

# Merci du signalement, ton bug peut attendre...

Magnus Manske est un développeur inconnu du grand public, on lui doit pourtant des contributions nombreuses et décisives pour le développement initial de Wikipédia, sa maintenance continue et son ingénierie, au point que les wikipédiens célèbrent chaque 25 janvier le *Magnus Manske Day*.

On imagine aisément à quel point ses journées sont bien remplies, d'autant qu'il donne son temps et son énergie pour le Libre sur son temps... libre !

Dans l'article dont nous vous proposons la traduction, il explique avec un brin de malice pourquoi les bugs apparemment les plus simples à traiter peuvent s'avérer les plus longs à régler... Deux minutes de lecture pour un petit article qui parlera à nos amis développeurs (ces indispensables travailleurs de l'ombre) comme il a parlé à Luc, notre tech warrior tout-terrain...

## Non, je n'ai pas corrigé ton bogue, voici pourquoi

par [Magnus Manske](#) – article original : [Why I didn't fix your bug](#)



Photo par Jason Krüger [[CC BY-SA 4.0](#)], via *Wikimedia Commons*

Beaucoup d'entre vous m'ont envoyé des rapports de bogue, des demandes de nouvelles fonctionnalités et signalé d'autres problèmes liés à mes outils dans le WikiVerse. Vous m'avez contacté via le *BitBucket Issue tracker* (et apparemment je suis aussi sur Phabricator maintenant), par Twitter, divers emails, des pages de discussion (les miennes, celles d'autres utilisateurs, via Wikitech, etc.), avec des applications de messagerie, et même en chair et en os.

**Et je n'ai rien fait. Je n'ai même pas répondu.**

Rien n'indique que j'ai vu le problème.

C'est frustrant, je sais. Tu veux juste que ce petit truc soit réparé. En tout cas, tu te figures que c'est un tout petit changement à opérer.

Voyons maintenant les ressources disponibles, ce qui, en l'occurrence, est mon temps. En commençant par les gros travaux (estimations générales, des variations saisonnières sont inévitables) :

Il y a 24 h dans une journée

- 9 h de travail (y compris le trajet en voiture)
- 7 h de sommeil (j'espère en tout cas)
- 2 h de vie privée (manger, faire de l'exercice, prendre une

douche, lire, passer du temps avec ma copine, etc.)

= il reste 6h

On ne peut pas discuter avec ça, n'est-ce pas ? Maintenant, 6h qui restent, c'est une estimation haute, évidemment ; le travail et la vie privée peuvent (et ça arrive) prendre bien plus de temps, sur une base quotidienne et très variable, comme c'est le cas pour chacun de nous.

## **Alors je peux régler ton problème, c'est ça ?**

Voyons voir :

6h

– 1h de maintenance (redémarrage des outils, mise à jour des [pages GLAM](#), ajout et correction des catalogues [mix“n”match](#), etc.)

– 3h de développement/réécriture (parce que c'est de là que viennent les outils)

= il reste 2h

Deux heures par jour, ça fait beaucoup, non ? En réalité, c'est beaucoup moins, mais restons-en là pour l'instant. Quelques-uns de mes outils n'ont pas de problèmes, mais beaucoup en ont plusieurs en cours, donc supposons que chaque outil en a un :

