⁴. Cette signature stéganographique²⁵, baptisée *watermarking*, repose sur de très légers détails 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 : retrouver la machine à partir de ses travaux.

C'est d'ailleurs notamment ainsi que la personne ayant diffusé en juin 2017 des documents *top secret* de la NSA sur le piratage des élections des États-Unis de 2016 par des hackers russes a été retrouvée. Des marques de l'imprimante utilisée pour imprimer les documents confidentiels étaient toujours présentes lors de leur publication par le journal *The Intercept*²⁶.

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 photocopies 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...

3.5.2 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 jusqu'à ce qu'un autre document les recouvre.

24. L' *Electronic Frontier Foundation* tente de maintenir une liste des constructeurs et de ces modèles d'imprimantes indiscrets [<https://www.eff.org/pages/list-printers-which-do-or-do-not-display-tracking-dots>] (en anglais).

25. Pour en savoir plus sur la stéganographie, nous conseillons la lecture de cet article Wikipédia, 2014, *Stéganographie* [<https://fr.wikipedia.org/wiki/St%C3%A9ganographie>].

26. Robert Graham, 2017, *How The Intercept Outed Reality Winner* [<http://blog.erratasec.com/2017/06/how-intercept-outed-reality-winner.html>] (en anglais).

La plupart des imprimantes laser 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.

[page 12]

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.

4.1.1 La métaphore du gâteau

Pour comprendre la différence entre logiciels libres et propriétaires, 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 qu'il va advenir de nos données.

4.1.2 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 grappiller 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 les personnes 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.

4.1.3 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 la fabriquer. Gardons en tête quand même qu'il n'y a rien de magique : les logiciels libres ne jettent 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. Cela implique trop souvent des savoir-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 discréction 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, dont le développement s'est arrêté en mai 2014. Il s'agissait d'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 est libre), exprimées en nombre de lignes de code source modifiées, sur une courte période de temps² :

1. Au sujet des « portes dérobées » voir l'article Wikipédia, 2014, *Porte dérobée* [https://fr.wikipedia.org/wiki/Porte_d%C3%A9rob%C3%A9e].

2. Linux Weekly News, 2017, *Some 4.10 Development statistics* [<https://lwn.net/Articles/713803/>] (en anglais).

Organisation	Pourcentage
Intel	20.4%
AMD	8.7%
Samsung	6.6%
Red Hat	4.8%
(Inconnu)	4.0%
Linaro	3.8%
SUSE	3.6%
(Aucun)	3.2%
IBM 3.0%	
(Consultant)	3.0%
Solarflare Comm.	2.3%
MediaTek	1.8%
Cavium	1.8%
<i>etc.</i>	

Alors... il n'est pas impossible qu'une personne qui a 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. Ce fut d'ailleurs le cas avec la faille connue sous le nom d'*Heartbleed*³. 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 : *libre 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 comptes d'utilisateurs ou d'utilisatrices sur un même ordinateur. Il faut bien savoir que les mots de passe qui protègent parfois ces comptes 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 16]

3. Wikipédia, 2014, *Heartbleed* [<https://fr.wikipedia.org/wiki/Heartbleed>].

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

[page 17] 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.

4.3.1 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 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 permettant de « récupérer » ou de « restaurer » des données⁴.

4.3.2 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 difficile 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 1,05⁵... 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 plusieurs fois par-dessus des données aléatoires. La meilleure façon de rendre inaccessible le contenu de ces fichiers « supprimés » est donc d'utiliser des logiciels qui s'assureront de réécrire plusieurs fois par-dessus. C'est ce qu'on appelle « écraser des données » (*wipe* en anglais).

4.3.3 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 actuels réorganisent leur contenu « intelligemment » : une partie du disque est réservée pour remplacer des endroits qui deviendraient défectueux. Ces opérations de remplacement sont difficilement détectables, et on ne peut jamais vraiment s'assurer que l'endroit sur lequel on réécrit est bien celui où le fichier a été écrit initialement...

4. C'est le cas de PhotoRec [http://www.cgsecurity.org/wiki/PhotoRec_FR].

5. Peter Gutmann, 1996, *Secure Deletion of Data from Magnetic and Solid-State Memory* [https://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html] (en anglais).

Pour les clés USB et les disques SSD (*Solid State Drive*), il est même sûr que dans la plupart des cas on réécrit à un endroit différent. La mémoire *flash*, utilisée par les clés USB et les disques SSD, arrête de fonctionner correctement après un certain nombre d'écritures⁶, et 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 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.

Pour les puces de mémoire *flash* d'une clé USB ou d'un disque SSD, même si ce n'est pas non plus immédiat, l'opération est beaucoup plus simple : il suffit d'un fer à souder, et d'un appareil permettant de lire directement les puces de mémoire. Ces derniers se trouvent pour environ 1 500 dollars⁷.

Les systèmes de fichiers « intelligents »

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

[page 17]

Les systèmes de fichiers développés ces dernières années, comme NTFS ou *ext4*, 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, *ext4*, peut fonctionner avec plusieurs modes. Celui le plus couramment utilisé ne met dans le journal que les noms des fichiers et d'autres métadonnées, 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 compressés⁸ ; etc.

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

[page 19]

6. Wikipédia, 2014, *Solid-state drive* [https://fr.wikipedia.org/wiki/Solid_State_Drive].

7. Le PC-3000 Flash SSD Edition [<http://www.pc-3000flash.com/>] est vendu comme un *outil professionnel de recouvrement de données sur des périphériques flash endommagés* (lien en anglais).

8. Page de manuel de *shred(1)* [<http://manpages.debian.net/cgi-bin/man.cgi?query=shred&locale=fr>].

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. Les recommandations actuelles sont donc de détruire méthodiquement les supports de ce type qui auraient pu contenir des données à faire disparaître⁹.

4.3.4 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.

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. De même que 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 »...¹⁰

4.3.5 Et pour ne laisser aucune trace ?

Malheureusement, il n'y a pas de méthode simple pour régler radicalement le problème. 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, comme *Tails*. Alors, il est possible de ne rien écrire sur le disque dur ni sur le *swap*, et de ne garder les informations que dans la mémoire vive (donc uniquement tant que l'ordinateur reste allumé).

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 les logiciels « non portables ». Pire, le discours faisant leur promotion participe à créer une illusion de sécurité avec d'énormes bêtises comme « *l'utilisation*

9. NIST, 2014, *Guidelines for Media Sanitization* [<http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-88r1.pdf>].

10. PhotoRec [http://www.cgsecurity.org/wiki/PhotoRec_FR] propose aussi ce genre de fonctionnalités.

des logiciels se fait de façon sécurisée et sans laisser d'informations personnelles sur les machines sur lesquelles vous utilisez votre FramaKey »¹¹. C'est malheureusement faux.

4.4.1 Principaux problèmes

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

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]

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.

¹¹. Cet extrait provient de la page d'accueil du site web de FramaKey [<https://www.framakey.org/>], une compilation de logiciels portables réalisée par Framasoft [<http://www.framasoft.net/>], un site français de promotion du logiciel libre.

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 numéro de carte bleue.

La *cryptanalyse* est le domaine consistant à « casser » les techniques cryptographiques, par exemple pour permettre de retrouver un message qui avait été protégé¹.

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. Une personne écrivant 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 *confidential*).

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 stéganographiques (comme celles utilisées par les imprimantes évoquées auparavant), dont nous ne parlerons pas ici.

[page 28]

5.1 Protéger des données des regards indiscrets

Comme l'ont bien compris les enfants utilisant des codes pour s'échanger des mots ou les 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 Wikipédia, 2014, *Cryptanalyse* [<https://fr.wikipedia.org/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.

5.1.1 Comment ça marche ?

Grosso modo, il y a seulement trois grandes idées pour comprendre comment on peut chiffrer des messages².

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

$$\boxed{A + 3 \text{ lettres} = D}$$

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 :

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

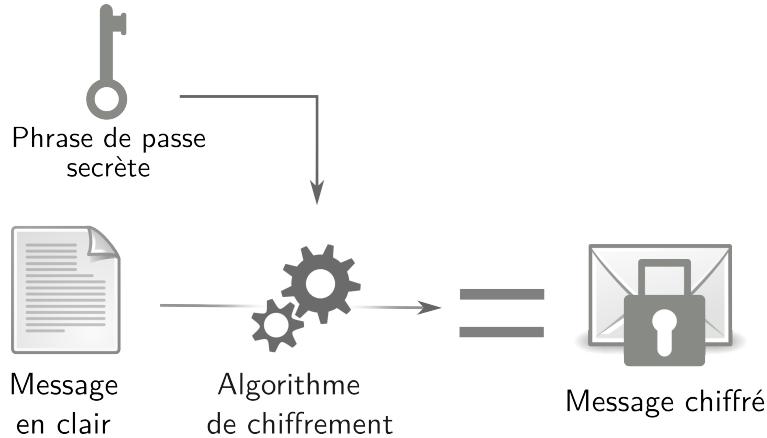
Ce que l'on appelle un *algorithme de chiffrement*, ce sont les différentes techniques utilisées pour transformer le texte original. Quant à la *clé* de chiffrement, c'est, par exemple dans le cas du chiffre de César, le nombre de caractères de décalage (3 en l'occurrence), ou dans la technique de diffusion, le nombre de lignes des colonnes. La valeur de cette clé est variable, on aurait tout aussi bien décider de faire des colonnes de 2 lignes, ou un décalage de 6 caractères.

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 ou 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.

5.1.2 Vous voulez un dessin ?

Concrètement, pour assurer la *confidentialité* de nos données, on utilise deux opérations : le chiffrement, puis le déchiffrement.



Première étape : le chiffrement

Pour un exemple d'usage pratique, prenons le message suivant³ :

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-----
jAOECQMCRM01mTSIONRg01kBWGQI76cQ0ocEvdBhX6BM2AU6aYSPYymSqj8ihFXu
wV1GVraWuwEt4XnLc3F+0xT3EaXINMHdH9oydA92WDkaqPEnjsWQs/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.

Deuxième étape : le déchiffrement

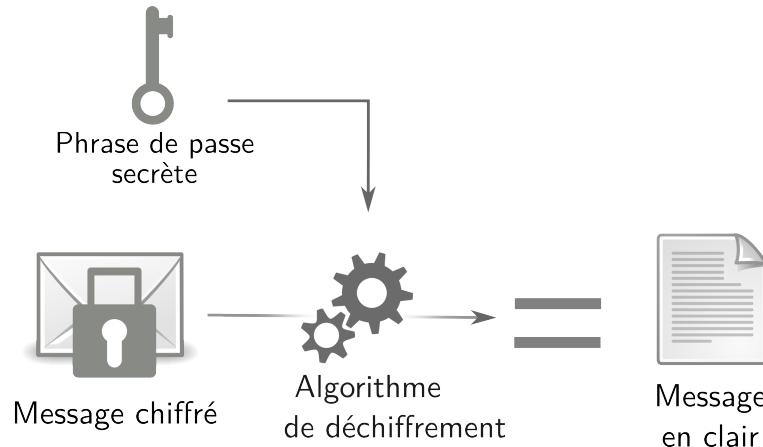
Pour le déchiffrement, il nous suffira d'utiliser de nouveau GnuPG, avec notre texte chiffré, cette fois. GnuPG nous demandera la phrase de passe ayant servi à chiffrer notre texte, et si cette dernière est correcte, on obtiendra enfin l'information qui nous manquait pour préparer le déjeuner.

5.1.3 Pour un disque dur...

Si on souhaite mettre sur un support de stockage (disque dur, clé USB, *etc.*) uniquement des données chiffrées, il va falloir que le système d'exploitation se charge de réaliser « à la volée » les opérations de chiffrement et de déchiffrement.

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>] (en anglais).

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.



Ainsi, chaque fois que des données devront être 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 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.

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 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.

5.1.4 Résumé et limites

La cryptographie permet donc de bien protéger ses données⁵, en chiffrant tout ou partie de son disque dur comme de tout autre support de stockage (clé USB, CD, etc.), ou de ses communications. De plus, les ordinateurs modernes sont suffisamment puissants pour que nous puissions faire du chiffrement une routine, plutôt que de le réservier 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.

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. Un article de rue89 sur les révélations de Snowden quant à l'impuissance de la NSA face au chiffrement Marie Gutlub, 2014, *Crimes de guerre et décryptage de données : nouvelles révélations de Snowden* [<http://tempsreel.nouvelobs.com/rue89/rue89-monde/20141229.RUE7224/crimes-de-guerre-et-decryptage-de-donnees-nouvelles-revelations-de-snowden.html>].

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 cela 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 rapidement⁶; ainsi les disques chiffrés deviennent 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 27]

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 pénal⁷ :

Est puni de trois ans d'emprisonnement et de 270 000 € 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 oeuvre, 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 oeuvre 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 450 000 € 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. Une personne a été mise en examen en 2016 en France pour « refus de remettre aux autorités

6. 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.

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

judiciaires ou de mettre en œuvre la convention secrète de déchiffrement d'un moyen de cryptologie »⁸, mais l'affaire n'a pas encore été jugée. De l'autre côté de la Manche une législation douanière similaire fait planer la possibilité de la prison ferme pour Muhammad Rabbani, directeur de l'organisation CAGE, pour avoir refusé de livrer ses mots de passe à la frontière⁹.

Cependant certains avocats¹⁰ pensent que cette loi s'appliquerait aux tiers et non à la personne intéressée en premier lieu, en vertu du droit à ne pas s'auto-incriminer¹¹.

Depuis 2014¹², les flics ont aussi le droit de réquisitionner n'importe qui susceptible « d'avoir connaissance des mesures appliquées pour protéger les données » et « de leur remettre les informations permettant d'accéder aux données »¹³.

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 par 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 complexe de s'assurer que les logiciels que l'on souhaite installer sur nos ordinateurs n'aient pas été modifiés en route afin d'y introduire des logiciels malveillants.

5.2.1 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 :

- 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.

8. Florian Reynaud et Soren Seelow, 2016, *Alertes à la bombe dans les lycées : le jeune homme placé sous le statut de témoin assisté*, Le Monde [http://www.lemonde.fr/pixels/article/2016/02/10/alerte-a-la-bombe-dans-les-lycees-un-jeune-homme-presente-a-un-juge_4862662_4408996.html].

9. Ewen MacAskill, 2017, *Campaign group to challenge UK over surrender of passwords at border control*, The Guardian [<https://www.theguardian.com/politics/2017/may/14/campaign-group-to-challenge-uk-over-surrender-of-passwords-at-border-control>] (en anglais).

10. Maître Éolas, 2014, *Allô oui j'écoute* [<http://www.maître-eolas.fr/post/2014/03/08/All%C3%A9-Oui-J-%C3%A9coute#c173067>].

11. Liberté, Libertés chéries, 2015, *Principe de loyauté et droit de ne pas contribuer à sa propre incrimination* [<https://libertescherries.blogspot.ro/2015/03/principe-de-loyaute-et-droit-de-ne-pas.html>].

12. République française, 2014, *loi n° 2014-1353 du 13 novembre 2014 renforçant les dispositions relatives à la lutte contre le terrorisme* [<https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000029754374>].

13. République française, *Code de procédure pénale*, article 57-1 [<https://www.legifrance.gouv.fr/affichCodeArticle.do?idArticle=LEGITEXT000032655328&cidTexte=LEGITEXT00006071154>].

Lorsque ces propriétés sont réunies, il nous suffit alors de comparer les petits morceaux issus de deux objets pour savoir s'il étaient identiques.

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, notamment la fonction SHA1¹⁴.

5.2.2 Vérifier l'intégrité d'un logiciel

Prenons un exemple : Alice a écrit un programme et le distribue sur des CD, que l'on peut trouver dans des clubs d'utilisateurs de GNU/Linux. Betty 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.

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

CD d'Alice	————→	94d93910609f65475a189d178ca6a45f SHA256 22b50c95416affb1d8feb125dc3069d0
------------	-------	---

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

CD de Betty	————→	94d93910609f65475a189d178ca6a45f SHA256 22b50c95416affb1d8feb125dc3069d0
-------------	-------	---

Comme les nombres sont les mêmes, Betty est contente, elle est sûre de bien utiliser le même CD que celui fourni par Alice.

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

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

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.

[page précédente]

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 Betty vérifiera la somme de contrôle du CD qui lui permettra d'installer la nouvelle version.

¹⁴. Marc Stevens et Al., 2017, *Announcing the first SHA1 collision*, Google Security Blog [<https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html>].

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

Malheureusement pour elle, et heureusement pour Betty, même avec de nombreux ordinateurs puissants, les chances de réussite de Carole 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, c'est à dire pouvoir vérifier leur *authenticité*...

5.2.3 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/Linux¹⁵, il faut que l'ordinateur puisse vérifier que le mot de passe que l'on entre est le bon. Cependant les mots de passe ne sont pas enregistrés en clair sur l'ordinateur, sinon il serait trop facile de les obtenir.

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

Lorsque l'on choisit 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 cryptographie *symétrique*.

Ceci en opposition avec la cryptographie *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 *second tome*.

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 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...

[tome 2 ch. 6]

15. Rappelons-nous que ces mots de passe ne servent pas à protéger les données [page 33] !

00	0010		0110	0010
0000	0001	011	00010	0111
101		01010		111
101000	011		101010011	1101
000111011	100		101110011	1
101111101	00000		1011011	1100
000001	01		0111	001

```

00111    11000
00101    11001111
10010    0011111
0100111000110100
11101000001100
11111111000111
11000000

```

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 ».

Ou au contraire, on se demande avec découragement si tout compte fait, on a vraiment quelque chose à cacher... une question qu'on a déjà écartée dans la préface.

Alors avant de retourner à la trappe cachée sous le tapis du salon, 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.

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 qu'on ait l'assurance qu'elles fonctionnent réellement ; *etc.*

On devrait donc commencer par se poser quelques questions simples, afin d'établir un *modèle de menace*¹.

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, il est difficile de ménager la chèvre et le choux...

6.2 Contre qui veut-on se protéger ?

Rapidement, se pose la question des capacités des personnes qui en auraient après ce que l'on veut protéger. Et là, ça se corse, déjà parce qu'il n'est pas facile de savoir ce que les plus qualifiées d'entre elles peuvent réellement faire, de quels moyens et de quels budgets elles bénéficient. En suivant l'actualité, et par divers autres biais,

1. Voir Electronic Frontier Foundation, 2015, *Une Introduction au Modèle de Menace* [<https://ssd.eff.org/fr/module/une-introduction-au-mod%C3%A8le-de-menace>].

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 39] 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 DGSE², 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'avoir la certitude de ce que ces entités peuvent faire, mais en même temps leur champ d'intervention est limité, et il y a peu de cas dans 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 elles peuvent intéresser. À partir de là, on peut avoir une vision de quels moyens sont à leur disposition (ou en tout cas, dans la mesure du possible, essayer de se renseigner) et en conséquence, définir une *politique de sécurité* adaptée.

2. Wikipédia, 2017, *Direction générale de la Sécurité extérieure* [https://fr.wikipedia.org/wiki/Direction_g%C3%A9n%C3%A9rale_de_la_S%C3%A9curit%C3%A9_ext%C3%A9rieure].

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 39]
[page 19]

Ces exemples nous apprennent quelque chose : de telles « solutions » ciblées ne sont daucune 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 51]

De façon globale, mais *située* : à une situation donnée correspond un ensemble singulier d'enjeux, de risques, de savoir-faire... et donc de possibilités d'action. Il n'existe pas de solution miracle convenant à tout le monde, et qui réglerait tous les problèmes d'un coup de baguette magique. La seule voie praticable, c'est d'en apprendre suffisamment pour être capable 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 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 et d'outils me protégeraient de façon suffisante contre les risques évalués précédemment ?

Pour ce faire, on peut partir de nos pratiques actuelles et se poser les questions suivantes :

1. Face à une telle politique de sécurité, quels angles d'attaque mes adversaires utiliseraient ?
2. Quels moyens devraient être mis en œuvre par mes adversaires ?
3. Ces moyens sont-ils à la portée de mes 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-même 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...

7.3.1 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...

7.3.2 Liste autorisée, liste bloquée

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 bloquée¹ : 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 bloquée fonctionne sur la base de la *confiance-sauf-dans-certains-cas*.

Le principe de la liste autorisée² 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 117]

Voilà pour les principes de base.

Toute politique de sécurité basée sur le principe de la *liste bloquée* 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 bloquée ; 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 bloquée* sont légion (comme nous allons le voir), au contraire de ceux s'appuyant sur la méthode *liste autorisée*, qui nous est donc, sans doute, moins familière.

Mettre en œuvre l'approche *liste autorisée* 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 bloquée*.

[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 bloquée, leurs bases de données doivent perpétuellement être mises à jour, systématiquement en retard. Une réponse à ce problème, avec l'approche liste autorisée, 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 *autorisées*, 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 autorisées 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 autorisée adéquate n'existe pas, sur des listes bloquées solides, de provenance connue, en gardant en tête le problème intrinsèque à cette méthode ; listes bloquées qu'on aidera éventuellement à compléter, en partageant nos découvertes.