2h = 120 min

/130 outils (estimation basse)<sup>1</sup>

= une moyenne de 55 secondes par outil

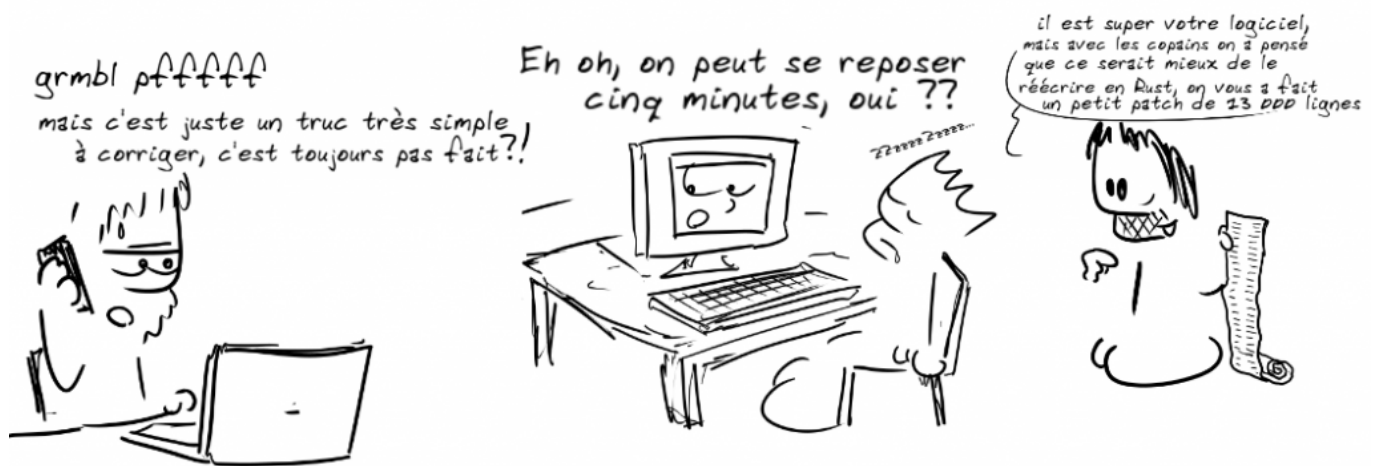
C'est assez de temps pour trouver et aborder le problème, ouvrir le(s) fichier(s) de code source, et... zut le temps s'est écoulé ! Désolé, problème suivant !

Donc, au lieu de tous les traiter, je m'occupe de l'un d'entre eux. Jusqu'à ce que ce soit réglé ou que j'abandonne. L'un ou l'autre peut prendre des minutes, des heures, voire des jours. Et pendant ce temps, je ne me penche pas sur les centaines d'autres problèmes. Parce que je ne peux rien faire pour eux à ce moment-là.

Alors, comment puis-je choisir un problème sur lequel travailler ? C'est une heuristique complexe calculée à partir des facteurs suivants :

- Nombre d'utilisateurs concernés
- Gravité (« problème de sécurité » vs « faute d'orthographe »)
- Opportunité (ce qui signifie que je l'ai remarqué lorsqu'il a été déposé)
- Disponibilité (est-ce que je me concentre sur autre chose lorsque je remarque le problème ?)
- Plaisir possible et humeur du moment (eh oui, car je suis bénévole, ça vous dérange ?).

Aucun événement particulier n'a été à l'origine de la publication de ce billet. Je le garderai en référence pour en donner le lien, quand l'occasion se présentera.



---

# Les algos peuvent vous pourrir la vie

Les algorithmes<sup>[1]</sup> ne sont guère qu'une série d'instructions pas-à-pas généralement exécutées par un programme sur une machine. Cependant leur complexité et leur opacité pour le commun des mortels sont redoutables, et bien plus encore leur omniprésence dans tous les compartiments de notre vie, y compris la plus intime. Si le code fait la loi, c'est justement parce que les algorithmes sont à la fois puissants, invasifs et sont devenus aujourd'hui indispensables.

L'article ci-dessous ne met pas l'accent sur les nombreux domaines où nous utilisons des algorithmes sans en avoir conscience, il pointe davantage les risques et menaces qu'ils représentent lorsque ce sont les algorithmes qui déterminent notre existence, à travers quelques exemples parmi bien d'autres. Il pose également l'intéressante question de la

*responsabilité de ceux qui élaborent les algorithmes. Suffira-t-il de réclamer des concepteurs d'algorithmes un sympathique engagement solennel à la manière de [celui des acteurs du Web](#) ?*

*Les codeurs dont les algos contrôlent nos vies, qui les contrôle ? Pouvons-nous avoir un droit de regard sur les algorithmes qui désormais menacent de régir nos vies ?*



## Les algorithmes sont formidables mais peuvent aussi ruiner des vies

Extrait de l'essai (en anglais) [The Formula: How Algorithms Solve All Our Problems—and Create More](#) par **Luke Dormehl**.

Source : article du magazine **Wired** [Algorithms are great and all, but can also ruin our lives](#)

Traduction Framalang : Wan, r0u, goofy, Sphinx, sinma, Omegax, ylluss, audionuma

Le 5 avril 2011, John Gass, 41 ans, a reçu un courrier du service d'enregistrement des véhicules motorisés (Registry of Motor Vehicles ou RMV) de l'État du Massachusetts. La lettre informait M. Gass que son permis de conduire avait été annulé, qu'il lui était désormais interdit de conduire et que cela prenait effet immédiatement. Le seul problème, c'est qu'en bon conducteur n'ayant pas commis d'infraction grave au code de la

route depuis des années, M. Gass n'avait aucune idée du motif de ce courrier.

Après plusieurs appels téléphoniques frénétiques, suivis par une entrevue avec les fonctionnaires du service, il en a appris la raison : son image avait été automatiquement signalée par un algorithme de reconnaissance faciale conçu pour parcourir une base de données de millions de permis de conduire de l'État, à la recherche de possibles fausses identités criminelles. L'algorithme avait déterminé que Gass ressemblait suffisamment à un autre conducteur du Massachusetts pour présumer d'une usurpation d'identité, d'où le courrier automatisé du RMV.

Les employés du RMV se sont montrés peu compréhensifs, affirmant qu'il revenait à l'individu accusé de prouver son identité en cas d'erreur quelconque et faisant valoir que les avantages de la protection du public l'emportaient largement sur les désagréments subis par les quelques victimes d'une accusation infondée.

John Gass est loin d'être la seule victime de ces erreurs d'algorithmes. En 2007, un bogue dans le nouveau système informatique du Département des services de santé de Californie a automatiquement mis fin aux allocations de milliers de personnes handicapées et de personnes âgées à bas revenus. Leurs frais d'assurance maladie n'étant plus payés, ces citoyens se sont alors retrouvés sans couverture médicale.

Là où le système précédent aurait notifié les personnes concernées qu'elles n'étaient plus considérées comme éligibles aux allocations en leur envoyant un courrier, le logiciel maintenant opérationnel, CalWIN, a été conçu pour les interrompre sans avertissement, à moins de se connecter soi-même et d'empêcher que cela n'arrive. Résultat : un grand nombre de ceux dont les frais n'étaient plus pris en charge ne s'en sont pas rendu compte avant de recevoir des factures médicales salées. Encore beaucoup n'avaient-ils pas les

compétences nécessaires en anglais pour naviguer dans le système de santé en ligne et trouver ce qui allait de travers.

Des failles similaires sont à l'origine de la radiation de votants des listes électorales sans notification, de petites entreprises considérées à tort comme inéligibles aux contrats gouvernementaux, et d'individus identifiés par erreur comme « parents mauvais payeurs ». Comme exemple notable de ce dernier cas, Walter Vollmer, mécanicien de 56 ans, a été ciblé à tort par le Service fédéral de localisation des parents, et s'est vu envoyer une facture de pension alimentaire à hauteur de 206 000 \$. L'épouse de M. Vollmer, 32 ans, a par la suite montré des tendances suicidaires, persuadée que son mari avait eu une vie cachée pendant la majeure partie de leur mariage.

Une possibilité tout aussi alarmante : qu'un algorithme puisse ficher par erreur un individu comme terroriste. Un sort qui attend chaque semaine environ 1500 voyageurs malchanceux qui prennent l'avion. Parmi les victimes passées de ces erreurs de corrélation de données, on retrouve d'anciens généraux de l'armée, un garçon de quatre ans, ainsi qu'un pilote d'*American Airlines*, qui a été détenu 80 fois au cours d'une même année.

Beaucoup de ces problèmes sont dus aux nouveaux rôles joués par les algorithmes dans l'application de la loi. Les budgets réduits menant à des réductions de personnel, les systèmes automatisés, auparavant de simples instruments administratifs, sont maintenant des décideurs à part entière.