1. parfois aussi appelée « liste noire ».

2. parfois aussi appelée « liste blanche ».

7.3.3 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 17] 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.

7.3.4 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.

[page 161]

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.

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 ;
2. le système d'exploitation est un logiciel propriétaire (exemple : Windows), et truffé de logiciels malveillants.

[page 19]

[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 précisément menaçant, selon la définition européenne du terrorisme¹, « 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 d'Orange 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

8.2.1 Que veut-on protéger ?

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

[page 51]

1. Union Européenne, 2017, Directive (UE) 2017/541 du Parlement européen et du Conseil du 15 mars 2017 relative à la lutte contre le terrorisme et remplaçant la décision-cadre 2002/475/JAI du Conseil et modifiant la décision 2005/671/JAI du Conseil, article 3 [<http://eur-lex.europa.eu/legal-content/FR/TXT/?uri=CELEX:32017L0541&qid=1495634630652>].

- 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.

8.2.2 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 enfant 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 enfant 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.

8.3.1 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.

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

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 18]
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. Il faudra également veiller à mettre régulièrement à jour le système, afin de combler les « trous de sécurité » que pourraient utiliser des logiciels malveillants. [page 24]

Pour effectuer ces étapes, se référer aux recettes suivantes :

- chiffrer un disque externe ou une clé USB, voir page 133 ;
- utiliser un système *live*, voir page 101 ;
- sauvegarder des données, voir page 139 ;
- effacer « pour de vrai », voir page 127 ;
- installer un système chiffré, voir page 107 ;
- garder un système à jour, voir page 161.

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

8.3.2 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 53]

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 :

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

8.3.3 Troisième étape : la loi comme moyen de coercition

1. Angle d'attaque : la police a le droit d'exiger que vous lui donnez accès aux informations chiffrées, comme expliqué dans le chapitre consacré à la cryptographie. [page 39]
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.

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'accèsibilité, 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.

Ensuite, c'est toute la question de la compartmentation 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 67] Le prochain cas d'usage étudiera de tels traitements de faveur, mais patience, mieux vaut pour l'instant terminer la lecture de celui-ci !

8.3.4 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.

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

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

[page 16] Mais si c'est sa mise en œuvre dans le noyau Linux qui est cassée, ça ne passera pas dans *Libération*, 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 Debian². 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...

2. La liste de diffusion se nomme *debian-security-announce* [<https://lists.debian.org/debian-security-announce/>].

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.

8.3.6 Angle d'attaque : *cold boot attack*

1. Angle d'attaque : la *cold boot attack* est décrite dans le chapitre consacré aux traces.
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 est donc très faible.

[-----]
page 19
[-----]

Il peut sembler superflu de se protéger contre cette 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 quelquesunes 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) ;
- rendre l'accès au compartiment de votre ordinateur contenant la RAM plus long et difficile, par exemple en le collant/soudant.

8.3.7 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 très compliquée la mémorisation « à la volée » par un observateur humain ;
- vérifier autour de soi, à la recherche d'éventuels yeux (humains ou électroniques) indésirables, avant de taper sa phrase de passe.
- cacher son clavier à l'aide de l'écran dans le cas d'un ordinateur portable, ou à l'aide d'un tissu³ (manteau, cape).

[-----]
page 93
[-----]

³. Dans Citizen Four, on peut voir Edward Snowden mettre un tissu par dessus lui et son ordinateur pour taper sa phrase de passe.

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

[page 107] 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 chiffrement 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, 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.

[page 23] 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...

Une protection 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 développerons 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 micrologiciel 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 praticable de s'en protéger.

8.3.9 Angle d'attaque : les logiciels malveillants

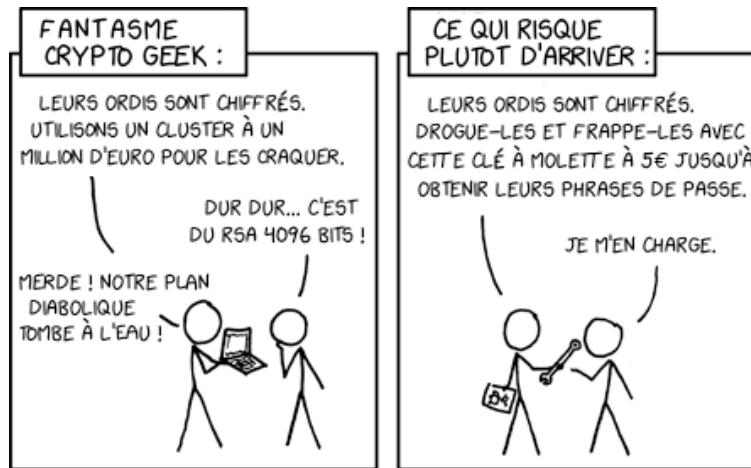
[page 23] Nous avons appris dans un chapitre précédent que des logiciels installés à notre insu sur un ordinateur peuvent nous dérober des données. Dans le cas présent, un tel logiciel est en mesure de transmettre la clé de chiffrement du disque dur à un adversaire... qui obtiendra ensuite, grâce à cette clé, l'accès aux données chiffrées, quand il aura un 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 de faire preuve d'une extrême prudence 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 119] 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, consacré aux réseaux, et à Internet en particulier, prolonge cette étude.

8.3.10 Angle d'attaque : la force brute

Attaquer un système cryptographique par « force brute », c'est-à-dire chercher la phrase de passe en testant une à une toutes les combinaisons possibles, est la plus simple, la plus stupide, et la plus lente des manières. Mais quand on ne peut pas mettre en œuvre un autre type d'attaque...



Dessin issu de XKCD, traduit par nos soins (<https://xkcd.com/538/>).

Pour le disque dur chiffré lors de la première étape, ça demande énormément de temps (de nombreuses années) et/ou énormément d'argent et des compétences pointues... du moins si la phrase de passe est solide.

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. Notamment celle d'aller demander directement la phrase de passe à la personne concernée, que ce soit de façon cordiale ou non...

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 59]
[page 107]

- 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.

9.1.1 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 maintenant de définir les risques auxquels exposent les pratiques décrites dans ce cas d'usage.

9.2.1 Que veut-on protéger ?

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

- 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 (glacée).

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.

9.2.2 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.

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 proposons tout de même un chemin praticable qui permet de limiter la casse. Allons donc voir à quoi ressemble ce chemin, en ignorant les paragraphes suivants, qui sont consacrés à GNU/Linux.

9.4 Le système *live* amnésique

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.

9.4.1 Liste bloquée vs. liste autorisée

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 bloquée », 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.

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

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.

9.4.2 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 :

- installer le logiciel dans le système *live* au début de chaque session de travail ;
- créer une clé *live* incluant notre logiciel dans son volume persistant ;
- faire du lobbying auprès des auteurs du système *live* pour qu'ils y ajoutent le logiciel souhaité ;

L'utilisation d'un système *live* 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.

À noter qu'il est possible d'installer une Debian dans une machine virtuelle, afin de satisfaire des besoins similaires mais cette solution est assez complexe, et ne sera donc pas documentée ici.

[page 55]

[page 101]

[page suivante]

[page 152]

9.5 Travailler sur un document sensible... sur un système *live*

[page 67] 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.

9.5.1 Télécharger et installer le système *live*

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

Si l'on ne dispose pas encore d'une copie de la dernière version du système live *Tails*, suivre la recette télécharger et installer un système *live* « discret », voir page 102.

À partir du premier périphérique *Tails* ainsi créé, nous allons créer une clé USB dédiée à notre projet. Pour cela, se munir du système *live* précédemment installé et le démarrer, voir page 95. Suivre ensuite la recette cloner une clé *Tails*, voir page 103. Suivre alors créer et configurer un volume persistant dans *Tails*, voir page 104. Activer uniquement l'option *Données personnelles*.

9.5.2 Installer un éventuel logiciel additionnel

Si l'on a besoin d'utiliser un logiciel qui n'est pas installé dans *Tails* et que l'on ne veut pas le réinstaller à chaque fois, suivre la recette installer un logiciel additionnel persistant dans *Tails*, voir page 104.

9.5.3 Utiliser le système *live*

Chaque fois que l'on souhaite travailler sur notre document, il suffira de menir le la clé contenant notre système *live* et sa persistance chiffrée pour démarrer dessus, voir page 95. On devra alors activer le volume persistant, voir page 104.

9.5.4 Supprimer le système *live*

[tome 2 ch. 9] Une fois notre projet terminé et imprimé ou publié en ligne, on peut éventuellement archiver le projet, voir page 79. Il nous faut ensuite supprimer le volume persistant, voir page 104.

9.5.5 Limites

[plus bas] Certaines limites, communes à cette méthode et à celle basée sur l'usage de Windows, sont exposées plus loin.

[page 77]

9.6 Travailler sur un document sensible... sous *Windows*

[page 67] 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.

9.6.1 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 bloquée, dont l'inefficacité a déjà été expliquée précédemment.

[page 53]

Il est maintenant temps de passer aux choses sérieuses.

9.6.2 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 autorisée* : 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*¹ 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.

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.

1. Pour plus d'informations, voir la page Wikipédia, 2014, *Virtualisation* [<https://fr.wikipedia.org/wiki/Virtualisation>].

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 stocké sur le disque dur de notre système Debian chiffré.

Installer le *Gestionnaire de machine virtuelle*

La recette « installer le *Gestionnaire de machine virtuelle*, voir page 152 » explique comment installer le logiciel *Gestionnaire de machine virtuelle*, qui nous servira à lancer Windows dans un compartiment étanche.

Installer un Windows « propre » dans le *Gestionnaire de machine virtuelle*

Préparons une image de disque virtuel *propre* : la recette « installer un Windows virtualisé, voir page 153 » explique comment installer Windows dans le *Gestionnaire de machine virtuelle* en lui coupant, dès le départ, tout accès au réseau.

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*² nécessaire à la réalisation des œuvres prémeditées : 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 mélangons 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*; la recette « Partager un dossier avec un système virtualisé, voir page 158 » explique comment procéder pratiquement.

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.

Prendre un instantané du Windows « propre »

Prenons maintenant un *instantané* de la machine virtuelle *propre* qui vient d'être préparée. C'est-à-dire : sauvegardons son état dans un coin. Par la suite, cet instantané servira de base de départ pour chaque nouveau projet.

La recette prendre un instantané d'une machine virtuelle, voir page 156 explique comment effectuer cette opération.

2. S'il est nécessaire de cacher qu'on fabrique des films, avoir des logiciels de montage vidéo peut être compromettant, parce qu'il serait plus difficile de nier cette activité, si cela s'avérait nécessaire.

Nouveau projet, nouveau départ

Mettons qu'un nouveau projet nécessitant l'utilisation de Windows débute ; voici la marche à suivre :

1. on restaure l'instantané de la machine virtuelle contenant l'installation de Windows propre ;
2. la machine virtuelle peut maintenant être démarrée dans son compartiment étanche ; elle servira **exclusivement** pour le nouveau projet, et devient désormais une machine virtuelle *sale* ;
3. au sein de cette nouvelle machine virtuelle *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 métadonnées des fichiers qu'ils enregistrent, et qu'il vaut mieux éviter de rendre possibles de fâcheux recoupements.

[page 22]

La recette restaurer l'état d'une machine virtuelle à partir d'un instantané, voir page 157 explique les détails techniques de la première étape. En ce qui concerne la création d'un nouvel utilisateur sur la version de Windows utilisée, la personne lisant ces pages est une fois encore certainement à même de la trouver du côté du *Panneau de configuration*.

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, eh 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. Moins on donne le droit à Windows d'écrire ici ou là, moins il laissera de traces gênantes.

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, voir page 158 » explique comment procéder pratiquement.

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.

Pour cela, on va 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 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.

Si l'on a besoin de graver un DVD, ou pourra ensuite le faire à partir de Debian.

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 « partager un dossier avec un système virtualisé, voir page 158 » et « chiffrer une clé USB, voir page 133 » expliquent comment procéder pratiquement.

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).

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

- l'image de disque est restaurée à son état « Propre » grâce à la recette « Restaurer l'état d'une machine virtuelle à partir d'un instantané », voir page 157 ;
- après avoir vérifié, une dernière fois, que tout ce qui doit être conservé a bien été archivé ailleurs, les dossiers partagés avec Windows sont effacés « pour de vrai », voir page 129 ;
- les traces laissées sur le disque dur sont effacées « pour de vrai », voir page 132 ;

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 ».

9.6.3 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 les attaques praticables contre ce système.

[page 60]

Traces laissées sur notre Debian chiffrée

La plupart des traces **les plus évidentes** de ce projet sont séparées du reste du système : tous les fichiers de travail sont stockées dans le fichier contenant l'image de disque virtuel. Le nom de la machine virtuelle, sa configuration ainsi que ses périodes d'utilisation laisseront par contre d'autres traces sur notre système Debian.

Si la catastrophe arrive pendant la réalisation du projet Le disque dur de l'ordinateur utilisé contient les fichiers de travail à l'intérieur de l'image de disque virtuel.

Si la catastrophe arrive plus tard L'image de disque virtuel étant convenablement nettoyée 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.

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 bloquée, 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 connaît bien désormais : journaux, mémoires vive et « virtuelle », sauvegardes automatiques.

[page 67]

[page 55]

[page 19]

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 77]

Sinon, creusons un peu.

Aller plus loin

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 61]

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

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 bloquée », avec tous les problèmes que ça pose.

[page 55]

Les 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 : le *Gestionnaire de machine virtuelle*.
[page 19] Comme le chapitre « traces à tous les étages » l'explique, diverses traces subsisteront donc, inévitablement, sur le disque dur **interne** de l'ordinateur utilisé.

Pour suivre cette piste :

- [page suivante]
[page 133]
- 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.

Utiliser un système *live* comme système *hôte* Le pendant de cette approche « liste bloquée » est une solution de type « liste autorisée », 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 :

- [page suivante]
[page 133]
[page 101]
- 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*.

9.7 Nettoyer les métadonnées du document terminé

Une fois notre document terminé, on l'exportera dans un format adapté à l'échange de documents – par exemple un PDF pour imprimer un texte, un fichier AVI ou Ogg pour publier une vidéo sur Internet, *etc.*

Considérons qu'on publie notre document sans prendre de plus amples précautions : un adversaire à qui il déplairait va probablement tout simplement commencer par télécharger le document en quête d'éventuelles métadonnées qui le rapprocheraient de ses auteurs.

Malgré les précautions qu'on a déjà prises, il est bon de nettoyer les éventuelles métadonnées présentes.

9.8 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 *live* ou sur l'envoûtement de l'infâme Windows.

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.

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 167]

[page 59]

[page 7]

[page 15]

[page 23]

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, [page 67] un film a été monté, compressé, et gravé sur DVD.

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

10.3.1 Que veut-on protéger ?

Que donnent les catégories définies lorsque nous parlions d'*évaluation des risques*, [page 51] appliquées à ce cas ?

- 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 meilleure confidentialité.

10.3.2 Contre qui veut-on se protéger ?

Les risques envisagés dans notre « nouveau départ » sont valables ici aussi : un cam- [page 59]

briolage, 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 inopportunne 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.

10.4 Méthode

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

1. créer une clé USB ou un disque dur externe chiffré, voir page 133 ;
2. copier les fichiers à archiver vers ce périphérique ;
3. supprimer et écraser le contenu des fichiers de travail, voir page 127.

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 ?

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 à peu près impossible l'accès aux données.

Face à ce problème, on peut envisager quelques pistes.

10.5.1 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.

10.5.2 Utiliser la même phrase de passe que pour son système quotidien

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 révéler 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 lesquels on accédera 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.

10.5.3 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 145]

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.

```

00110 01000
00101 011001111
000100 0011111
0010011000 010100
0111010000 01000
011111100 00111
011000 0001

```

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 49]

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 9.0 (surnommée Stretch)¹ et Tails² (*The Amnesic Incognito Live System*).

Pour autant, ces recettes sont généralement concoctables avec d'autres distributions basées sur Debian, telles qu'Ubuntu³ ou Linux Mint⁴.

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 59]

[page 101]

1. <https://www.debian.org/releases/stretch/>

2. <https://tails.boum.org/>

3. <https://www.ubuntu-fr.org/>

4. <https://www.linuxmint.com/>

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.

Enfin l'exemplaire que vous avez entre les mains n'est pas forcément à jour aux vues des versions actuelles des différents outils impliqués. La version en ligne [<https://guide.boum.org/>] du Guide d'Autodéfense Numérique a des chances d'être plus à jour.

Utiliser un terminal

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

 Durée : 15 à 30 minutes.

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 rôle¹. 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 3, il faut d'ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper `term` et cliquer sur *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.

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

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 (ou `return`), le résultat est qu'on ouvre un éditeur de texte. On aurait pu faire exactement la même chose en appuyant sur la touche (sur un Mac) et en tapant `texte` puis en cliquant sur `gedit`. Par contre, on ne pourra pas entrer de nouvelle commande dans notre terminal tant que l'on n'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.

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.

11.2.1 Syntaxe

Prenons par exemple cette commande, `sfill`, qui permet à peu près les mêmes opérations que `nautilus-wipe`, un outil graphique qui sera présenté plus tard :

commande option option argument

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 séparé des autres par un (ou plusieurs) espace(s). 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 connaître les options et les arguments d'une commande, pas de mystère : chacune dispose normalement d'une page de manuel. Pour y accéder, il suffit de taper dans le *Terminal* `man` suivi du nom de la commande, puis d'appuyer sur la touche (ou `return`). Ces dernières peuvent toutefois être difficiles à comprendre par leur aspect technique, et ne sont parfois disponibles qu'en anglais.

11.2.2 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.

11.2.3 Exécution

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

11.2.4 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 enfoncee appuyer sur la touche **[C]**. On arrête alors la commande immédiatement, un peu comme quand on ferme la fenêtre d'un programme.

11.2.5 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 la touche **[4]**...

D'autres symboles sont rarement utilisés en dehors du terminal, mais sont disponibles avec les claviers standards. Ils sont même indiqués sur le clavier, et accessibles à l'aide de la touche **[Alt]** de droite, notée **[Alt Gr]**. 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
[Alt Gr] + [2]	~	tilde
[Alt Gr] + [3]	#	dièse
[Alt Gr] + [4]	{	accolade gauche
[Alt Gr] + [5]	[crochet gauche
[Alt Gr] + [6]		pipe

Touches	Résultat	Nom du symbole
[Alt Gr] + [8]	\	antislash
[Alt Gr] + [0]	@	arobase
[Alt Gr] + [)]]	crochet droit
[Alt Gr] + [=]	}	accolade droite

11.2.6 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 Privilèges d'administration

Certaines commandes qui viennent modifier le système nécessitent des droits d'administration. Elles pourront alors accéder à l'intégralité du système, sans restriction... avec les risques que cela comporte, donc.

Pour exécuter une commande avec les droits d'administration, il faut mettre `pkexec` avant le nom de la commande. Une fenêtre demande alors un mot de passe avant d'exécuter la commande.

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 [enter] (ou [return]) 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 :

```
Bureau
essai
```

Bureau 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 **Bureau** 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 fichier. On va donc :

- taper **rm -v** dans notre terminal,
- taper un espace afin de séparer 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 **[enter]** (ou **return**) et constater que l'ordinateur répond :

```
< /home/LOGIN/essai > supprimé
```

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 :

```
Bureau
```

[page 34] 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.

11.6 Attention aux traces !

[page 21] La plupart des *shells* enregistrent automatiquement les lignes de commande que l'on a tapées dans un fichier « d'historique ». C'est bien pratique pour retrouver plus tard des commandes que l'on a pu utiliser, mais cela laisse également sur le disque une trace de nos activités.

Le *shell* standard dans Debian s'appelle **bash**. Avec ce dernier, pour désactiver temporairement l'enregistrement de l'historique dans le terminal que l'on utilise, il suffit de faire :



```
unset HISTFILE
```

[page 129] Par ailleurs, les commandes sont enregistrées dans le fichier caché **.bash_history** (qui se trouve dans le *Dossier personnel*). On peut donc avoir envie de le nettoyer de temps en temps.

11.7 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 console² » 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.editions-eyrolles.com/Chapitres/9782212124248/Pages-63-82_Novak.pdf

Choisir une phrase de passe

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : 10 minutes environ.

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 emails, 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*¹, ça évite de la noter sur un papier, grave erreur qui rend caduc l'intérêt de se faire une phrase de passe béton. Mais, et c'est tout aussi important, une bonne phrase de passe doit être *aussi difficile à deviner que possible*. Évitons donc les phrases de passe formées de 15 mots composés de caractères aléatoires qu'on aura oublié à peine 15 minutes après l'avoir trouvé. De même que les paroles de tubes disco des années 80.

Une technique simple pour trouver une bonne phrase de passe, difficile à deviner, mais néanmoins facile à retenir, est de fabriquer une phrase qui n'est pas issue d'un texte existant. En effet, que ce soit des paroles de chansons, le vers d'un poème, ou une citation d'un livre, des outils comme le projet Gutenberg² rendent de plus en plus facile le test de phrases de passe tirées de la littérature existante³.

Même s'il faut ici vous en remettre à votre imagination, nous pouvons quand même donner quelques pistes quant aux bonnes habitudes à avoir lors du choix d'une phrase de passe.

1. Fabriquons tout d'abord une phrase dont on se souviendra aisément. Faisons tourner nos ménings un instant.

1. Randall Munroe, 2014, *Password Strength* [<https://xkcd.com/936/>] (en anglais).

2. Wikipédia, 2017, *Projet Gutenberg* [https://fr.wikipedia.org/wiki/Projet_Gutenberg].

3. Dan Goodin, 2013, *How the Bible and YouTube are fueling the next frontier of password cracking* [<http://arstechnica.com/security/2013/10/how-the-bible-and-youtube-are-fueling-the-next-frontier-of-password-cracking/>] (en anglais).

2. Trouvons dans cette phrase ce que l'on peut modifier pour la rendre plus difficilement devinable. On peut ainsi penser à y ajouter de l'argot, des mots de différentes langues, mettre des majuscules là où l'on ne les attend pas, remplacer des caractères par d'autres, laisser libre cours à son imagination quant à l'orthographe, *etc.*

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 absentes ou difficiles à retrouver, et surtout de mauvais codage des caractères, si l'on est amené à taper notre phrase de passe sur un clavier différent de celui dont on a l'habitude.

Un exemple, prenons cette phrase sans sens apparent :

correct cheval pile agraphe

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

korect sheVall-peEla ! graff

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

N'oublions toutefois pas que si trouver une telle phrase de passe n'est pas sans effort, cela ne dispense aucunement d'en trouver une différente par support que l'on chiffre. L'usage d'une même phrase de passe, ou pire d'un même mot de passe, pour une variété de choses différentes, boîtes mail, compte *PayPal*, banque en ligne *etc.* peut rapidement s'avérer désastreux si elle est dévoilée. Par exemple si le service de banque en ligne se fait pirater.

Démarrer sur un CD, un DVD ou une clé USB

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

 Durée : 1 minute à 20 minutes environ.

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 micrologiciel. On a [page 14] vu que c'est lui qui permet de choisir le périphérique (disque dur, clé USB, CD-ROM ou DVD, etc.) où se trouve le système qu'on veut utiliser.

13.1 Essayer naïvement

Commencer par mettre le CD ou le DVD 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 micrologiciels récents, il est souvent possible de choisir un périphérique de démarrage au cas par cas. Mais ce n'est pas toujours possible, notamment pour certains ordinateurs récents équipés de Windows 10, pour lesquels la manipulation est plus compliquée. Il faudra entre autre désactiver le *Secure boot*¹ et sans doute chercher sur le web comment démarrer sur une clé USB sur ce modèle d'ordinateur particulier.

(Re)démarrer l'ordinateur en regardant attentivement les tout 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 F2 ou F12 ou F9 ou Échap.

1. Disabling Secure Boot [https://msdn.microsoft.com/fr-fr/windows/hardware/commercialize/desktop/disabling-secure-boot] (en anglais)

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*².

Mais revenons à nos PC. Souvent, le micrologiciel 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 en question (ne pas maintenir la touche enfoncee, 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 HDD: BDS GH87766319819 |
| 8: Legacy Floppy Drives |
| |
| <Enter Setup> |
| |
+-----+
```

Si ça marche, c'est gagné. Choisir la bonne entrée dans ce menu, en se déplaçant avec les flèches du clavier **[↑]** et **[↓]**, puis appuyer sur **[↵]** (ou **[return]**). 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 micrologiciel

Si choisir un périphérique de démarrage temporaire ne fonctionne pas, il va falloir rentrer dans le micrologiciel pour choisir manuellement l'ordre de démarrage. Pour pimenter un peu la chose, les micrologiciels sont quasiment tous différents, de telle sorte qu'il est impossible de donner une recette qui marche systématiquement³.

13.3.1 Entrer dans l'interface de configuration du micrologiciel

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 micrologiciel. Cette touche est souvent **[Suppr]** (**[Delete]**, **[Del]**) ou **[F2]**, parfois **[F1]**, **[F10]**, **[F12]**, **[Échap]**, **[←]** (ou **[tab]**), voire autre chose encore.

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

3. Des tutoriels illustrés pour quelques BIOS sont disponibles sur cette page [<https://www.hiren.info/pages/bios-boot-cdrom>].

Voici un tableau qui résume les touches d'accès au micrologiciel pour quelques fabricants d'ordinateurs communs⁴.

Fabriquant	Touches observées
Acer	[F1], [F2], [Suppr]
Compaq	[F10]
Dell	[F2]
Fujitsu	[F2]
HP	[F1], [F2], [F10], [F12], [Échap]
IBM	[F1]
Lenovo	[F1]
NEC	[F2]
Packard Bell	[F1], [F2], [Suppr]
Samsung	[F2]
Sony	[F1], [F2], [F3]
Toshiba	[F1], [F2], [F12], [Échap]

Souvent, le micrologiciel 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 enfoncee, 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 micrologiciel soit configuré pour afficher un logo plutôt que ses messages. Essayer d'appuyer sur [Échap] ou sur [←] (ou [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.

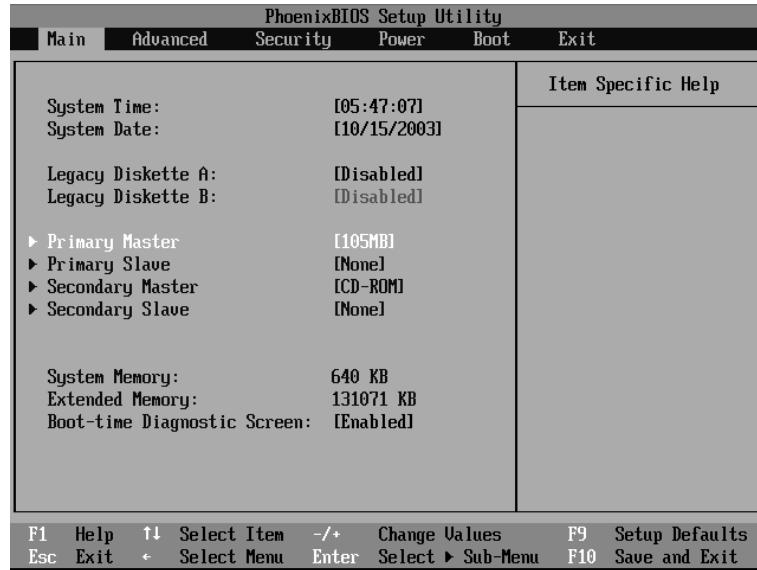
13.3.2 Modifier la séquence de démarrage

Une fois dans le micrologiciel, 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... Elle est souvent en anglais : 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 $\leftarrow\uparrow\downarrow\rightarrow$: *Move* (en anglais, déplacer se dit « *move* »). Il s'agit des flèches du clavier [↓] et [↑] et/ou [←] et [→]. Parfois, la touche [←] (ou [tab]) est utile aussi.

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

4. Tim Fisher, 2014, *BIOS Setup Utility Access Keys for Popular Computer Systems* [https://pcsupport.about.com/od/fixtheproblem/a/biosaccess_pc.htm] ainsi que Michael StevensTech, 2014, *How to access/enter Motherboard BIOS* [http://michaelstevenstech.com/bios_manufacturer.htm] (liens en anglais).



Un écran de BIOS

- 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/DVD 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 DVDROM**, soit sélectionnée. Parfois, ce sont plutôt les touches *Page suivante* ou *PgDown* et *Page précédente* ou *PgUp* 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.

13.3.3 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'absence de CD, de DVD 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 micrologiciel, 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.

13.3.4 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 appuie sur [**d**] (ou [**return**]).

Utiliser un système *live*

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : 30 minutes à une heure, plus environ 30 minutes de téléchargement.

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 18]

14.1 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 Tails (*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, du moins au bout d'un certain temps.

[page 12]

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 l'installer sur une clé USB ou de le graver sur un DVD.

Il est en général conseillé d'utiliser Tails sur une clé USB : cela permet d'utiliser certaines fonctionnalités qui ne sont pas disponibles en DVD comme les mises à jour automatiques et la persistance.

Néanmoins, étant donné qu'il est possible d'écrire des données sur une clé USB alors que ça ne l'est pas sur un DVD, cela rend possible pour des personnes malveillantes de modifier votre système *live* pour, par exemple, enregistrer vos mots de passe ou vos frappes au clavier. Si l'on choisit pour ces raisons d'utiliser un DVD, il ne faudra pas négliger de faire les mises à jour manuellement, sous peine d'utiliser un système contenant des failles connues !

[page 24]

14.2 Télécharger et installer *Tails*

On va expliquer ici comment télécharger la dernière version de Tails à partir de son site web officiel, vérifier son authenticité et l'installer sur une clé USB ou la graver sur un DVD.

Si l'on dispose déjà d'une installation de la dernière version de Tails, il est possible de la dupliquer simplement. Suivre pour cela l'outil cloner Tails.

Pour télécharger et installer Tails, on va utiliser l'assistant officiel, disponible sur la page web <https://tails.boum.org/install/index.fr.html>.

Attention : ce guide fournit des explications sur la vérification de l'authenticité de l'image de Tails. Lorsqu'on arrive à la section « Vérifier l'image ISO » de la documentation officielle de Tails, se rapporter à vérifier son authenticité.

14.2.1 Télécharger Tails

Tails peut être téléchargé de deux manières ; soit directement *via* un navigateur web (en HTTP), soit à l'aide de BitTorrent.

Dans tous les cas, il faudra suivre l'assistant d'installation de Tails [<https://tails.boum.org/install/index.fr.html>] correspondant au système d'exploitation que l'on utilise.

Téléchargement direct

Si l'on choisit de télécharger l'image directement avec son navigateur web, il faudra au préalable installer le module complémentaire de Firefox permettant de vérifier l'intégrité de Tails. Une fois l'image téléchargée, si l'on désire aller plus loin dans la vérification de l'image ISO et également s'assurer de son authenticité, il faudra cliquer sur **signature OpenPGP** et suivre les instructions correspondantes.

Téléchargement *via* BitTorrent

Si l'on choisit de télécharger l'image *via* BitTorrent, le téléchargement contiendra à la fois l'image ISO et la signature permettant de vérifier son authenticité.

14.2.2 Vérifier l'authenticité du système live

Le module complémentaire de téléchargement de Tails effectue une première vérification de l'image téléchargée. Il garantit notamment¹ que l'image correspond à celle distribuée par le site de Tails, mais ne protège pas contre une attaque sur le site de Tails.

L'image du système *live* que l'on vient de télécharger est signée numériquement avec OpenPGP. On va utiliser cette signature pour vérifier l'authenticité de l'image téléchargée de façon plus robuste.

Commencer par importer la clé OpenPGP qui signe les ISO de Tails :

Tails developers
tails@boum.org 'offline long-term identity key'

Puis vérifier la signature numérique de l'image ISO. L'empreinte observée par les auteurs de ce guide est, en admettant que c'est un exemplaire original que l'on a entre les mains :

¹. le modèle de menace auquel répond le module complémentaire de téléchargement de Tails est documenté sur le site de Tails (en anglais) [<https://tails.boum.org/blueprint/bootstrapping/extension/>].

A490 D0F4 D311 A415 3E2B B7CA DBB8 02B2 58AC D84F

14.2.3 Installer Tails sur le support choisi

Retourner à l'assistant d'installation de Tails [<https://tails.boum.org/install/index.fr.html>], pour y trouver les instructions d'installation de Tails sur une clé USB qui correspondent à votre système d'exploitation.

Si en revanche on préfère installer Tails sur un DVD, il faudra se rendre à la page dédiée [<https://tails.boum.org/install/dvd/index.fr.html>].

14.3 Cloner ou mettre à jour une clé *Tails*

Une fois que l'on dispose d'un DVD ou d'une clé USB de Tails, il est possible de le dupliquer, par exemple pour créer une clé USB avec la persistance correspondant à une nouvelle identité contextuelle, pour donner une clé Tails à une connaissance, ou encore pour mettre à jour une clé USB contenant une ancienne version de Tails.

Pour ce faire, on va suivre la documentation officielle de Tails, qui est disponible à partir de n'importe quel DVD ou clé USB de Tails, même sans connexion à Internet.

Démarrer Tails. Sur le bureau, cliquer sur l'icône *Documentation de Tails*. Dans le menu à droite, cliquer sur *Documentation*. Dans l'index qui s'ouvre, chercher la section *Téléchargement et installation* et cliquer sur la page *Installation depuis un autre Tails*. C'est celle-ci qu'il s'agira de suivre.

Pour mettre à jour la clé ainsi créée, il faudra par la suite suivre la page *Mettre à jour une Clé USB ou une carte SD Tails* dans la section *Premiers pas avec Tails*.

[page ci-contre]

[plus bas]

14.4 Démarrer sur un système *live*

Dès que la copie ou la 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 micrologiciel pour qu'il démarre sur le bon support : voir la recette expliquant comment démarrer sur un média externe pour les détails.

[page 95]

Au démarrage, Tails affiche un écran qui permet de choisir, entre autres options, la langue d'affichage et la disposition du clavier.

14.5 Utiliser la persistance de *Tails*

Lorsqu'on utilise Tails à partir d'une clé USB, il est possible de créer un volume persistant chiffré sur l'espace libre de la clé créée par l'Installeur de Tails.

[page suivante]

Les données contenues dans le volume persistant sont sauvegardées et restent disponibles d'une session d'utilisation de Tails à l'autre. Le volume persistant permet de sauvegarder des fichiers personnels, des clés de chiffrement, des configurations ou des logiciels qui ne sont pas installés par défaut dans Tails.

Une fois le volume persistant créé, on peut choisir de l'activer, ou non, à chaque démarrage de Tails.

[page suivante]

On pourra enfin le supprimer lorsqu'on ne voudra plus pouvoir accéder à ses données.

[page 104]

L'utilisation d'un volume persistant n'est toutefois pas sans conséquences quant aux traces laissées. C'est pourquoi il faudra commencer par lire la page d'avertisements concernant l'usage de la persistance.

Pour cela, cliquer sur l'icône *Documentation de Tails* se trouvant sur bureau. Dans le menu à droite, cliquer sur *Documentation*. Dans l'index qui s'ouvre, chercher la section *Premiers pas avec Tails* et cliquer sur la page *Avertissements à propos de la persistance*.

14.5.1 Créer et configurer un volume persistant

L'objectif de cette recette est de créer et de configurer un volume persistant sur une clé USB Tails.

Pour ce faire, on va suivre la documentation officielle de Tails, qui est disponible à partir de n'importe quelle clé USB ou DVD de Tails, même sans connexion à Internet.

Démarrer Tails. Sur le bureau, cliquer sur l'icône *Documentation de Tails*. Dans le menu à droite, cliquer sur *Documentation*. Dans l'index qui s'ouvre, chercher la section *Premier pas avec Tails* et cliquer sur la page *Créer et configurer le volume persistant*. Suivre cette page de documentation.

Si l'on a déjà un volume persistant et que l'on souhaite simplement modifier ses paramètres, aller directement à la section *Options de persistance*.

14.5.2 Activer et utiliser un volume persistant

L'objectif de cette recette est d'activer le volume persistant nouvellement créé sur une clé USB de Tails.

Pour ce faire, on va suivre la documentation officielle de Tails, qui est disponible à partir de n'importe quelle clé USB ou DVD de Tails, même sans connexion à Internet.

Démarrer Tails. Sur le bureau, cliquer sur l'icône *Documentation de Tails*. Dans le menu à droite, cliquer sur *Documentation*. Dans l'index qui s'ouvre, chercher la section *Premier pas avec Tails* et cliquer sur la page *Persistance* et enfin *Activer et utiliser le volume persistant* et suivre cette page de documentation.

14.5.3 Supprimer un volume persistant

L'objectif de cette recette est de supprimer un volume persistant créé précédemment sur une clé USB de Tails.

Pour ce faire, on va suivre la documentation officielle de Tails, qui est disponible à partir de n'importe quelle clé USB ou DVD de Tails, même sans connexion à Internet.

Démarrer Tails. Sur le bureau, cliquer sur l'icône *Documentation de Tails*. Dans le menu à droite, cliquer sur *Documentation*. Dans l'index qui s'ouvre, chercher la section *Premier pas avec Tails*, cliquer sur *Supprimer le volume persistant* puis suivre cette page de documentation.

14.5.4 Installer un logiciel additionnel persistant dans Tails

Tails contient des logiciels adaptés à la plupart des tâches courantes lors de l'utilisation d'Internet et de la création de documents. Toutefois, pour des projets spécifiques, on peut avoir besoin d'installer un logiciel spécifique dans *Tails*, comme par exemple un logiciel de conception et de simulation de circuits électroniques.

Lorsque *Tails* est installé sur une clé USB, il est possible de configurer un volume persistant pour qu'un ou plusieurs logiciels spécifiques soient installés de manière automatique à chaque démarrage.

Trouver le nom de paquet à installer On a besoin du nom exact du paquet à installer. Pour le trouver, suivre la recette [trouver un logiciel](#). Par exemple, notre logiciel de conception de circuits électroniques est fourni par le paquet `geda`.

Configurer les logiciels additionnels Si l'on ne dispose pas encore d'une clé USB contenant un volume persistant, il faut en créer une grâce aux recettes [cloner une clé Tails](#) puis [créer et configurer un volume persistant dans Tails](#).

On suivra ensuite l'outil [créer et configurer un volume persistant dans Tails](#) pour

activer les options *Paquets APT* et *Listes APT*.

Redémarrer alors *Tails*. Une fois redémarré, au niveau de l'écran d'accueil, après avoir choisi le français comme langue, taper la phrase de passe du volume persistant sous la section *Stockage persistant chiffré* puis cliquer sur *Déverrouiller*. Ensuite, cliquer sur le **+** situé sous la rubrique *Paramètres supplémentaires*, puis cliquer sur *Mot de passe d'administration*, le rentrer deux fois et cliquer sur *Ajouter*. Enfin, cliquer sur *Démarrer Tails*.

Une fois sur le bureau, ouvrir un *Terminal administrateur* à partir de la vue d'ensemble des Activités, s'obtenant en appuyant sur la touche  ( sur un Mac). Entrer le mot de passe choisi précédemment puis *Valider*. Le terminal s'ouvre, taper dedans :

#_

```
gedit /live/persistence/TailsData_unlocked/  
→ live-additional-software.conf
```

Puis appuyer sur *Entrer*.

Un fichier texte s'ouvre. Chaque ligne de ce fichier doit contenir le nom exact d'un paquet à installer. On va donc ajouter une ligne avec le nom du paquet trouvé précédemment. Dans notre exemple, on ajoutera la ligne :

```
geda
```

Enregistrer le fichier en cliquant sur *Enregistrer*. On peut maintenant fermer le fichier texte ainsi que le terminal et redémarrer *Tails*. Une fois la persistance activée et l'ordinateur connecté au réseau, notre logiciel devrait s'installer automatiquement au bout d'un temps plus ou moins long en fonction de la taille et du nombre de logiciels à installer (une fenêtre signale alors le succès de l'installation).

[page ci-contre]

Installer un système chiffré

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : compter une journée, avec plusieurs périodes d'attente (parfois longues).

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.

Il est possible d'installer un système d'exploitation GNU/Linux comme Debian ou Ubuntu, 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.

15.1 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.

15.1.1 Limites d'un système chiffré

Nous recommandons chaudement les lectures préalables suivantes :

- le chapitre concernant le chiffrement (et ses limites),
[page 39]
- 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 59]

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

15.1.2 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 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...

15.1.3 Limites dans la prise en charge du matériel

Utiliser un système d'exploitation libre comme Debian a un désavantage : les fabricants de matériel y font généralement peu attention. Il arrive donc qu'il ne soit pas facile, voire complètement impossible, d'utiliser un ordinateur ou l'un de ses périphériques avec Debian.

La situation s'améliore depuis quelques années : le fonctionnement du matériel tend à s'homogénéiser, et surtout, la diffusion des systèmes libres pousse de plus en plus les fabricants à aider, directement ou non, à ce que leur matériel fonctionne¹.

Cependant, avant de remplacer un système d'exploitation, cela peut être une bonne idée de s'assurer que le matériel nécessaire fonctionne bien à l'aide d'un système *live*. Le système Tails, par exemple, est basé sur Debian. Le matériel qui fonctionne avec l'un devrait donc fonctionner avec l'autre sans difficultés.

15.2 Télécharger un support d'installation

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

15.2.1 Avec ou sans microcodes non-libres ?

Certains périphériques de l'ordinateur peuvent nécessiter, pour fonctionner, que le système leur fournisse un « microcode » (ou *firmware*) qui n'est pas libre... mais ce n'est pas toujours le cas.

Un micro—quoi ?

Ces microcodes sont des programmes qui ont la particularité de s'exécuter sur des puces électroniques à l'intérieur du périphérique et non sur le processeur de l'ordinateur. C'est par exemple le cas du programme qui contrôlera le déplacement des parties mécaniques d'un disque dur ou le fonctionnement du système de radio d'une carte Wi-Fi. On ne se rend pas forcément compte qu'ils existent... car la plupart des microcodes sont livrés directement avec le matériel.

Mais pour d'autres périphériques, le système d'exploitation doit envoyer le microcode à un composant lors de son initialisation.

Ceux qui sont libres sont livrés avec le programme d'installation de Debian. Malheureusement, la plupart des microcodes ne sont pas libres. Nous devons donc mettre nous-mêmes à disposition du programme d'installation tout microcode non-libre nécessaire au fonctionnement de l'ordinateur : c'est typiquement le cas pour certaines cartes Wi-Fi.

Encore une histoire de compromis

Si l'on installe notre système chiffré sur un ordinateur portable, il est très probable que des microcodes soient nécessaires pour faire marcher le Wi-Fi, voire pour avoir un affichage de bonne qualité.

Sur un ordinateur fixe sans Wi-Fi, il est assez plausible que notre système chiffré fonctionne correctement sans microcodes.

1. Pour certains matériels, des problèmes peuvent venir de défauts dans le fonctionnement des microcodes intégrés. Ces problèmes sont parfois corrigés par des mises à jour que fournissent les fabricants. Cela peut donc être une bonne idée de faire les mises à jour du micrologiciel (BIOS ou UEFI), de l'*Embedded Controller* ou d'autres composants avant de procéder à l'installation. Malheureusement, ces procédures diffèrent trop d'un matériel à un autre pour être détaillées dans cet ouvrage, mais peuvent en général être trouvées sur le site du constructeur...

Même s'il n'y pas à notre connaissance d'éléments qui prouvent son utilisation, on peut envisager que le microcode propriétaire d'une carte Wi-Fi nous espionne à notre insu... sauf que sans microcode, elle ne fonctionnera tout simplement pas. C'est encore une fois une histoire de compromis.

15.2.2 L'image d'installation par le réseau

Le plus rapide est d'utiliser une image d'installation par le réseau. Elle contient uniquement les tout premiers morceaux du système. L'installateur télécharge ensuite, depuis Internet, les logiciels à installer. 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 à l'intérieur du logiciel d'installation).

Il existe plusieurs fichiers (également appelés « images ») contenant une copie de l'image d'installation, selon l'architecture du processeur. Dans la plupart des cas, il faudra télécharger celui dont le nom se termine par `amd64-i386-netinst.iso`, dit multi-architecture, qui conviendra pour les architectures 32 et 64 bits et qui fonctionnera sur la plupart des ordinateurs domestiques fabriqués après 2006².

[page 10]

Il faut choisir entre la version contenant les microprogrammes non-libres³ et celle entièrement libre sans les microprogrammes⁴.

15.2.3 L'image avec l'environnement graphique

S'il n'est pas possible de connecter à Internet l'ordinateur sur lequel on souhaite installer Debian, il est possible de télécharger un DVD contenant tout le système de base ainsi que l'environnement graphique habituel. Cela nécessite d'avoir accès à un graveur de DVD ou à une clé USB d'au moins 4GB.

De la même manière que pour l'installation par le réseau, on choisira le DVD correspondant à notre architecture⁵.

Seul le premier DVD est nécessaire pour réaliser l'installation. Le nom du fichier à télécharger se termine soit par `-i386-DVD-1.iso` (32 bits), soit par `-amd64-DVD-1.iso` (64 bits).

15.3 Vérifier l'empreinte du support d'installation

Il est bon de s'assurer que le téléchargement de l'image s'est bien déroulé en vérifiant l'empreinte de l'installateur, pour s'assurer de son intégrité et de son authenticité. Nous allons procéder en deux étapes, une première nous assurant de son intégrité, et une seconde assurant son authenticité.

[page 39]

Pour cela, il est nécessaire de démarrer sur un système déjà installé. Si l'on a accès à un ordinateur sous GNU/Linux, par exemple celui d'une amie, tout va bien. Si on ne dispose que d'un système *live*, il est par exemple possible de mettre l'image téléchargée sur une clé USB, puis de vérifier l'empreinte à partir du système *live*.

[page 101]

15.3.1 Vérifier l'intégrité du support d'installation

Pour cela suivre l'outil concernant les sommes de contrôle. Il sera nécessaire de calculer

[page 149]

2. Des ordinateurs portables utilisant l'architecture de processeur [page 10] ARM apparaissent, mais les auteurs de ce guide n'en ont encore jamais rencontré pour tester.

3. Les images multi-architectures d'installation par le réseau contenant les microcodes non-libres sur le site de Debian [<https://cdimage.debian.org/cdimage/unofficial/non-free/cd-including-firmware/current/multi-arch/iso-cd/>].

4. Les images multi-architectures officielles d'installation par le réseau sur le site de Debian [<https://cdimage.debian.org/cdimage/release/current/multi-arch/iso-cd/>].

5. Les DVD d'installation de Debian sont disponibles ici pour les différentes architectures [<https://www.debian.org/CD/http-ftp/index.fr.html#stable>].

la somme de contrôle SHA512 du support d'installation téléchargé et vérifier si celle-ci correspond à celle contenue dans le fichier **SHA512SUMS** contenu dans le même dossier que le support d'installation. Fichier que l'on devra télécharger également.

15.3.2 Vérifier l'authenticité du support d'installation

Si la vérification de l'intégrité s'est bien déroulée, à savoir si les deux sommes de contrôles calculées correspondent, on peut poursuivre le processus afin de vérifier son authenticité. En effet, un adversaire pourrait fournir un support d'installation corrompu et sa somme de contrôle. La vérification précédente nous permettrait simplement de constater que le fichier téléchargé est bien celui qui était disponible sur le site web, pas qu'il est celui qu'on espère avoir. Le deuxième tome explique comment s'assurer de l'authenticité de l'installeur téléchargé, car l'empreinte est signée avec GnuPG, qui utilise la cryptographie asymétrique, il faudra donc directement se référer à l'outil concernant la vérification d'une signature. Après avoir téléchargé le fichier **SHA1SUMS.sign**, suivre les étapes permettant de vérifier la signature cryptographique du fichier **SHA512SUMS**.

15.4 Préparer les supports d'installation

Une fois l'image du support d'installation choisie, téléchargée et vérifiée, il nous reste à l'installer sur une clé USB, un CD ou un DVD.

15.4.1 Créer une clé USB d'installation

Si l'on dispose d'une clé USB vide, ou contenant uniquement des données auxquelles on ne tient pas et qu'on a accès à un système basé sur Linux tel que Debian ou Tails, c'est l'option la plus rapide.



Attention : les données éventuellement présentes sur la clé seront perdues. Par contre, il serait facile de procéder à une analyse pour retrouver les fichiers dont le contenu n'aurait pas été écrasé auparavant...

Ouvrir *Disques* à partir de la vue d'ensemble des Activités : appuyer sur la touche () sur un Mac), puis taper **disque** et cliquer sur *Disques*.

Une fois *Disques* ouvert, on peut brancher notre clé USB. Une entrée correspondant à cette dernière devrait apparaître dans la liste située à gauche. Cliquer dessus pour la sélectionner.

Cliquer ensuite sur le menu et sélectionner *Restaurer l'image disque...*. Dans *Image à restaurer*, sélectionner l'image ISO précédemment téléchargée. Cliquer sur *Démarrer la restauration...*.

Une fenêtre nous demande *Voulez-vous vraiment écrire l'image disque sur le périphérique ?*. Vérifier que la taille et le modèle du périphérique concerné correspondent bien à la taille et au modèle de notre clé USB. Si c'est le cas, cliquer sur *Restaurer*.

Le mot de passe d'administration nous est alors demandé, le taper et *S'authentifier* pour lancer l'écriture de la clé d'installation. Une fois la restauration terminée, cliquer sur pour éjecter la clé.

15.4.2 Graver l'image d'installation sur un CD ou un DVD

Si l'on a pas de clé USB ou pas d'accès à un système Linux, on peut graver l'image d'installation sur un CD ou un DVD.

Le fichier téléchargé est une « image ISO », c'est-à-dire un format de fichiers que la plupart des logiciels de gravure reconnaissent comme « image CD brute ». En général, si on insère un disque vierge dans son lecteur, le logiciel de gravure s'occupe tout seul de transformer cette image en l'écrivant sur le disque vierge — en tout cas, ça marche avec Tails, 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 InfraRecorder⁶ fera parfaitement l'affaire.

15.5 L'installation proprement dite

Pour installer la Debian chiffrée depuis le support d'installation (CD, DVD ou clé USB), il faut démarrer sur celui-ci en suivant la recette correspondante.

[page 95]

À partir de là, l'installation proprement dite peut commencer : prévoir du temps devant soi et quelques mots croisés, car l'ordinateur pourra travailler longtemps sans surveillance particulière.

Vérifier, dans le cas d'un CD d'installation par le réseau, que le câble reliant l'ordinateur au réseau est bien branché. Et si il s'agit d'un ordinateur portable, vérifier que le câble d'alimentation est branché, car il n'y a pas de notification de batterie faible durant l'installation.

Le programme d'installation de Debian dispose de sa propre documentation⁷. En cas de doute à la lecture des étapes décrites par la suite, cela peut valoir le coup d'y jeter un œil. Par ailleurs, pour la plupart des choix qu'il nous demande de faire, le programme d'installation nous proposera automatiquement une réponse qui fonctionne généralement...

15.5.1 Lancement de l'installeur

On démarre donc sur le support d'installation (CD, DVD ou clé USB). Un premier menu nommé *Debian GNU/Linux installer boot menu* apparaît.

Dans le cas où on a choisi un CD multi-architecture, l'option sélectionnée automatiquement par l'installeur sera en principe *Graphical install* et une option *32-bit install options* sera disponible ; dans ce cas, le programme d'installation a détecté que le processeur est compatible avec l'architecture `amd64`, qui apporte quelques avantages en termes de sécurité. Appuyer sur la touche ↴ (ou ↵) pour lancer la suite du programme d'installation.

[page 10]

15.5.2 Choisir la langue et la disposition du clavier

- Après un peu de patience, un menu nommé *Select a language* apparaît : l'installeur propose de choisir une langue pour la suite de l'installation. Sélectionner *Français*. Pour passer à l'étape suivante, il faudra à chaque fois choisir *Continue*.
- Un menu demande le pays, pour peaufiner l'adaptation du système. Choisir son lieu géographique.
- Dans *Configurer le clavier*, le choix par défaut *Français* convient si l'on a un clavier français « azerty ».
- L'installeur charge ensuite les fichiers dont il a besoin.

6. <http://infrarecorder.org/>

7. Le manuel d'installation est disponible dans de nombreuses versions [<https://www.debian.org/releases/stable/installmanual.fr.html>]. On suivra celui correspondant à l'architecture du processeur [page 10].

15.5.3 Configuration du réseau et baptême de la machine

- L'installateur prend alors un peu de temps pour configurer le réseau. Si notre ordinateur possède plusieurs cartes réseau, il faut choisir celle dont on va se servir pour l'installation. Le choix par défaut est généralement le bon, il s'agit d'une carte réseau *Ethernet*.
- On nous demande ensuite 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 ou modifiés avec le système qu'on est en train d'installer.
- L'installateur demande alors un *Domaine*. Sans entrer dans les détails, mieux vaut laisser ce champ vide (donc effacer ce que le programme peut éventuellement avoir pré-rempli).

15.5.4 Créer les utilisateurs et choisir les mots de passe

Le programme d'installation nous demande maintenant de choisir le *mot de passe du superutilisateur*. C'est un mot de passe qui serait nécessaire pour réaliser les tâches d'administration de l'ordinateur : mises à jour, installation de logiciels, modifications majeures du système, *etc.*

Il est toutefois plus simple de s'épargner un mot de passe supplémentaire, et de permettre que le premier compte utilisateur créé sur le système ait le droit de faire des opérations d'administration⁸, en redemandant le mot de passe. Pour cela, il suffit de ne pas entrer de mot de passe pour le superutilisateur : laisser simplement la case vide et choisir *Continuer*.

- 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 ou modifiés dans cette session ; il peut donc être intéressant 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'installateur 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'installateur demande un mot de passe pour l'utilisateur. C'est lui qui aura le droit d'administrer l'ordinateur, si l'on a décidé de ne pas entrer un mot de passe « superutilisateur » précédemment.

15.5.5 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 le système actuellement installé, il correspond en général au premier disque de la liste. La taille du disque est un indice permettant de ne pas se tromper, pour ne pas essayer d'installer Debian sur la clé USB contenant l'installateur par exemple.
- L'installateur propose ensuite différents *Schéma de partitionnement*. Choisir *Tout dans une seule partition*.
- L'installateur 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 ?*

⁸. Ce mode est appelé *sudo*, car dans le terminal, il sera possible, en ajoutant **sudo** au début de la ligne, d'exécuter une commande en tant que « superutilisateur ».

- L’installeur va alors remplacer l’ancien contenu du disque par des données aléatoires. C’est très long — plusieurs heures sur un gros disque — et ça laisse donc du temps pour faire autre chose !
- L’installeur demande alors une *Phrase secrète de chiffrement*. Choisir une bonne phrase de passe et la taper, puis confirmer la phrase de passe en la tapant une seconde fois. [page 93]
- L’installeur montre une liste de toutes les partitions qu’il va créer. Il est possible de lui faire confiance en laissant *Terminer le partitionnement et appliquer les changements* sélectionné.
- L’installeur prévient qu’il va écrire des modifications 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.

15.5.6 Installation du système de base

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

15.5.7 Configurer l’outil de gestion des paquets

Si l’installeur propose d’utiliser un support d’installation autre que celui utilisé pour démarrer l’installeur, le choix par défaut, *Non*, convient.

La question suivante concerne le serveur à utiliser pour télécharger les logiciels. Si elle n’apparaît pas à ce moment, pas d’inquiétude, c’est simplement que l’installeur utilisé n’a pas besoin du réseau pour poursuivre. Dans ce cas, elle arrivera un peu plus tard au cours de l’installation.

- L’installeur demande de choisir le *Pays du 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.

15.5.8 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 de divulguer beaucoup d’informations supplémentaires : 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 quels sont les *Logiciels à installer*. Ses propositions conviennent en général : *environnement de bureau Debian, serveur d’impression et Utilitaires usuels du système*.

L’installeur télécharge alors tout le reste du système Debian GNU/Linux et l’installe. C’est long, il y a le temps d’aller faire autre chose.

15.5.9 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 ». À la question *Installer le programme de démarrage GRUB sur le secteur d’amorçage ?* répondre *Oui*.

L'installateur demande le *Périphérique où sera installé le programme de démarrage*. Choisir le disque dur interne, qui sera en général */dev/sda*. Si le doute persiste, un bon indice est de choisir le premier disque dans la liste dont le nom contient *ata* ou *sata*.

Lorsqu'il a terminé, l'installateur propose de redémarrer l'ordinateur en vérifiant que le support d'installation (CD, DVD, clé USB) n'est plus inséré lors du redémarrage. Choisir *Continuer*.

15.5.10 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 : « *Please unlock disk* ». La taper, sans s'inquiéter que rien ne s'affiche, et appuyer sur la touche (ou) à la fin⁹.

Après le démarrage d'un certain nombre de programmes, un écran apparaît avec le nom de la machine et le nom du compte utilisateur entré précédemment. Il faut sélectionner ce dernier, puis entrer le mot de passe associé.

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. La vue d'ensemble des Activités, que l'on peut ouvrir en cliquant sur *Activités* en haut à gauche de l'écran ou en appuyant sur la touche () sur un Mac) permet d'accéder aux nombreux logiciels déjà installés. Pour trouver un logiciel, on peut taper un mot décrivant sa fonction (par exemple *image* pour trouver les logiciels qui travaillent avec des images). Pour afficher tous les logiciels installés, cliquer sur en bas à gauche. Des pages d'aide contenant de nombreux conseils et astuces sont accessibles en tapant *Aide* dans la vue d'ensemble des Activités.

15.6 Quelques pistes pour continuer

- [page 139] Il peut maintenant être utile d'apprendre à sauvegarder des données... et à en effacer « pour de vrai ».
- [page 127] Il est également important d'apprendre à garder son système à jour. Des problèmes affectant les logiciels sont découverts régulièrement, et il est important d'installer les corrections au fur et à mesure de leur disponibilité.
- [page 161]

15.7 Un peu de documentation sur *Debian* et *GNU/Linux*

Voici quelques références de documentations sur Debian et GNU/Linux :

- Le guide de référence officiel de Debian¹⁰ ;
- La page d'accueil de la documentation officielle d'utilisation de Debian¹¹ ;
- Le cahier de l'administrateur Debian¹².

On peut trouver beaucoup de documentation sur l'utilisation de GNU/Linux. Si elles sont souvent très utiles, elles sont, comme beaucoup de choses sur Internet malheureusement, de qualité inégale. En particulier, beaucoup d'entre elles arrêteront de

9. Si l'on n'est pas très à l'aise avec la frappe au clavier, il arrive souvent dans les premiers temps qu'on fasse une erreur de frappe dans la phrase, et c'est d'autant plus probable qu'aucun caractère ne s'affiche. Ne pas s'inquiéter des erreurs répétées, et insister jusqu'à réussir à taper la phrase sans faute... au bout de quelque temps, elle sera « rentrée dans les doigts », et les fautes de frappe se feront plus rares. Cela dit, il ne coûte rien de vérifier qu'on n'a pas malencontreusement laissé la touche VerrMaj enfonce, auquel cas l'on pourrait s'acharner longtemps sur le clavier sans pour autant arriver à déverrouiller le disque dur.

10. <https://www.debian.org/doc/manuals/debian-reference/index.fr.html>

11. <https://www.debian.org/doc/user-manuals.fr.html>

12. <https://debian-handbook.info/browse/fr-FR/stable/>

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 références de wikis et des forums :

- Le wiki officiel de Debian¹³ (partiellement traduit de l'anglais) ;
- Le forum en français sur Debian [debian-fr.org](https://www.debian-fr.org)¹⁴ ;

13. <https://wiki.debian.org/fr/FrontPage>

14. <https://www.debian-fr.org/>

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 doit parfois 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 119]
- 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 122]
- 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 124]

16.1 Trouver un logiciel

 Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

 Durée : de 5 minutes (si l'on connaît le nom du logiciel que l'on cherche) à une demi-heure (si l'on part de zéro).

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.

Pour faire des choix éclairés, lorsque plusieurs logiciels permettent d'effectuer une même tâche, voir [choisir un logiciel](#).

- Ouvrir le *Gestionnaire de paquets* : afficher la vue d'ensemble des Activités en appuyant sur la touche  () sur un Mac), puis taper **paquet** et cliquer sur *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 des mots-clés ou un nom d'application dans la case *Rechercher* (par exemple « dictionnaire allemand openoffice »). Les descriptions des applications peu courantes sont rarement traduites en français. Avec quelques bases d'anglais, il est souvent intéressant d'essayer des mots-clés dans cette langue. Cliquer sur *Rechercher* pour lancer la recherche ;
 - 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](#).

16.2 Critères de choix

C *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

⌚ *Durée : une demi-heure à une heure.*

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.

L'intérêt d'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 [page 31] déporter les logiciels libres disponibles.

16.2.1 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.

16.2.2 Maturité

L'attrait de la nouveauté 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.

16.2.3 Processus de production et « communauté »

L'étiquette *logiciel libre* est un critère essentiellement juridique, qui ne doit jamais [page 32] 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 :

- 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 ?
- Ces discussions rassemblent-elles 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 ?

16.2.4 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 popularité¹, 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.

16.2.5 Passé de sécurité

Voici de nouveau un critère à double tranchant.

1. Debian.org, 2014, *Debian Popularity Contest* [<http://popcon.debian.org/>] (en anglais).

On peut commencer par jeter un oeil sur le suivi de sécurité² 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. Tout comme une nombre relativement élevé de failles, même récentes peut indiquer une communauté de développement active et être meilleur signe qu'aucune faille de sécurité pour un logiciel dont très peu de personnes ne s'occupent au final.

16.2.6 Équipe de développement

Qui a écrit, qui écrit ce logiciel ? Si l'on a réussi à répondre 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* [<https://security-tracker.debian.org/tracker/>].

16.3 Installer un paquet *Debian*

- ⌚ *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*
- ⌚ *Durée : 5 minutes, plus le temps de téléchargement et d'installation (quelques secondes à plusieurs heures selon la taille des logiciels à installer et la vitesse de la connexion).*

16.3.1 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 de la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis en tapant *paquet* puis en cliquant sur *Gestionnaire de paquets Synaptic*

Puisque le gestionnaire de paquets permet de modifier les logiciels installés sur l'ordinateur, un mot de passe est nécessaire pour l'ouvrir.

16.3.2 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 les serveurs de Debian.

16.3.3 Rechercher le paquet à installer

Ensuite, on va trouver le paquet qu'on veut installer. On clique sur l'icône *Rechercher* [page 118] dans la barre d'outils. Là, si on connaît le nom de ce paquet (par exemple grâce à la section précédente), on l'écrit dans la case *Rechercher*, et on sélectionne *Nom* dans la liste déroulante nommée *Rechercher dans*.

16.3.4 Sélectionner le paquet à installer

Vient alors la phase d'installation proprement dite du paquet trouvé précédemment. Il y a différentes façons de 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 clic-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 sélectionner le paquet désiré d'un clic gauche, sans cliquer sur la case à cocher, puis choisir, dans le menu *Paquet*, *Forcer la version... . . .* Sélectionner alors la version désirée. La suite ne change pas.

16.3.5 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 *Résumé* 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, le gestionnaire de paquets 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 Utiliser des logiciels rétroportés

 Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

 Durée : un quart d'heure à une demi-heure.

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 des logiciels plus récents que ceux contenus dans la dernière version stable de Debian, dits *rétroportés*, ou *backports* en anglais.



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* dont on parle ici 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.

16.4.1 Ouvrir le gestionnaire de paquets

Ouvrir le *Gestionnaire de paquets Synaptic* : afficher la vue d'ensemble des Activités en appuyant sur la touche  () sur un Mac), puis taper *paquet* et cliquer sur *Gestionnaire de paquets Synaptic*.

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. Aller dans le menu *Configuration*, puis *Dépôts*.

16.4.2 Configurer l'emplacement du dépôt

Cliquer sur l'onglet *Other Software*, puis sur le bouton *Add*.

Dans *Ligne APT* entrer le répertoire APT à ajouter :

```
deb http://ftp.fr.debian.org/debian/ stretch-backports main
```

Dans ce cas, la *version du dépôt* est *stretch-backports* et la *catégorie* est *main*. Si l'on souhaite également installer des logiciels non-libres, on peut ajouter *contrib* et *non-free* en plus de *main* :

```
deb http://ftp.fr.debian.org/debian/ stretch-backports main
  ↳ contrib non-free
```

Une fois que c'est fait, cliquer sur  *Ajouter une source de mise à jour*.

16.4.3 Mettre à jour les paquets disponibles

Il reste ensuite à fermer la fenêtre *Software & Updates* en cliquant sur le bouton *Fermer*. Il est possible qu'une fenêtre *Dépôts modifiés* apparaisse alors, auquel cas il faudra cliquer sur *Recharger*. Si ce n'est pas le cas, à partir de la fenêtre principale du *Gestionnaire de paquets*, il faut cliquer sur *Recharger* pour mettre à jour les listes de paquets.

À l'été 2017, un bug fait qu'une fenêtre intitulée *Une erreur s'est produite* risque d'apparaître, affichant un message d'erreur de ce type :

```
W: http://ftp.fr.debian.org/debian/dists/stretch-backports/
InRelease: The key(s) in the keyring /etc/apt/trusted.gpg
are ignored as the file is not readable by user '_apt'
executing apt-key
```

Ce n'est pas une erreur bloquante, mais un avertissement, il suffit de cliquer sur le bouton *Fermer* et de l'ignorer.

Pour installer un paquet à partir des rétroportages, suivre l'outil [installer un paquet](#) [page 122] et choisir d'installer une version particulière lorsque la question se présente.

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 34]

[page 34]

17.1 Un peu de théorie

17.1.1 La méthode de Gutmann

La documentation¹ du paquet *secure-delete*, que nous utiliserons dans la prochaine recette, inspirée d'une publication de Peter Gutmann publiée en 1996², 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. [...]*

17.1.2 Le compromis adopté

L'étude de Peter Gutmann 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* qui nous dit, en substance, que pour un disque dur « récent »³, les 38 écritures successives ne sont plus nécessaires : il suffit d'écraser les données quelques fois avec des données aléatoires. Mais mis à part la nature et le nombre de réécritures, le processus décrit précédemment reste tout à fait d'actualité pour les disques durs actuels. Cependant, cette méthode n'est pas adaptée aux disques SSD. Or, les disques SSD tendent de nos jours à remplacer les disques durs...

[page 34]

De surcroît, le NIST (*National Institute of Standards and Technology*, organisme gouvernemental états-unien définissant les protocoles de sécurité utilisés, entre autres, par les administrations de ce pays) a publié une étude récente⁴ de la NSA, qui semble

1. Fichier `README.gz` installé sur une Debian dans `/usr/share/doc/secure-delete`.

2. Peter Gutmann, 1996, *Secure Deletion of Data from Magnetic and Solid-State Memory* [https://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html] (en anglais).

3. Utilisant la technologie PRML [<https://fr.wikipedia.org/wiki/PRML>], apparue en 1990 [<http://www.storagereview.com/guide/histFirsts.html>].

4. NIST, 2006, *Guidelines for Media Sanitization* [http://csrc.nist.gov/publications/nistpubs/800-88/NISTSP800-88_with-errata.pdf] (en anglais).

conclure que sur les disques durs modernes, les données sont tellement collées les unes aux autres qu'il devient pratiquement 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 dans les recettes qui suivent de quelques passages aléatoires, en évoquant tout de même la mise en œuvre de la méthode originale de Gutmann.

[page 53] Il s'agira une fois de plus 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.

17.1.3 Pour les clés USB, disques SSD et autres mémoires *flash*

Pour les clés USB ou autre mémoire *flash* - comme les cartes SD, ou disques SSD - une étude datant de 2011⁵ a montré que la situation était réellement problématique.

Cette étude démontre qu'il est impossible, quel que soit le nombre de réécritures, d'avoir la garantie que tout le contenu d'un fichier donné a bien été recouvert. Même si cela rend inaccessibles les données en branchant simplement la clé, elles sont toujours visibles pour quiconque regarderait directement dans les puces de mémoire *flash*.

La seule méthode qui a fonctionné de façon systématique était de réécrire plusieurs fois l'intégralité de la clé USB. Dans la plupart des cas, deux passages ont suffi, mais sur certains modèles, vingt réécritures ont été nécessaires avant que les données ne disparaissent pour de bon.

[page 133] Partant de ces constats, la réponse préventive semble être de chiffrer systématiquement les clés USB, opération rendant vraiment plus difficile l'extraction des informations directement depuis les puces de mémoire *flash*. Et pour nettoyer a posteriori, l'érasrement entier, malgré ses limites, protège tout de même contre les attaques purement logicielles.

17.1.4 D'autres limites de l'effacement « sécurisé »



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*, *ext4*, 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 première partie.

[page 35]

17.2 Sur d'autres systèmes

[page 31]

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 est donc bien plus difficile de leur faire confiance.

17.3 Allons-y

On peut effacer le contenu :

- de fichiers individuels, voir page suivante ;
- de tout un périphérique, voir page ci-contre ;
- de fichiers déjà supprimés, voir page 132.

5. Michael Wei et Al, 2011, *Reliably Erasing Data From Flash-Based Solid State Drives* [https://www.usenix.org/legacy/events/fast11/tech/full_papers/Wei.pdf] (en anglais).

17.4 Supprimer des fichiers... et leur contenu

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : 5 minutes de préparation, puis quelques secondes à plusieurs heures d'attente en fonction de la taille du fichier à effacer et de la méthode utilisée.

Voici donc la méthode à suivre pour se débarrasser de fichiers, en prenant soin de rendre illisible ce qu'ils contenaient.

⚠ **Attention !** Cette méthode ne fonctionne qu'avec les disques durs mécaniques. Après avoir recouvert le contenu de fichiers sur une clé USB (ou tout autre support de stockage utilisant de la mémoire *flash* comme une carte SD ou un disque SSD) il y a de fortes chances qu'il se trouve encore inscrit dans une région inaccessible du périphérique !

17.4.1 Installer les logiciels nécessaires

Si ce n'est pas déjà fait, il nous faut installer nautilus-wipe, puis redémarrer l'ordinateur. [page 122]

Ce paquet est présent par défaut dans *Tails*.

17.4.2 Supprimer des fichiers et leur contenu à partir du navigateur de fichiers

Dans *Tails*

Afin de supprimer des fichiers et leur contenu en utilisant *Tails*, consulter la documentation en cliquant sur l'icône *Documentation de Tails* se trouvant sur le bureau. Dans le menu à droite, cliquer sur *Documentation*. Dans l'index qui s'ouvre, chercher la section *Chiffrement et vie privée* et cliquer sur la page *Effacer des fichiers de façon sécurisée et nettoyer l'espace disque avec Nautilus Wipe*.

Avec une Debian chiffrée

Afin de supprimer des fichiers et leur contenu depuis le navigateur de fichiers, naviguer jusqu'au fichier, faire un clic-droit sur celui-ci et sélectionner *Écaser*. Une fenêtre s'ouvre nous demandant de confirmer la suppression, et proposant également quelques *Options*.

Nous pouvons choisir le nombre de passes à effectuer afin de recouvrir les données de notre périphérique ainsi que quelques options de comportement lors de l'effacement des données. Les options par défaut sont suffisantes pour les disques durs actuels.

Cliquer ensuite sur *Écaser*. Une fois l'effacement terminé, une fenêtre *L'érasement a réussi* s'ouvre, précisant que *Le(s) élément(s) ont été écrasé(s) avec succès*.

17.5 Effacer « pour de vrai » tout un disque

Avant de se débarrasser d'un disque dur, de le recycler, de réinstaller un système propre, ou encore d'envoyer un ordinateur en panne au Service Après Vente, 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 [page 139]

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.

17.6 Effacer tout le contenu d'un disque

C *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

⌚ *Durée : 5 minutes de préparation, puis plusieurs heures d'attente en fonction de la taille du 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... s'il 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.

La commande est très simple. Elle exige seulement de connaître l'emplacement du périphérique (son chemin) que l'on veut effacer, puis de faire preuve de patience car le processus prend plusieurs heures.

17.6.1 Trouver le chemin du périphérique

Avant tout, il faut savoir repérer sans se tromper le chemin utilisé par le système d'exploitation pour désigner le support de stockage qu'on veut effacer.

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 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.

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 l'Utilitaire de disque

Ouvrir *Disques* : afficher la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper `disque` et cliquer sur *Disques*.

Chercher le chemin du périphérique

La partie située à gauche indique la liste des disques connus du système. On peut cliquer sur l'un d'entre eux afin de voir plus d'informations apparaître sur la partie droite. Les icônes, la taille indiquée ainsi que le nom des disques devraient permettre d'identifier celui que l'on cherche.

Si cela ne suffit pas, il est possible de jeter un œil à l'organisation des partitions, en regardant le tableau qui apparaît dans la partie droite :

- 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* ou *ext4*;

- 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 *LUKS* ;
- si le disque à effacer contenait un système Windows, il doit y avoir une ou plusieurs partitions notées *ntfs* ou *fat32*.

Par ailleurs, le périphérique correspondant au disque interne est généralement le premier de la liste.

Une fois le disque trouvé et sélectionné, on pourra lire le chemin du disque dans la partie de droite en bas, à côté de l'étiquette *Périphérique*.

Le chemin du périphérique commence par `/dev/` suivi de trois lettres et d'un chiffre, les premières caractères étant dans la plupart des cas `sd`, `hd`, ou `mmcblk` : par exemple, `/dev/sdx1`. Noter le chemin quelque part, sans le chiffre (par exemple `/dev/sdx`) : il faudra l'écrire tout à l'heure à la place de `LE_PÉRIPHÉRIQUE`.



Attention ! Ce chemin n'est pas nécessairement toujours le même. Il vaut mieux recommencer cette courte procédure après avoir redémarré l'ordinateur, branché ou débranché une clé USB ou un disque dur. Cela évitera les mauvaises surprises... comme perdre le contenu d'un *autre* disque dur.

17.6.2 Lancer la commande `shred`

Ouvrir un *Terminal* : ouvrir la vue d'ensemble activités en appuyant sur la touche (ou sur un Mac), puis taper `term` et cliquer sur *Terminal*. [page 87]

Saisir la commande suivante en veillant à remplacer `LE_PÉRIPHÉRIQUE` par le chemin de périphérique déterminé précédemment :



```
pkexec 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 (ou `return`). Un mot de passe est demandé, car cette commande nécessite les priviléges d'administration, le saisir. La commande `shred` va alors écrire dans le terminal ce qu'elle fait (puisque l'on le lui a demandé 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/sdb: pass 1/3 (random)...
shred: /dev/sdb: pass 2/3 (random)...
shred: /dev/sdb: pass 3/3 (random)...
```

À la fin de la procédure, le terminal affiche à nouveau le signe `$`, qui symbolise l'invite de commande. On peut alors fermer le terminal.

17.6.3 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 table de partitions, ni système de fichiers. Pour le réutiliser, il est nécessaire de créer entièrement au moins une nouvelle partition et son système de fichiers, avec *Disques* par exemple. [page 17]

[page 17]

[page 17]

17.7 Rendre irrécupérables des données déjà supprimées

 *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

 *Durée : 5 minutes de préparation, puis de plusieurs minutes à plusieurs heures d'attente, selon la taille du disque à nettoyer et selon la méthode utilisée.*

Lorsque des fichiers ont déjà été effacés sans précautions particulières, les données qu'ils contenaient se trouvent toujours sur le disque. L'objectif de cette recette est de recouvrir les données qui subsisteraient, en écrasant l'espace libre d'un disque dur. Cette méthode ne supprime donc aucun fichier visible dans le navigateur de fichiers.



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 montrer tout l'espace libre au logiciel chargé d'y recouvrir les traces. Voir à ce sujet la première partie. Il ne faut pas non plus faire confiance à cette méthode pour une clé USB, les cartes SD ou disques SSD et préférer recouvrir plusieurs fois l'intégralité des données qu'elle contient.

Dans Tails

Le paquet `nautilus-wipe` est déjà installé par défaut dans Tails. Il nous suffit donc de consulter la documentation, en cliquant sur l'icône *Documentation de Tails* se trouvant sur bureau.

Dans le menu à droite, cliquer sur *Documentation*. Puis, dans l'index qui s'ouvre, chercher la section *Chiffrement et vie privée* et cliquer sur la page *Effacer des fichiers de façon sécurisée et nettoyer l'espace disque avec Nautilus Wipe*.

Avec une Debian chiffrée

Si ce n'est pas déjà fait, il nous faut installer le paquet, voir page 122 `nautilus-wipe`, puis redémarrer l'ordinateur.

Il nous faut ensuite ouvrir un navigateur de fichiers, puis naviguer jusqu'au disque que l'on veut nettoyer. Effectuez ensuite un clic-droit dans la partie droite du navigateur de fichiers et sélectionnez *Écaser l'espace disque disponible*. Une fenêtre s'ouvre nous demandant de confirmer la suppression de l'espace disque disponible, et proposant également quelques *Options*.

Nous pouvons choisir le nombre de passes effectuées afin de recouvrir les données de notre périphérique, ainsi que quelques options de comportement lors de l'effacement des données. Les options par défaut sont suffisantes pour les disques durs actuels.

Cliquer ensuite sur *Écaser l'espace disque disponible*. L'effacement peut prendre du temps. Dans certains cas, le mot de passe d'administration est demandé.

Remarquons qu'un dossier appelé `tmp.XXXXXXXXXX` est créé dans le dossier. Nautilus Wipe va créer un fichier à l'intérieur et en augmenter la taille autant que possible, afin d'utiliser tout l'espace libre disponible, puis l'écrasera de manière sécurisée. Une fois l'effacement terminé, une fenêtre *L'écrasement a réussi* s'ouvre, précisant que *L'espace disque disponible sur la partition ou le périphérique « ... » a été écrasé avec succès*.

Partitionner et chiffrer un disque dur

Nous allons maintenant aborder le chiffrement d'un périphérique, afin d'y stocker des données de manière chiffrée.

Une fois un disque 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, consulter la partie sur la cryptographie.

[page 39]

Une fois la phrase de passe saisie, 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 dans lesquels on a suffisamment confiance.

[page 53]

En effet, non seulement ceux-ci auront accès aux données déchiffrées, mais des traces de la présence du disque dur seront également gardées sur l'ordinateur. C'est pourquoi nous vous conseillons de l'utiliser sur un système GNU/Linux chiffré ou un système live amnésique.

[page 19]

[page 107]

[page 101]

Il peut s'agir d'un disque dur, d'un disque SSD, d'une clé USB, d'une carte SD ou encore d'une partie seulement d'un de ces périphériques. On peut en effet découper un disque dur ou une clé USB en plusieurs morceaux indépendants, qu'on appelle des partitions.

[page 17]

Ci-dessous, on parlera de disque, sachant que, sauf mention contraire, le terme vaut aussi bien pour un disque dur interne qu'un disque dur externe, ou que tout type de périphérique à mémoire flash, comme une clé USB, un disque SSD, ou une carte SD.

Si on veut avoir un endroit sur le disque où mettre des données qui ne seront pas confidentielles, et auxquelles on pourra accéder sur des ordinateurs non dignes de confiance, il est possible de découper le disque en deux partitions :

1. 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 ;
2. une partition chiffrée, avec les données confidentielles, qu'on ouvre que sur les ordinateurs auxquels on fait confiance.

18.1 Chiffrer un disque avec LUKS et dm-crypt

On va expliquer comment chiffrer un disque avec les méthodes standards sous GNU/Linux, appelées `dm-crypt` et LUKS. Ce système est maintenant bien intégré avec les environnements de bureau, et la plupart des opérations sont donc possibles sans avoir besoin d'outils particuliers.

18.2 D'autres logiciels que l'on déconseille

Il existe d'autres logiciels de chiffrement comme FileVault¹, qui est intégré dans Mac OS X ou BitLocker² pour Windows — mais il s'agit de logiciels propriétaires — ou encore VeraCrypt [<https://veracrypt.codeplex.com/>]. Cependant, 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.

18.3 En pratique

Si le disque a déjà servi, il peut être une bonne idée de commencer par recouvrir ses données, voir page 127.

Si le disque à chiffrer ne dispose pas d'espace libre, le formater, voir page ci-contre.

Ensuite, si l'on souhaite chiffrer une partie seulement du disque, il faut créer une partition en clair, voir page 136.

À la suite de quoi il ne reste plus qu'à l'initialiser pour contenir des données chiffrées, voir page 136.

Le voilà enfin prêt à être utilisé, voir page 137.

1. Une des dernières analyses récentes de FileVault [https://www.cl.cam.ac.uk/~osc22/docs/slides_fv2_ifip_2013.pdf] date de 2012. En plus d'être sensible aux mêmes attaques que d'autres systèmes, FileVault a quelques faiblesses qu'il faut préciser : la phrase de passe de chiffrement est souvent identique au mot de passe de la session, généralement faible ; le fait d'enregistrer un « mot de passe principal » ouvre un nouveau champ d'attaques. Néanmoins, en gardant en tête que cela offre un niveau de protection limité, cela vaut tout de même la peine d'activer FileVault sur un ordinateur avec Mac OS X.

2. Wikipédia, 2017, *BitLocker Drive Encryption* [https://fr.wikipedia.org/wiki/BitLocker_Drive_Encryption].

18.4 Préparer un disque à chiffrer

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : environ 10 minutes.

Ci-dessous, on parlera toujours de disque, sachant que ça vaut aussi bien pour un disque interne ou externe que pour une clé USB, une carte SD ou un disque SSD, sauf si on précise le contraire.

La procédure que l'on explique ici implique d'effacer toutes les données qui se trouvent sur le disque³. Si l'on a déjà de l'espace non partitionné sur son disque, on peut directement passer à l'étape de chiffrage.

[page suivante]

18.4.1 Installer les paquets nécessaires

Pour chiffrer notre disque, on a besoin d'avoir installé les paquets `secure-delete`, [\[page 122\]](#) `dosfstools` et `cryptsetup`. Si l'on utilise Tails, ces paquets sont déjà installés.

18.4.2 Formater le disque avec l'Utilitaire de disque

Pour ouvrir l'application *Disques* à partir de la vue d'ensemble des Activités : taper sur la touche  () sur un Mac), puis taper `disque` et cliquer sur *Disques*.

Dans la fenêtre de *Disques* la partie de gauche liste les disques connus du système ; la partie de droite permet d'effectuer des actions.

Choisir le périphérique

À gauche, il y a la liste des disques. Si l'ordinateur utilisé contient un système chiffré, il y a aussi les volumes chiffrés de notre système.

Les icônes, la taille indiquée ainsi que le nom des disques devraient permettre d'identifier celui que l'on cherche.

Une fois le disque repéré, le sélectionner dans la liste. Les informations qui s'affichent dans la partie droite de la fenêtre doivent permettre de confirmer qu'on a sélectionné le bon disque.

Démonter les volumes

Si le volume est monté, une icône carrée est visible dans la partie de droite, sous la représentation graphique du disque dans la rubrique *Volumes*. Cliquer sur ce bouton afin de démonter le volume. Si ce disque contient plusieurs volumes, les sélectionner un par un dans la rubrique *Volumes* et les démonter de la même façon si besoin.

Reformater le disque

⚠ **Attention !** Formater un disque revient à supprimer tous les fichiers qui s'y trouvent.

Dans la barre supérieure du logiciel, cliquer sur l'icône , puis sur *Formater le disque...*. Une fenêtre s'ouvre et propose d'effacer ou non les données sur le support, et de formater le disque. En fonction du contexte, et des limites déjà abordées précédemment à ce sujet, choisir si les données doivent être effacées ou non. Laisser *Compatible avec tous les systèmes et périphériques* dans *Partitionnement*, puis cliquer sur le bouton *Formater...*

[page 34]

³. On pourrait également utiliser le logiciel GParted. Plus difficile à utiliser que l'*Utilitaire de disque*, ce dernier a l'avantage de savoir *redimensionner* une partition déjà existante tout en gardant les fichiers qui s'y trouvent.

Disques demande si l'on veut vraiment formater le périphérique. C'est le moment de vérifier que l'on a choisi le bon périphérique avant de faire une bêtise. Si c'est bien le cas, confirmer en cliquant sur *Formater*.

18.5 Créer une partition non chiffrée

⌚ *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

⌚ *Durée : 2 minutes.*

Si on le souhaite, c'est le moment de créer une partition non chiffrée où l'on met des données qui ne sont pas confidentielles, et que l'on peut utiliser depuis tous les ordinateurs sans avoir à taper de phrase de passe.

[plus bas] Si l'on désire chiffrer le disque en entier, on peut directement passer à l'étape suivante.

Toujours dans *Disques* avec le disque sélectionné, dans la partie droite cliquer sur la zone *Espace disponible* du schéma des *Volumes*. En-dessous, cliquer ensuite sur le symbole **+**.

Choisir la taille voulue pour la partition non chiffrée dans le champs dédié. L'espace laissé libre nous servira pour la partition chiffrée. Dans *Type*, choisir *Compatible avec tous les systèmes et périphériques (FAT)*. On peut aussi choisir un nom pour la partition. Une fois cela fait, cliquer sur *Créer*.

18.6 Créer une partition chiffrée

⌚ *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

⌚ *Durée : 10 minutes + quelques minutes à plusieurs heures pour remplir l'espace libre, selon la taille de la partition.*

18.6.1 Créer la partition chiffrée

Toujours dans *Disques* avec le disque sélectionné, dans la partie droite cliquer sur la zone *Espace disponible* du schéma des *Volumes*. En-dessous, cliquer ensuite sur le symbole **+**.

[page 93] Dans la section *Type*, choisir *Chiffré, compatible avec les systèmes Linux (LUKS + Ext4)*, puis entrer une bonne phrase de passe pour le volume chiffré dans les deux champs appropriés. Il est aussi possible de donner un nom à ce volume. Enfin, cliquer sur *Créer*.

18.6.2 Remplir la partition 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 nos propres données, et complique donc la vie des personnes qui voudraient tenter de les déchiffrer.

Sur le schéma des *Volumes*, repérer *Partition [...] LUKS* et sélectionner le *Système de Fichiers* situé en dessous. Sous le schéma, cliquer sur ►.

En bas de la fenêtre, dans *Contenu*, un lien apparaît après *Monté sur*. Cliquer sur ce lien pour ouvrir le dossier, puis suivre l'outil servant à rendre irrécupérables des données déjà supprimées.^[page 132]

Le processus dure de 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).

18.6.3 Débrancher *correctement* le disque dur

Dans le navigateur de fichiers, cliquer sur le symbole , puis débrancher physiquement le disque (le cas échéant).

Le disque chiffré est maintenant utilisable.

18.7 Utiliser un disque dur chiffré

C *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

⌚ *Durée : 2 minutes, quelques heures... ou jamais, selon notre capacité à retenir la phrase de passe.*

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...

18.7.1 Avec Debian (ou autre GNU/Linux)

Sur un système GNU/Linux avec un environnement de bureau configuré pour ouvrir automatiquement les médias externes, une fenêtre apparaît pour demander la phrase de passe lorsqu'on branche un disque externe contenant des données chiffrées.

Si ce n'est pas le cas, elle apparaîtra quand on demandera au système d'ouvrir la partition chiffrée, par exemple à partir de *Fichiers*.

Pour fermer la partition chiffrée, il suffit de démonter le disque comme on le fait habituellement.

18.7.2 Avec d'autres systèmes

Nous ne connaissons pas de moyen simple d'accéder à la partition chiffrée du disque ni sous Windows, ni sous Mac OS X. Même si des solutions peuvent exister⁴, il est bon de rappeler qu'il s'agit de systèmes d'exploitation propriétaires, en lesquels il n'y a aucune raison d'avoir confiance.^[page 31]

Alors le mieux à faire, pour mettre sur le disque 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 le disque, comme expliqué précédemment.^[page précédente]

4. Pour les anciennes versions de Windows (jusqu'à Vista), il était possible d'utiliser FreeOTFE (<https://sourceforge.net/projects/freeotfe.mirror/>).

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, sur un autre support de stockage que celui où se trouvent les données.

Bien entendu, si on prend le soin de mettre nos données de travail sur des disques durs ou des clés USB chiffrés, il est nécessaire que ces copies soient chiffrées, elles aussi.

Deux autres points à ne pas négliger pour mettre en place une bonne *politique de sauvegardes* :

- définir une méthode pour effectuer **régulièrement** ses sauvegardes,
- tester de temps à autre si les sauvegardes sont toujours bien lisibles.

Ce dernier aspect est vraiment à ne pas négliger. Perdre les données originales est souvent pénible. S'apercevoir ensuite que les sauvegardes ne permettent pas de *restaurer* ce qu'on a perdu transforme la situation en catastrophe.

Dans le même ordre d'idée, cela paraît également une bonne idée de ne pas stocker les sauvegardes au même endroit que les données originales. Sinon, on risque que les deux soient perdues ou détruites simultanément...

19.1 Gestionnaire de fichiers et stockage chiffré

Réaliser des sauvegardes est avant tout une question de rigueur et de discipline. Dans les cas simples, on peut se passer de logiciels spécialement prévus pour réaliser des sauvegardes, et se contenter simplement d'effectuer des copies avec le gestionnaire de fichiers.

19.1.1 Effectuer les sauvegardes

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : pour la première fois, le temps de chiffrer le support de stockage et de décider des fichiers à sauvegarder ; et ensuite cela dépend de la quantité de données à sauvegarder.

Le chiffrement de nos sauvegardes sera assuré par le chiffrement du support de stockage externe (clé USB ou disque dur). [page 133]

Pour effectuer les copies avec régularité et sans trop y passer de temps, il est recommandé :

- d'avoir quelque part une liste des fichiers et dossiers à sauvegarder ;

- de se fabriquer un petit calendrier des jours ou semaines où l'on fera ses sauvegardes, avec des cases que l'on cochera après les avoir faites.

Une bonne pratique consiste à créer un dossier avec la date de la sauvegarde pour y copier les données. Cela permet de garder facilement plusieurs sauvegardes si on le souhaite, et de supprimer tout aussi facilement les sauvegardes précédentes.

19.1.2 Restaurer une sauvegarde

C *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

⌚ *Durée : selon la quantité de données à restaurer.*

En cas de perte des données originales, la restauration se fait aussi simplement que la sauvegarde : en effectuant des copies dans l'autre sens.

19.1.3 S'assurer que les sauvegardes sont toujours lisibles

C *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

⌚ *Durée : environ 5 minutes + attendre que la vérification se fasse.*

Si l'on a effectué nos sauvegardes sur un support de stockage externe, il faut commencer par le brancher sur l'ordinateur.

La méthode évidente pour s'assurer que les sauvegardes sont toujours lisibles est sans doute de simuler une restauration. Procéder ainsi a un inconvénient... de taille : il faut avoir assez d'espace libre à notre disposition pour recopier l'ensemble des données sauvegardées vers un dossier temporaire que l'on supprime ensuite.

Voici une autre méthode, peut-être moins facile à mettre en œuvre, mais qui n'a pas cette contrainte. Elle nécessite d'utiliser un *Terminal*.

On va commencer la commande en tapant (**sans faire ↵** (ou **return**)) :



`find`

Ajouter ensuite un espace. Puis il faut indiquer le dossier contenant les sauvegardes, ce que l'on va faire avec la souris, en attrapant l'icône du dossier et en l'amenant sur le terminal. Après avoir relâché le bouton, ce qui est affiché doit ressembler à :



`find '/media/externe/sauvegardes'`

Il faut ensuite taper la fin de la commande pour que le tout ressemble à :



```
find '/media/externe/sauvegardes' -type f -print0 | xargs -0
↪      cat > /dev/null
```

La lecture se lance dès qu'on a appuyé sur **↵** (ou **return**). La ligne suivante devrait rester vide jusqu'à la fin de l'opération.

Après de la patience et le retour du \$ de l'invite de commande, on peut fermer le terminal.

Si des messages d'erreurs sont apparus dans l'intervalle, tels que « *Erreur d'entrée/-sortie* » ou « *Input/output error* », cela indique que la sauvegarde est corrompue. En règle générale, il faut alors se débarrasser du support (CD ou DVD, clé USB ou disque dur), en prendre un autre et refaire une nouvelle sauvegarde.

Note : ces deux méthodes partagent le défaut de ne pas vérifier l'intégrité des données. Mettre en place un mécanisme pour le faire est difficile sans recourir à des logiciels de sauvegarde plus complexes.

[page 44]

19.2 En utilisant *Déjà Dup*

⌚ Durée : 5 minutes pour installer le logiciel.

On peut également préférer utiliser un logiciel spécialisé dans la réalisation des sauvegardes. L'un d'entre eux, baptisé « *Déjà Dup* », a comme avantages d'être facile à utiliser, et de réaliser des sauvegardes chiffrées. Ces sauvegardes sont également « incrémentales », c'est-à-dire que les fichiers inchangés depuis la sauvegarde précédente ne sont pas copiés une nouvelle fois, et qu'il est possible d'accéder aux fichiers tels qu'ils étaient à chacune des sauvegardes.

Ce qui le rend aussi simple peut être aussi une limite : il ne sait gérer qu'une seule configuration à la fois. On ne peut donc pas sauvegarder des dossiers différents sur des supports différents à des fréquences différentes. C'est surtout l'outil idéal pour sauvegarder l'essentiel du contenu de son dossier personnel, mais pas beaucoup plus.

Il n'est pas livré avec l'environnement par défaut, il est donc nécessaire d'installer le paquet `Debian déjà-dup` pour pouvoir s'en servir.

[page 122]

19.2.1 Effectuer une sauvegarde

⌚ Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : 15 minutes environ pour la configuration, de quelques minutes à plusieurs heures pour la sauvegarde, selon la taille de ce qu'on veut copier.

Ouvrir *Sauvegardes* à partir de la vue d'ensemble des Activités : taper sur la touche  () sur un Mac), puis taper `sauv` et cliquer sur *Sauvegardes*.

Au lancement, l'interface ne correspond pas à ce qui devrait être affiché. Pour y remédier, dans le menu de gauche, cliquer sur *Dossiers à enregister* puis, toujours dans le menu de gauche, sur *Vue d'ensemble*. Maintenant l'interface correspond à *Vue d'ensemble*.

Dans la *Vue d'ensemble*, deux boutons nous accueillent, l'un pour *Restaurer* des fichiers d'une sauvegarde existante, l'autre pour *Démarrer la sauvegarde*.

Mais avant d'effectuer une sauvegarde, il faut configurer ce qu'on veut sauvegarder et où.

1. Dans le menu de gauche, cliquer sur *Dossiers à enregistrer*, pour voir la liste des dossiers à inclure dans la sauvegarde. Laisser le *Dossier personnel*, qui est suffisant dans la plupart des cas.
2. Dans *Dossiers à ignorer*, on peut rajouter les dossiers à ignorer contenant des fichiers souvent volumineux mais plus faciles à retrouver, comme *Vidéos* ou *Musique*.

- [page 93]
3. Dans *Emplacement de stockage* choisir l'emplacement de la sauvegarde. Pour stocker les sauvegardes sur un disque externe, brancher le disque en question à l'ordinateur, puis dans la liste *Emplacement de Stockage* cliquer sur *Dossier local* puis sélectionner le périphérique voulu... .
 4. Si l'on désire effectuer ces sauvegardes automatiquement, cliquer sur *Planification* pour choisir la fréquence de sauvegarde et la durée de conservation des données.
 5. On peut maintenant retourner à la *Vue d'ensemble*. Dans le cas où l'on a choisi une sauvegarde automatique, activer le curseur situé dans la barre de titre. Sinon, cliquer sur *Démarrer la sauvegarde*. Une nouvelle fenêtre s'ouvre, nous demandant une *phrase de passe* pour chiffrer¹ notre nouvelle sauvegarde. une fois la phrase de passe confirmée, cliquer sur *continuer* pour démarrer la sauvegarde.
 6. On peut maintenant fermer *Sauvegardes*. Lors de la prochaine sauvegarde, si les paramètres ne sont pas changés, il suffira de donner la phrase de passe de la sauvegarde pour que cette dernière soit mise à jour.

On peut modifier plus tard tous ces paramètres en redémarrant *Sauvegardes*.

Lorsque la planification des sauvegardes est activée et que le temps indiqué depuis la précédente sauvegarde est écoulé, *Déjà Dup* affiche un message de notification pour nous signifier qu'il effectuera la prochaine sauvegarde dès que le support externe sera de nouveau branché sur l'ordinateur. Et dès que ce sera le cas, une fenêtre s'ouvrira automatiquement pour demander de saisir la *phrase de passe* nécessaire pour mettre à jour la sauvegarde.

19.2.2 Restaurer une sauvegarde

 *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

 *Durée : 5 minutes pour configurer, de quelques minutes à plusieurs heures pour la restauration, selon la taille de nos sauvegardes.*

Ouvrir *Sauvegardes* à partir de la vue d'ensemble des Activités : appuyer sur la touche  ( sur un Mac), puis taper **sauv** et cliquer sur *Sauvegardes*.

L'opération de restauration démarre en cliquant sur le bouton *Restaurer*.

Si c'est la première fois qu'on utilise *Sauvegardes* (par exemple pour restaurer son dossier personnel après la perte d'un disque dur), il nous demande d'indiquer le dossier où ont été effectuées les sauvegardes. Sinon, il utilise le dossier déjà configuré.

Après un court délai, *Sauvegardes* nous demande de choisir, avec sa date, la sauvegarde à restaurer. Cliquer ensuite sur *Suivant*.

Il faut ensuite indiquer le dossier où seront écrits les fichiers issus de la sauvegarde. On peut soit *Restaurer les fichiers vers leurs emplacements d'origine* (ce qui remplace éventuellement des fichiers par la version qui se trouvait dans la sauvegarde), soit *Restaurer vers un dossier spécifique* à indiquer. Cliquer ensuite sur *Suivant*.

Pour finir, un dernier écran résume les paramètres de cette restauration. Après avoir cliqué sur *Restaurer*, une fenêtre s'ouvre, si nécessaire, pour demander le mot de passe du superutilisateur (par exemple, pour restaurer les permissions de certains fichiers).

1. Si le support externe est chiffré, on peut éventuellement décider de ne pas chiffrer les fichiers sauvegardés. Cela fait une phrase de passe de moins à inventer et à retenir. On perd néanmoins la possibilité de compartimenter les accès, au cas où le support externe servirait à d'autres choses que les sauvegardes.

Si la sauvegarde a été chiffrée, *Déjà Dup* nous demande ensuite la phrase de passe. Une fois que l'on aura cliqué sur *Continuer*, l'écriture des fichiers en provenance de la sauvegarde commencera pour de bon.

19.2.3 S'assurer que les sauvegardes sont toujours lisibles

C *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

 *Durée : de quelques minutes à plusieurs heures, selon la taille de nos sauvegardes.*

Le fonctionnement incrémental de *Déjà Dup* assure déjà superficiellement que les sauvegardes précédentes soient lisibles. Néanmoins cela ne constitue pas une garantie...

Malheureusement, la meilleure méthode actuellement disponible avec *Déjà Dup* pour s'assurer que l'on peut restaurer ses sauvegardes est... de faire une restauration vers un dossier temporaire que l'on effacera après. C'est loin d'être pratique, vu qu'il faut avoir accès à un disque dur chiffré suffisamment grand.

On peut toutefois s'assurer que les fichiers contenant les sauvegardes restent lisibles en utilisant les mêmes méthodes que celles décrites précédemment.

[page 140]

Partager un secret

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

 Durée : une heure environ.

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 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 réunissant quelques-uns¹.

20.1 Partager une phrase de passe

L'usage le plus pratique est de partager comme secret la phrase de passe d'un support chiffré. [page 133]

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]

20.1.1 Installer le paquet nécessaire

Pour réaliser le partage du secret, on utilisera le programme `ssss-split`. On trouve ce programme parmi ceux fournis par le système live *Tails*, cependant, pour en disposer sur une Debian chiffrée, il est nécessaire d'installer le paquet `Debian_ssss`. [page 122]

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]

20.1.2 Générer une phrase de passe 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 en tapant la commande :



```
head -c 32 /dev/random | base64
```

1. Pour plus de détails, voir l'article de Wikipédia sur les secrets répartis [https://fr.wikipedia.org/wiki/Secret_r%C3%A9parti].

L'ordinateur va répondre quelque chose comme :

```
7rZw00u+8v1stea980uyU1efwNzHaKX9CuZ/TK0bRWY=
```

Si l'on désire une phrase de passe de plus ou moins de 32 caractères, il suffira de remplacer 32 par le nombre de caractères désirés. Sélectionner cette ligne à l'aide de la souris et la copier dans le presse-papiers (*via* le menu *Édition → Copier*).

20.1.3 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 `NOMBRE_DE_MORCEAUX_NECESSAIRES` est le nombre de morceaux qu'il sera nécessaire de réunir pour retrouver la phrase de passe de départ. Le `NOMBRE_DE_MORCEAUX_TOTAL` correspond au nombres de morceaux en lesquels la phrase de passe sera découpée. 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 `↵` (ou `return`) 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-b8d576a1a8091760b18f125e12bb6f2b1f2dd9d93f7072ec69b129b27bb8e
↪      97536ea85c7f6dceeb7b4399ea49
2-af83f0af05fc207e3b466caef30ec4d39c060800371feab93594350b7699a
↪      8db9594bfc71ed9cd2bf314b738
3-4718cb58873dab22d24e526931b061a6ac331613d8fe79b2172213fa767ca
↪      a57d29a6243ec0e6cf77b6cbb64
4-143a1efcde7f4f5658415a150fcac6da04f697ebfeb9427b59dca57b50ec7
↪      55510b0e57ccc594e6b1a1eeb04
5-fca1250b5cbe40ab14964d2cd7463af34c389f81158d1707b6a838a50097
↪      7d957be38f83e8eefb79266e74a
6-ebf7a305f14bf3143b801a222cc1c857b7e8582119374925274f9f335d283
↪      677f4c002f8d68bcce722ebba1f
```

20.1.4 Créer le support chiffré

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.

[page 133]

20.2 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.

Cette étape doit idéalement être faite à partir d'un système *live* afin de ne pas laisser de traces du secret partagé.

[page 101]

20.2.1 Installer les paquets nécessaires

Comme précédemment, si le programme n'est pas disponible sur le système on aura besoin d'avoir installé le paquet `ssss` et d'avoir ouvert un terminal.

[page 122]

20.2.2 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 :



```
sssss-combine -t NOMBRE_DE_MORCEAUX_A_DISPOSITION
```

Le programme demande ensuite de saisir les morceaux à notre disposition. Il faut taper `↵` (ou `return`) 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]: 4-143a1efcde7f4f5658415a150fcac6da04f697ebfeb9427b
→ 59dca57b50ec755510b0e57ccc594e6b1a1eeb04
Share [2/3]: 2-af83f0af05fc207e3b466caef30ec4d39c060800371feab9
→ 3594350b7699a8db9594bfc71ed9cd2bf314b738
Share [3/3]: 6-ebf7a305f14bf3143b801a222cc1c857b7e8582119374925
→ 274f9f335d283677f4c002f8d68bcce722ebba1f
Resulting secret: 7rZw00u+8v1stea980uyU1efwNzHaKX9CuZ/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]: 4-143a1efcde7f4f5658415a150fcac6da04f697ebfeb9427b
→ 59dca57b50ec755510b0e57ccc594e6b1a1eeb04
Share [2/3]: 2-af83f0af05fc207e3b466caef30ec4d39c060800371feab9
→ 3594350b7699a8db9594bfc71ed9cd2bf31ab738
Share [3/3]: 6-ebf7a305f14bf3143b801a222cc1c857b7e8582119374925
→ 274f9f335d283677f4c002f8d68bcce722ebba1f
Resulting secret: .....L.fm.....6 .....v..w.a....[....zS.....
WARNING: binary data detected, use -x mode instead.
```

20.2.3 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.

Utiliser les sommes de contrôle

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : 5 à 10 minutes.

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 Betty dans une lettre que le programme que cette dernière peut télécharger sur son site a pour somme de contrôle SHA256 171a0233a4112858db23621dd5ffa31d269cbdb4e75bc206ada58ddab444651f et que le fichier qu'elle a reçu a la même somme de contrôle, il est quasiment sûr que personne n'a falsifié le programme en chemin, et elle peut exécuter le programme sans trop de craintes.

[page 44]

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.

21.1 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.

[page 44]

Il est possible d'utiliser un outil graphique tout aussi bien qu'un terminal pour effectuer de tels calculs, nous n'allons cependant pas détailler l'utilisation d'un terminal.

21.1.1 Installer les logiciels nécessaires

Si le paquet `nautilus-gtkhash` n'est pas encore installé, l'installer, puis redémarrer l'ordinateur. Ce paquet est installé par défaut dans *Tails*.

[page 122]

21.1.2 Calculer la somme de contrôle

Ouvrir *Fichiers* à partir de la vue d'ensemble des Activités : appuyer sur la touche  () sur un Mac), puis taper `fich` et cliquer sur *Fichiers*.

Sélectionner le fichier pour lequel on veut obtenir des sommes de contrôle, puis effectuer un clic-droit dessus. Dans le menu contextuel qui apparaît, choisir *Propriétés*, puis aller dans l'onglet *Résumés*.

De nombreuses *Fonctions de hachage* sont proposées, avec trois sélections par défaut, MD5, SHA1, SHA256. Si une autre somme de contrôle que celles-ci est nécessaire, cocher la case correspondante. Cliquer sur *Hachage*. Les sommes de contrôles apparaissent alors dans la colonne *Résumé*.

21.2 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.

De la même manière, pour permettre à d'autres personnes de vérifier l'intégrité d'un fichier, on leur fera parvenir la somme de contrôle selon les mêmes méthodes.

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.

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.

[plus bas]

Installer et utiliser un système virtualisé

L'objectif de cet ensemble de recettes est d'utiliser un système d'exploitation virtuel (appelé *invité*) à l'intérieur de notre système GNU/Linux (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 expliquant comment travailler sur un document sensible sous Windows.

[page 70]



Attention : nous conseillons dans les éditions précédentes de ce *Guide* d'utiliser le logiciel *VirtualBox*, seulement celui-ci n'est plus disponible dans Debian. Si l'on utilisait cet outil auparavant, il faudra soit réinstaller notre machine virtuelle, soit la migrer de *VirtualBox* au *Gestionnaire de machine virtuelle*. Cette procédure n'est pas documentée dans ce *Guide*, mais on pourra toujours se lancer dans l'aventure en suivant des instructions disponibles sur le web¹.

1. Malte Gerken, 2017, *Migrate a VM from VirtualBox to libvirt* [<https://maltegerken.de/blog/2017/01/migrate-a-vm-from-virtualbox-to-libvirt/>] (en anglais).

22.1 Installer le *Gestionnaire de machine virtuelle*

 Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

 Durée : un quart d'heure environ.

22.1.1 Principe

L'objectif de cette recette est d'installer le *Gestionnaire de machine virtuelle*, logiciel qui nous permettra de faire fonctionner un système Windows virtuel à l'intérieur de notre système Debian GNU/Linux.

22.1.2 Installer le *Gestionnaire de machine virtuelle*

 page 122 L'étape suivante est donc d'installer le paquet `virt-manager`.

22.1.3 Vérifier l'installation

Pour lancer le *Gestionnaire de machine virtuelle*, ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper `virt` et cliquer sur *Gestionnaire de machine virtuelle*. Notre mot de passe d'administration nous est alors demandé, c'est normal.

22.2 Installer un *Windows* virtualisé

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : 20 minutes environ, plus le temps d'installer Windows (de 30 minutes à plus d'une heure).

Avant toute chose, se munir d'un CD d'installation de la version de Windows appropriée, et l'insérer dans le lecteur CD/DVD. Si une fenêtre affichant le contenu du CD s'ouvre automatiquement, la refermer ou l'ignorer.

22.2.1 Créer une nouvelle machine virtuelle

Pour lancer le *Gestionnaire de machine virtuelle*, ouvrir la vue d'ensemble des Activités en appuyant sur la touche (sur un Mac), puis taper **virt** et cliquer sur *Gestionnaire de machine virtuelle*.

Le programme démarre, il faut alors donner le mot de passe demandé et s'authentifier. Cliquer sur le menu *Fichier* puis *Nouvelle machine virtuelle* et suivre les 5 étapes de l'assistant. À la fin de chaque étape le bouton *Suivant* permet de passer à l'étape suivante.

- Étape 1 : Sélectionner *Média d'installation local (image ISO ou CD-ROM)*.
- Étape 2 : Utiliser un *CD-ROM ou un DVD* devrait être automatiquement coché, dans le cas contraire le sélectionner. Si le *Type* de l'OS et sa version sont mal reconnus, décocher *Déetecter automatiquement le système d'exploitation en se basant sur le média d'installation* pour les choisir manuellement.
- Étape 3 : Dans la fenêtre suivante, indiquer la taille de *Mémoire vive* et le nombre de *CPU* dédiés à la machine virtuelle. Voici les minimum recommandés pour les dernières versions de Windows :

Version	Mémoire (RAM)	CPU
Windows 7	1024 MiB	1
Windows 8	2048 MiB	1
Windows 10	2048 MiB	1

- Étape 4 : Choisir la taille de l'image virtuelle. Sachant qu'on veut accueillir tout un Windows, elle doit être conséquente : 20 GiB est un minimum.
- Étape 5 : Indiquer un *Nom* pour la machine virtuelle puis sélectionner *Personnaliser la configuration avant l'installation*.

Enfin cliquer sur *Terminer*.

Si un message *Le réseau virtuel n'est pas actif.* s'affiche, cliquer sur *Oui* pour le démarrer, sans quoi il n'est pas possible de passer à l'étape suivante.

Dans la fenêtre qui s'ouvre, sélectionner le matériel *NIC* qui gère l'interface réseau virtuelle, puis cliquer sur *Enlever*. Dans la fenêtre de confirmation, choisir *Oui*. La machine virtuelle est désormais isolée du réseau.

Ajouter ensuite un canal nécessaire au partage de dossier entre le système hôte et le système invité. Pour ce faire, cliquer sur le bouton en bas à gauche *Ajouter un matériel*. Dans la fenêtre qui s'affiche, cliquer sur *Canal* dans la liste de gauche. Dans la liste déroulante *Nom*, sélectionner *org.spice-space.webdav.0*, puis cliquer sur le bouton *Terminer*.

Cliquer sur *Commencer l'installation* pour lancer l'installation de Windows.

22.2.2 Installer Windows dans la machine virtuelle

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, qui dépend de notre version de Windows mais il faut préciser :

- Ne pas mettre d'informations personnelles lorsque le *Nom* et l'*Organisation* sont demandés. Mettre par exemple « user ».
- Pareil si l'on souhaite entrer un numéro de série de Windows, un lien pourrait être fait avec nous si celui-ci nous a été attribué officiellement.
- 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.

Une fois l'installation terminée, éteindre le Windows virtuel en cliquant sur le menu *Machine virtuelle* → *Éteindre* → *Éteindre*. Depuis la fenêtre de la machine virtuelle, cliquer sur le menu *Afficher* → *Détails*. Dans la liste de gauche, choisir *IDE CD-ROM 1*, puis derrière *Chemin de la source* cliquer sur *Déconnecter*.

Pour éjecter le CD/DVD, ouvrir la vue d'ensemble des Activités en appuyant sur la touche (sur un Mac), puis taper **f****i****c** et cliquer sur *Fichiers*. Dans la liste qui se trouve à gauche de cette fenêtre, repérer la ligne correspondant au CD/DVD et cliquer sur l'icône correspondante.

22.2.3 Préparer les outils invités pour le *Gestionnaire de machine virtuelle*

Des pilotes spécifiques permettent d'améliorer l'interaction entre le *Gestionnaire de machine virtuelle* et le système Windows invité grâce à une technologie appelée SPICE : il s'agit des outils invités et du service de partage de dossiers.

Depuis le système hôte, télécharger l'installateur Windows des outils invités pour SPICE [<https://www.spice-space.org/download/windows/spice-guest-tools/spice-guest-tools-latest.exe>]. Pour vérifier le fichier téléchargé, il est possible de télécharger sa signature [<https://www.spice-space.org/download/windows/spice-guest-tools/spice-guest-tools-latest.exe.sign>].

Aller ensuite sur la page web de l'installateur du service WebDAV pour SPICE [<https://www.spice-space.org/download/windows/spice-webdavd/>]. Cliquer sur le lien de téléchargement de la dernière version correspondant à l'architecture de notre machine virtuelle. Le nom se termine par *-x86-latest.msi* pour un Windows 32 bits ou *-64-latest.msi* pour un Windows 64 bits.

Pour transférer les deux installateurs de la machine hôte au Windows invité, on va préparer une image de disque au format *ISO* les contenant grâce au logiciel *Brasero*. Pour lancer *Brasero*, ouvrir la vue d'ensemble des Activités en appuyant sur la touche (sur un Mac), puis taper **b****r****a** et cliquer sur *Brasero*.

Choisir *Projet de données*. Cliquer sur l'icône et ajouter les fichiers précédemment téléchargés *spice-guest-tools* et *spice-webdavd*.

Dans la liste déroulante en bas de la fenêtre, choisir *Fichier image* puis cliquer sur *Graver*.... Choisir par exemple comme nom de fichier *outils invites pour Windows* et cliquer sur *Créer une image*. Une fois l'image créée, fermer *Brasero*.

22.2.4 Installer les outils invités pour le *Gestionnaire de machine virtuelle*

Retourner dans le *Gestionnaire de machine virtuelle* et partager l'image de disque ISO que l'on vient de créer avec la machine virtuelle en suivant la recette partager un CD

avec le système virtualisé.

Ensuite, toujours dans le *Gestionnaire de machine virtuelle*, démarrer la machine virtuelle en effectuant un clic-droit dessus, puis en cliquant sur *Ouvrir*. Dans la nouvelle fenêtre qui s'ouvre, aller dans le menu *Machine virtuelle* puis cliquer sur *Démarrer*.

Une fois à l'intérieur du système Windows virtuel, ouvrir le CD-ROM virtuel et double-cliquer sur le fichier **spice-guest-tools**. Chaque fois que Windows demande s'il faut autoriser un programme inconnu (c'est-à-dire non vérifié par Microsoft), l'accepter. Accepter aussi toutes les autres demandes du programme d'installation en cliquant sur *Suivant*.

Faire de même avec **spice-webdavd**.

Maintenant, il est possible de copier-coller du texte entre la machine hôte et la machine virtuelle et de copier-coller des fichiers de la machine hôte à la machine virtuelle Windows. Il est aussi devenu possible de modifier l'affichage de la machine virtuelle en fonction de la taille de la fenêtre qui accueille Windows. Cliquer sur le menu *Afficher* → *Mettre à l'échelle l'affichage* et sélectionner *Redimensionnement automatique de la machine virtuelle avec fenêtre*.

L'installation du Windows virtuel est maintenant terminée.

22.3 Prendre un instantané d'une machine virtuelle

 *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

 *Durée : 5 minutes.*

[page 70] Pour suivre la méthode permettant de travailler sur un document sensible sous Windows, on peut avoir besoin de sauvegarder l'état d'une machine virtuelle. Pour cela, on va utiliser la gestion des instantanés des machines virtuelles, appelés aussi *Snapshot* en anglais.

Ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper *virt* et cliquer sur *Gestionnaire de machine virtuelle* et entrer le mot de passe. Sélectionner la machine virtuelle souhaitée et cliquer sur *Ouvrir*. Si elle est en cours de fonctionnement, l'éteindre en cliquant sur le menu *Machine virtuelle* → *Éteindre* → *Éteindre*.

Cliquer sur *Afficher* → *Instantanés*. Dans la liste de gauche, cliquer sur le bouton en bas. Dans la fenêtre qui apparaît, indiquer le *Nom* de l'instantané, par exemple « Windows propre ». Ajouter éventuellement une *Description* puis cliquer sur *Terminer*.

22.4 Restaurer l'état d'une machine virtuelle à partir d'un instantané

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : selon la taille du disque.

L'objectif de cette recette est de restaurer l'état d'une machine virtuelle à partir d'un instantané créé précédemment. Ainsi, il sera possible de l'utiliser pour un nouveau projet, comme le recommande la méthode préconisée pour travailler sur un document sensible sous Windows.

[page précédente]
[page 70]

22.4.1 Afficher les instantanés

Commençons donc par ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper `virt` et cliquer sur *Gestionnaire de machine virtuelle* puis entrer le mot de passe. Sélectionner la machine virtuelle souhaitée et cliquer sur *Ouvrir*. Dans la nouvelle fenêtre, cliquer sur le menu *Afficher* et sélectionner *Instantanés*.

22.4.2 Choisir et restaurer un instantané

Sélectionner l'instantané souhaité à partir duquel restaurer l'état de la machine (par exemple « Windows propre »). Cliquer sur le bouton ►, en bas à gauche. Une nouvelle fenêtre apparaît, demandant si nous sommes sûrs de vouloir exécuter l'instantané sélectionné. Exécuter cet instantané aura pour conséquence que toutes les modifications effectuées dans la machine virtuelle depuis la création de cet instantané seront perdues. Si nous sommes sûres de notre choix, cliquer sur *Oui*, sinon, cliquer sur *Non*.

Le *Gestionnaire de machine virtuelle* va restaurer l'état de la machine virtuelle tel qu'il était au moment où l'instantané a été pris.

22.5 Partager un CD ou un DVD avec un système virtualisé

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : 10 minutes environ.

Cette étape est nécessaire pour installer dans le système Windows virtualisé des logiciels que l'on a sur CD/DVD ou sur une image de disque ISO.

Commencer par ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper `virt` et cliquer sur *Gestionnaire de machine virtuelle*. Enfin, entrer le mot de passe demandé.

Selectionner la machine virtuelle Windows avec laquelle on souhaite partager un CD ou un DVD. Afficher la vue détaillée de la machine virtuelle en cliquant sur le menu *Afficher → Détails*. Dans la liste des matériels à gauche, sélectionner *IDE CD-ROM 1* et cliquer sur le bouton *Se connecter*.

Il y a alors deux possibilités :

- si l'on souhaite partager un CD ou un DVD physique avec la machine virtuelle, insérer le CD ou le DVD dans le lecteur et attendre quelques instants. Choisir alors *CD-ROM ou DVD*.
- si l'on souhaite plutôt partager une image ISO, choisir *Emplacement de l'image ISO*, puis cliquer sur *Parcourir*. . . Dans la fenêtre qui s'affiche, choisir *Parcourir en local* et ouvrir l'image ISO.

Dans tous les cas, finir en choisissant *Valider*. Retourner à Windows avec *Afficher → Console*. Windows devrait alors détecter le CD inséré. Si ce n'est pas le cas, on peut aller le chercher dans le *Poste de travail*. Si ça ne marche pas du premier coup, recommencer l'opération.

Quand on a fini d'utiliser le CD dans Windows, l'éjecter depuis Windows, puis retourner dans la vue détaillée de la machine virtuelle avec *Afficher → Détails*, sélectionner *IDE CD-ROM 1* et cliquer sur le bouton *Déconnecter*.

22.6 Partager un dossier avec un système virtualisé

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : 10 minutes environ.

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.

⚠ **Attention** : en apprenant à utiliser ce système de partage, on pourrait peut-être vouloir le configurer pour donner accès à la totalité des disques 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

Ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper `fic` et cliquer sur *Fichiers*. Ensuite, choisir l'emplacement où on veut mettre ce dossier d'échange. Par exemple : dans le *Dossier personnel* cliquer sur le

bouton  puis sur le bouton *Créer un dossier* et lui donner un nom évocateur (« *Dossier lisible par Windows* » ou « *Dossier où Windows peut écrire* », par exemple). C'est dans ce dossier qu'il faudra mettre les fichiers que l'on veut transférer au Windows.

Installer l'afficheur distant

Actuellement, le *Gestionnaire de machine virtuelle* ne permet pas l'activation du partage de dossiers. Il est nécessaire d'utiliser le logiciel *Afficheur distant*. L'étape suivante est donc d'installer le paquet `virt-viewer`.

[page 122]

Activer le partage du dossier

Pour activer le partage de dossier, il faut d'abord démarrer la machine virtuelle Windows depuis le *Gestionnaire de machine virtuelle*.

Pour accéder au *Gestionnaire de machine virtuelle*, ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper `virt` et cliquer sur *Gestionnaire de machine virtuelle*. Entrer le mot de passe demandé.

Dans la fenêtre du *Gestionnaire de machine virtuelle*, effectuer un clic-droit sur la machine virtuelle désirée (par exemple, *Windows propre*) et cliquer sur *Démarrer*.

La machine virtuelle démarre alors, mais son écran n'est pas visible. Nous utiliserons l'*Afficheur distant* pour y accéder.

Ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper `affi` et cliquer sur *Afficheur distant*. La première fois, il faut entrer l'adresse de la machine virtuelle dans le champ *Adresse de la connexion*. Généralement, l'adresse est : `spice://localhost:5900`. Si plus d'une machine virtuelle est en cours de fonctionnement la première démarrée aura l'adresse `spice://localhost:5900`, la deuxième `spice://localhost:5901` et ainsi de suite. À partir de la deuxième fois, il est possible de cliquer sur l'adresse désirée dans *Connexions récentes*. Cliquer sur le bouton *Connecter*.



Attention : avant de cocher la case *Partager le dossier*, 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.

Depuis la fenêtre, de *Afficheur distant*, contenant la machine virtuelle Windows , cliquer sur le menu *Fichier → Préférences*. Dans la fenêtre qui s'affiche, sélectionner le dossier que l'on souhaite partager grâce au bouton situé à droite. Pour sélectionner le *Dossier lisible par Windows* il est nécessaire de choisir *Autre...* dans le menu déroulant. Cocher la case *Partager le dossier*.



Toujours cocher la case *Lecture seule* sauf si l'on souhaite faire sortir des fichiers du Windows virtualisé, auquel cas on donnera un nom explicite comme *Dossier où Windows peut écrire* au dossier partagé.

Copier les fichiers

Dans la machine virtuelle Windows, après un petit moment, le disque réseau *Z* devrait être accessible depuis l'explorateur de fichiers, sous *Poste de travail*. Sinon, on peut tenter dans l'ordre de :

- cliquer sur actualiser dans l'explorateur de fichiers ;
- fermer et ré-ouvrir l'explorateur de fichiers ;
- et enfin redémarrer Windows.

Une fois le disque *Z* visible, c'est-à-dire le dossier que nous avons choisi de partager, il est possible de lire tous les fichiers et dossiers qu'il contient et de copier ce qui nous intéresse vers un autre dossier dans Windows.

Arrêter le partage

Pour une raison ou pour une autre, on peut vouloir arrêter de partager le dossier avec Windows.

Ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper **affi** et cliquer sur *Afficheur distant*. La première fois, il faut entrer l'adresse de la machine virtuelle dans le champ *Adresse de la connexion*. Généralement, l'adresse est : `spice://localhost:5900`. Si plus d'une machine virtuelle est en cours de fonctionnement la première démarrée aura l'adresse `spice://localhost:5900`, la deuxième `spice://localhost:5901` et ainsi de suite. À partir de la deuxième fois, il est possible de cliquer sur l'adresse désirée dans *Connexions récentes*. Cliquer sur le bouton *Connecter*.

Depuis la fenêtre de l'*Afficheur distant* contenant la machine virtuelle Windows, cliquer sur le menu *Fichier → Préférences*. Dans la fenêtre qui s'affiche, décocher la case *Partager le dossier*.

Le dossier sélectionné n'est maintenant plus accessible depuis Windows.

Garder un système à jour

Comme expliqué précédemment, les logiciels malveillants se faufilent dans nos ordinateurs, entre autres, par l'intermédiaire de « failles de sécurité ». [page 24]

Des corrections pour ces erreurs de programmation (ou de conception) sont régulièrement mises à disposition, au fur et à mesure qu'elles sont identifiées. Une fois que ces corrections sont disponibles, il est particulièrement important de remplacer les anciennes versions des logiciels. En effet, les problèmes corrigés, qui pouvaient n'avoir auparavant été identifiés que par quelques spécialistes, sont ensuite connus et référencés publiquement... donc plus faciles à exploiter.

23.1 Garder *Tails* à jour

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

Durée : 30 minutes à 1 heure, plus environ 30 minutes de téléchargement.

Un système live étant une collection indivisible de logiciels, exécutés à partir d'un DVD ou d'une clé USB, la seule solution praticable pour utiliser les dernières versions de ces logiciels est de s'assurer qu'on utilise bien la dernière version du système *live*. [page 101]

Au démarrage du système *live Tails*, *Mise à niveau proposée* ou *Nouvelle version disponible* apparaît pour nous prévenir lorsqu'une nouvelle version qui corrige des failles de sécurité est disponible.

Dans le cas où l'on utilise un DVD, il faut détruire celui contenant l'ancienne version et en graver un nouveau. Sauf si celui-ci est réinscriptible, auquel cas il suffira de l'effacer pour y graver la dernière version de *Tails*.

Pour une clé USB et dans la mesure où l'on dispose d'une connexion Internet, on peut utiliser le *Tails Upgrader* directement. Il suffit de cliquer sur *mettre à jour maintenant* et de suivre l'assistant tout au long du processus. Si une erreur se produit, ou s'il est nécessaire d'utiliser une autre méthode de mise à jour, l'assistant nous orientera vers la page de la documentation appropriée.

Celle-ci se trouve à partir de la *Documentation de Tails* se trouvant sur le bureau. Dans le menu à droite, cliquer sur *Documentation*. Dans l'index qui s'ouvre, chercher la section *Premier pas avec Tails* et cliquer sur la page *Mettre à jour une clé USB ou une carte SD Tails*.

23.2 Garder à jour un système chiffré

Une fois installé, un système chiffré doit être gardé à jour pour qu'on puisse continuer [page 107]

de lui faire confiance. Les sections qui suivent concernent le système *Debian*, mais les concepts s'appliquent dans les grandes lignes à quasiment tous les autres systèmes.

Le projet *Debian* publie, à peu près tous les deux ans, une version *stable*. Cela représente un énorme effort pour coordonner la compatibilité des différentes versions des logiciels, effectuer de nombreux tests et s'assurer qu'il n'y reste aucun défaut majeur.

23.3 Les mises à jour quotidiennes d'un système chiffré

C *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

⌚ *Durée : une minute pour lancer la mise à jour, plus un temps variable pour les téléchargements et l'installation, pendant lequel on peut continuer à utiliser son ordinateur.*

Tout l'intérêt d'une version *stable* de *Debian* est que par la suite, les logiciels qui la composent ne sont plus modifiés en profondeur : ne seront ajoutées que les améliorations de traduction, les corrections de problèmes liés à la sécurité ou empêchant d'utiliser normalement un programme.

Ces nouvelles versions peuvent donc être en général installées « les yeux fermés », elles ne devraient pas perturber les petites habitudes qu'on a prises.

Lorsqu'on a installé l'*environnement graphique de bureau*, le système vérifiera automatiquement, lorsqu'il sera connecté à Internet, la disponibilité de nouvelles versions dans les dépôts configurés.

page 124
Lorsque c'est le cas, une notification indiquant que *Des mises à jours logicielles sont disponibles* apparaîtra.

Cliquer sur la notification, ce qui ouvre *Logiciels*.

Une liste des mises à jour s'affiche. Cliquer sur *Redémarrer et installer* pour redémarrer l'ordinateur en installant les mises à jour. Le mot de passe d'administration nous est demandé, le saisir. Confirmer ensuite en cliquant à nouveau sur *Redémarrer et installer*. L'ordinateur redémarre et demande le mot de passe de chiffrement du disque dur, avant d'installer les mises à jour, puis de redémarrer à nouveau sur un système à jour.

23.4 Passage à une nouvelle version stable

C *Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.*

⌚ *Durée : une demi-journée à une journée, dont un long temps de téléchargement pendant lequel on peut continuer à utiliser son ordinateur, et un long temps d'installation pendant lequel il vaut mieux ne plus l'utiliser.*

Lorsqu'une nouvelle version *stable* de *Debian* sort, le projet veille à garder à jour la précédente version *stable*, appelée *oldstable* pendant une durée d'**un an**¹.

Il est donc nécessaire de profiter de cette période pour prendre le temps de mettre à jour son système vers cette nouvelle version. C'est un processus plus délicat que les mises à jour quotidiennes – pas nécessairement dans sa réalisation même, mais dans le fait qu'il est ensuite nécessaire de s'adapter aux changements qu'auront connu les logiciels que nous utilisons habituellement.

1. Debian, 2017, *DebianOldStable* [<https://wiki.debian.org/fr/DebianOldStable>].

23.4.1 Passage de Jessie à Stretch

La procédure détaillée ici concerne la mise à jour de la version de Debian baptisée *Jessie* ou 8, sortie en avril 2015, à la version *Stretch* ou 9, sortie en juin 2017.

Nous documenterons ici une procédure de mise à jour simplifiée qui a été testée sur des installations de Debian *Jessie* avec un environnement graphique de bureau GNOME et des logiciels provenant uniquement des dépôts officiels de Debian.

Elle nécessite de disposer, pour la durée de la mise à jour, d'une connexion à Internet.

Attention! Cette procédure simplifiée a moins de chances de fonctionner lorsqu'on a bidouillé son système en ajoutant des sources de mises à jour non officielles.

Si c'est le cas, se référer aux notes de publication officielles du projet Debian², notamment la partie Mises à niveau depuis Debian 8 (*Jessie*)³ et la partie Problèmes à connaître pour *Stretch*⁴.

Mettre à jour sa Debian *Jessie*

Avant tout, il est nécessaire de disposer d'une Debian *Jessie* à jour. Sans cela, la mise à niveau risque fort de ne pas fonctionner. Au cas où ces mises à jour n'auraient pas été faites au quotidien, c'est le moment de rattraper le retard. S'il vous est proposé de redémarrer, suite à de nombreuses mises à jour, le faire avant de procéder à la suite des opérations.

[page 162]

S'assurer d'avoir assez d'espace libre sur le disque dur

Avant d'éviter toute mauvaise surprise, il faut avoir au moins 4 Go d'espace libre sur le disque dur qui contient le système.

Ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper **fich** et cliquer sur *Fichiers*. Dans la barre de gauche, cliquer sur *Ordinateur*. Dans le menu , choisir *Propriétés*. Dans la fenêtre qui s'ouvre, l'information qui nous intéresse se trouve avant *libre*.

Libérer de l'espace sur le disque si nécessaire S'il n'y a pas assez d'espace sur le disque dur, une solution est d'effacer d'anciennes mises à jour devenues obsolètes. Pour cela, ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper **paquet** et cliquer sur *Gestionnaire de paquets*. Puisque le gestionnaire de paquets permet de modifier les logiciels installés sur l'ordinateur, un mot de passe est nécessaire pour l'ouvrir.

Dans le menu *Configuration* choisir *Préférences*, puis sélectionner l'onglet *Fichiers* et cliquer sur le bouton *Supprimer les paquets en cache*, puis sur *Valider* et fermer le *Gestionnaire de paquets Synaptic*.

Vérifier à nouveau l'espace disque disponible, comme expliqué ci-dessus. Si cela ne suffit pas, il faudra supprimer certains de nos propres fichiers ou supprimer des logiciels.

Désactiver l'économiseur d'écran

Lors de la mise à jour, l'économiseur d'écran peut se bloquer, et laisser l'écran verrouillé. Il est donc prudent de le désactiver pour le temps de la mise à jour.

Pour cela, ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper **param** et cliquer sur *Paramètres*.

2. <https://www.debian.org/releases/stretch/amd64/release-notes/index.fr.html>

3. <https://www.debian.org/releases/stretch/amd64/release-notes/ch-upgrading.fr.html>

4. <https://www.debian.org/releases/stretch/amd64/release-notes/ch-information.fr.html>

Cliquer ensuite sur *Confidentialité*. Cliquer sur *Verrouillage de l'écran*. Dans la fenêtre qui s'affiche, désactiver *Verrouillage automatique de l'écran*. Fermer cette fenêtre en cliquant sur \times , puis à nouveau sur \times en haut à droite, pour fermer la fenêtre *Confidentialité*.

Installer un logiciel nécessaire pour la procédure de mise à jour

Pour que notre système chiffré nous pose les questions liées à la mise à jour dans une fenêtre graphique, il faut installer le paquet `python-glade2`.

Mettre à jour les dépôts Debian utilisés

La mise à jour n'est testée qu'avec les paquets officiellement fournis par Debian *Jessie*. On va donc :

- d'une part, utiliser les dépôts officiels de la nouvelle version stable ;
- d'autre part, désactiver tous les autres *dépôts* Debian, y compris les dépôts *backports*.

Pour cela, ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper *paquet* et cliquer sur *Gestionnaire de paquets*.

Vu qu'on va choisir à quels programmes on fait confiance, on doit entrer le mot de passe d'administration.

Dans le menu *Configuration* choisir *Dépôts*. Cliquer ensuite sur les lignes une par une et :

- décocher l'entrée de tous les dépôts avec une URL non-officielle (autre que *debian.org*) ainsi que le dépôt *backports* ;
- si le champ *Distribution* est *jessie*, le remplacer par *stretch* ;
- si le champ *Distribution* est *jessie/upgrades* ou *jessie-updates*, le remplacer par *stretch/upgrades* ou *stretch-updates* respectivement.

Fermer cette fenêtre avec *Valider*, et recharger les dépôts en cliquant sur *Recharger* si le *Gestionnaire de paquets Synaptic* ne le demande pas de lui-même.

Lancer la mise à jour proprement dite

Cliquer sur le bouton *Tout mettre à niveau*. Une fenêtre intitulée *Prévoir d'effectuer d'autres changements ?* s'affiche alors. Choisir *Ajouter à la sélection*.

Lancer la mise à jour proprement dite en cliquant sur *Appliquer*. Une fenêtre intitulée *Appliquer les modifications suivantes ?* s'affiche. Dans *Résumé*, on peut vérifier qu'un grand nombre de paquets vont être mis à niveau. Cliquer sur *Appliquer*.

Le système télécharge alors les mises à jour depuis Internet, ce qui dans ce cas peut prendre entre quelques dizaines de minutes et plusieurs heures suivant la qualité de notre connexion.

Une fenêtre *Changelogs* peut alors s'afficher. Elle affiche en anglais une liste de changements importants qui vont être appliqués. Cliquer sur *Fermer*.

La fenêtre *Installation et suppression de logiciels* affiche la progression de la mise à jour.

Une fenêtre intitulée *Configuring libc6* (configuration de libc6) pourra alors s'afficher. Cocher *Restart services during package upgrades without asking ?* (redémarrer les services durant les mises à jour de paquets sans demander) puis cliquer sur *Forward* (suivant).

Une fenêtre *synaptic* peut aussi s'afficher. Elle indique qu'un fichier de configuration a été modifié et nous demande si on veut le remplacer par sa nouvelle version. Le *Garder* ou le *Remplacer* est un choix qui dépend de l'importance des modifications que l'on a pu y apporter ainsi que des nouveautés proposées. Il n'y a donc pas de réponse générique ici. Il faudra soit comparer les versions, soit jouer à pile ou face.

Une fois la mise à jour terminée, une fenêtre intitulée *Les modifications ont été appliquées* s'affiche. Cliquer sur *Fermer*, puis fermer le *gestionnaire de paquets*.

Premier redémarrage

Le moment est maintenant venu de redémarrer le système, en utilisant le menu  en haut à gauche et en choisissant *Redémarrer*.

Réactiver les dépôts Debian supplémentaires

On peut maintenant souffler. Le plus gros est fait. Il reste toutefois encore quelques petits ajustements...

Si l'on a désactivé des dépôts non officiels avant la mise à jour, c'est le moment de vérifier qu'on en a toujours besoin avec la nouvelle version de Debian. Si oui, les réactiver. On peut également réactiver l'économiseur d'écran si on l'a désactivé auparavant.

[page 124]

Réactiver l'économiseur d'écran

Ouvrir la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper *param* et cliquer sur *Paramètres*.

Cliquer sur *Confidentialité*. Cliquer sur *Verrouillage de l'écran*. Dans la fenêtre qui s'affiche, activer *Verrouillage automatique de l'écran*. Fermer cette fenêtre en cliquant sur *x*, puis à nouveau sur *x* en haut à droite, pour fermer la fenêtre *Confidentialité*.

S'assurer que le nouveau système fonctionne correctement

Il peut être utile de s'assurer que les actions et les commandes les plus courantes sont fonctionnelles. Le cas échéant, il pourrait être nécessaire de diagnostiquer et de résoudre les problèmes. Il vaut certainement mieux le faire dès la prise de contact avec le nouveau système, afin de pouvoir repartir pour deux ans avec un système fonctionnel. Les problèmes les plus courants sont souvent décrits, avec les astuces pour les résoudre, dans diverses documentations sur Debian et GNU/Linux.

[page 114]

Rappelons également qu'il existe des notes de publication officielles du projet Debian⁵.

5. <https://www.debian.org/releases/stretch/amd64/release-notes/index.fr.html>

Nettoyer les métadonnées d'un document

C Les logiciels évoluent, c'est pourquoi il est vivement conseillé d'utiliser la version la plus à jour de cet outil, qui est disponible sur le site web <https://guide.boum.org/>.

⌚ Durée : quelques minutes.

L'objectif de cet outil est d'effacer les métadonnées présentes dans un document avant sa publication. Ces métadonnées ne sont pas les mêmes dans tous les formats de documents : certaines sont plus difficiles à nettoyer que d'autres, voire impossibles. Cependant, la plupart des formats utilisés pour échanger des documents terminés, que ce soient des textes, des images, du son ou de la vidéo, sont « nettoyables ».

[page 22]

L'outil qu'on va utiliser ici est le *Metadata Anonymisation Toolkit* (*MAT*) qui permet de nettoyer aisément de nombreux formats de fichiers.

⚠️ Attention ! Nettoyer les métadonnées n'anonymise pas le contenu des fichiers, et n'enlève pas les éventuels marquages¹ qui seraient inclus dans le contenu lui-même.

24.1 Installer les logiciels nécessaires

Sous Tails, le *Metadata Anonymisation Toolkit* est déjà installé. Sur un système où le paquet `mat` n'est pas encore présent, il faut l'installer.

[page 122]

24.2 Ouvrir le *Metadata Anonymisation Toolkit*

Ouvrir *MAT* à partir de la vue d'ensemble des Activités : taper sur la touche  (⌘ sur un Mac), puis taper `mat` et cliquer sur *MAT*.

24.3 Ajouter des fichiers à nettoyer

Ajouter le fichier à nettoyer en cliquant sur le bouton *Ajouter*. Sélectionner le fichier et cliquer sur *Valider*. Il est également possible de le glisser directement dans *Metadata Anonymisation Toolkit*. On peut aussi ajouter plusieurs fichiers et les nettoyer en même temps. Si le logiciel sait nettoyer le fichier sélectionné, il l'ajoute à la liste de fichiers à nettoyer. Passer alors directement au paragraphe ci-dessous.

1. Voir à ce sujet Wikipédia, 2014, *Tatouage numérique* [https://fr.wikipedia.org/wiki/Tatouage_num%C3%A9rique] et Wikipédia, 2014, *Stéganographie* [<https://fr.wikipedia.org/wiki/St%C3%A9ganographie>].

S'il affiche un message « Non supporté », il faut convertir le document dans un format de fichiers supporté par le *MAT* avec une autre application avant de pouvoir le nettoyer. La liste des formats supportés est disponible à partir de *Aide → Information*.

Souvent, il suffit de l'exporter dans un format d'échange de fichiers commun. Ainsi, le *MAT* ne peut pas nettoyer le fichier *XCF* du programme de manipulation d'images GIMP, mais il sait nettoyer les images exportées en *JPEG* ou *PNG*.

24.4 Nettoyer les fichiers

Une fois le fichier ajouté, le nettoyer en cliquant sur le bouton *Nettoyer*. On peut alors fermer *Metadata Anonymisation Toolkit*.

Qui parle ?

Malheureusement, nous n'avons pas de réponse simple à offrir à cette question, mais souhaitons en dire quelques mots.

Tout d'abord, nous tenons à la possibilité de publier un livre de manière anonyme, et ce pour plusieurs raisons. L'une d'elles, que nous avons développée dans la préface, est qu'à la question « Rien à cacher ? », nous répondons à l'unissons « si ! ». L'anonymat est donc d'abord une manière de se protéger. De plus, nous choisissons de ne pas nous mettre en avant individuellement, afin d'écartier le *qui* du devant de la scène et de laisser le *quoi* sous les projecteurs.

Ensuite, depuis les premières parutions de ce *guide*, le nombre de personnes ayant participé, de près ou de loin, à sa rédaction, sa correction, son édition, rend à la fois large, évolutif et non clairement défini le collectif qui fait vivre ce projet.

Enfin, nous estimons avoir laissé suffisamment de traces au fil de ces pages pour permettre à toute personne nous lisant de nous situer, au moins partiellement, concernant notre rapport à l'informatique, qu'il soit technique, politique ou éthique.

*
* *

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 savoir-faire techniques, réservés d'ordinaire à de rares spécialistes. D'autre part, la justesse des indications fournies peut avoir des implications sur la sérénité des personnes qui les mettraient en œuvre. Les petites erreurs qui nous auront échappées 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 et savoir-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 soulevées par ce guide nous importent, que ce soit techniquement ou politiquement, depuis une dizaine d'années ;
- nous connaissons plutôt bien le fonctionnement de certains systèmes d'exploitation, et plus particulièrement celui de Debian GNU/Linux ;
- nous avons des bonnes bases en cryptographie, mais sommes très loin de pouvoir prétendre maîtriser le sujet.

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 proportionnelles aux conséquences en jeu.

Index

A

algorithme, 40
application, 16
architecture, 11
archivage, 79
argument, 88
authenticité, 44

B

backdoor, voir porte dérobée
bibliothèque, 16
binaire, 11
BIOS, 14, 64
boot, voir démarrage
bug, 21

C

cache, 35
carte-mère, 10
CD ou DVD, 158
chemin d'un fichier, 88
cheval de Troie, 24
chiffrement, 39
 chiffrer un système, 107
 chiffrer une clé, 133
clé de chiffrement, 40, 41
Code de la sécurité intérieure, 24
Code de procédure pénale, 23, 44
Code pénal, 43
code source, 31
cold boot attack, 19, 42, 63
collision, 44
confidentialité, 39
CPU, voir processeur
cryptanalyse, 39
cryptographie, 39
 cryptographie asymétrique, 46
 cryptographie symétrique, 46
cryptologie, 42

D

Debian, 16, 107
Déjà Dup, 141

démarrage, 95
disque dur, 12, 34
distribution, 32
dépôt de paquets, 124

E

écrasement des données, 34
effacement, 34
électricité, 15
empreinte, voir somme de contrôle
en-tête LUKS, 41
enregistreur de frappe, 27
espace d'échange, voir mémoire virtuelle

F

FAT32, 17
fonction de hachage, 44
force brute, 65
format de fichiers, 18
formatage, 17, 36

G

Gestionnaire de machine virtuelle, 72, 152
GNU/Linux, 16, 32
GnuPG, 41

H

hachage, 44
HADOPI, 26
hibernation, 20
historique, 21

I

imprimante, 28
installation d'un logiciel, 117
installation d'un système, 107
installeur, 107
intégrité, 44

J

journal, 21

journalisation, 35

K

keylogger, voir enregistreur de frappe

L

library, voir bibliothèque

licence libre, 32

licence propriétaire, 31

ligne de commande, 87

liste autorisée, 55

liste bloquée, 55

log, voir journaux

logiciel, 15

logiciel espion, 24

logiciel libre, 31, 32

logiciel malveillant, 23, 24, 64

logiciel open source, 32

logiciel portable, 36

logiciel propriétaire, 31

lois, 23

loi relative au renseignement, 24
loi renforçant la lutte contre le
crime organisé, le terrorisme
[...], 23

loi renforçant les dispositions rela-
tives à la lutte contre le ter-
rorisme, 44

LUKS, 42, 133

M

malware, voir logiciel malveillant

mémoire *flash*, 14, 34, 128

mémoire morte, voir mémoire persis-
tante

mémoire persistante, 12

mémoire virtuelle, 18, 20

mémoire vive, 12, 19

micrologiciel, 14, 64

mise à jour, 161

mot de passe, 33

méta-données, 22

N

noyau, 16

NTFS, 17

numérisation, 10

O

ondes, 15

open source, 32

option, 88

OS, voir système d'exploitation

P

paquet Debian, 117

partition, 17

phrase de passe, 39, 93

pilote, 16

politique de sécurité, 53

porte dérobée, 31

pourriel, 23, 24

processeur, 10

programme, 15

périphérique, 14

R

RAM, voir mémoire vive

rootkit, 24

S

sauvegarde, 139

sauvegardes automatiques, 21

secure-delete, 127

sfill, 132

shred, 131

signature numérique, 46

signature stéganographique, 28

somme de contrôle, 44, 149

spam, voir pourriel

spyware, voir logiciel espion

SSD, voir mémoire *flash*

stéganographie, 28

swap, 18, 20

Synaptic, 118, 122

yntaxe, 88

système de fichiers, 17, 35

système d'exploitation, 16

système hôte, 72

système invité, 72

système *live*, 36, 70, 101

T

terminal, 87

traces, 19

transistor, 10

TrueCrypt, 32

U

UEFI, 14, 64

upgrade, voir mise à jour

USB, 14

V

veille, 20

VirtualBox, 151

virtualisation, 71, 151

virus, 24

W

watermarking, voir signature stégano-
graphique

Windows, 70, 153

wipe, voir érassement des données

Crédits

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 12, 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, domaine public, trouvée sur :

https://commons.wikimedia.org/wiki/File:MSATA_SSD_16_GB_Sandisk--SDSA3DD-016G-2494.jpg.

Photo page 14 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.

Les autres schémas sont faits par les auteurs du guide et utilisent des icônes : de GNOME Project, licence CC BY-SA 3.0 ; de Silvestre Herrera, licence GPLv2 trouvées sur <http://www.silvestre.com.ar/> ; du domaine public trouvées sur <http://openclipart.org>.

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.