Dans nombre de cas, le problème est plus vaste que la simple recherche d'un bon algorithme pour une tâche donnée. Il touche à la croyance problématique selon laquelle toutes les tâches possibles et imaginables peuvent être automatisées. Prenez par exemple l'extraction de données, utilisée pour découvrir les complots terroristes : de telles attaques sont statistiquement rares et ne se conforment pas à un profil bien défini comme, par exemple, les achats sur Amazon. Les voyageurs finissent



par abandonner une grande partie de leur vie privée au profit des algorithmes d'extraction de données, avec peu de résultats, si ce n'est des faux-positifs. Comme le note Bruce Schneier, le célèbre expert en sécurité informatique :

*Chercher des complots terroristes... c'est comme chercher une aiguille dans une botte de foin, ce n'est pas en accumulant davantage de foin sur le tas qu'on va rendre le problème plus facile à résoudre. Nous ferions bien mieux de laisser les personnes chargées d'enquêtes sur de possibles complots prendre la main sur les ordinateurs, plutôt que de laisser les ordinateurs faire le travail et les laisser décider sur qui l'on doit enquêter.*

Bien qu'il soit clair qu'un sujet aussi brûlant que le terrorisme est un candidat parfait pour ce type de solutions, le problème central se résume encore une fois à cette promesse fantomatique de l'*objectivité* des algorithmes. « Nous sommes tous absolument effrayés par la subjectivité et l'inconstance du comportement humain », explique Danielle Citron, professeur de droit à l'Université du Maryland. « Et à l'inverse, nous manifestons une confiance excessive pour tout ce que peuvent accomplir les ordinateurs ».

Le professeur Citron suggère que l'erreur vient de ce que nous « faisons confiance aux algorithmes, parce que nous les percevons comme objectifs, alors qu'en réalité ce sont des humains qui les conçoivent, et peuvent ainsi leur inculquer toutes sortes de préjugés et d'opinions ». Autrement dit, un algorithme informatique a beau être impartial dans son exécution, cela ne veut pas dire qu'il n'a pas de préjugés codés à l'intérieur.

Ces erreurs de jugement, implicites ou explicites, peuvent être causées par un ou deux programmeurs, mais aussi par des difficultés d'ordre technique. Par exemple, les algorithmes utilisés dans la reconnaissance faciale avaient par le passé

de meilleurs taux de réussite pour les hommes que pour les femmes, et meilleurs pour les personnes de couleur que pour les Blancs.

Ce n'est pas par préjugé délibéré qu'un algorithme ciblera plus d'hommes afro-américains que de femmes blanches, mais cela ne change rien au résultat. De tels biais peuvent aussi venir de combinaisons plus abstraites, enfouies dans le chaos des corrélations de jeux de données.

Prenez par exemple l'histoire de l'afro-américaine [Latanya Sweeney](#), docteure de l'Université d'Harvard. En effectuant des recherches sur Google, elle fut choquée de découvrir que les résultats de ses recherches étaient accompagnés de publicités demandant : « Avez-vous déjà été arrêté(e) ? ». Ces annonces n'apparaissaient pas pour ses collègues blancs. Sweeney se lança alors dans une étude, démontrant que les outils d'apprentissage automatique utilisés par Google étaient incidemment racistes, en associant plus souvent des noms donnés à des personnes noires avec des publicités ayant trait aux rapports d'arrestation.

Le système de recommandation de Google Play révèle un problème similaire : il suggère aux utilisateurs qui téléchargent [Grindr](#), un outil de réseautage social basé sur la localisation pour les gays, de télécharger également une application qui assure le suivi géolocalisé des délinquants sexuels. Au vu de ces deux cas, devons-nous conclure que les algorithmes ont fait une erreur, ou plutôt qu'ils sont révélateurs des préjugés inhérents à leurs concepteurs ? Ou, ce qui semble plus probable, ne seraient-ils pas révélateurs d'associations inappropriées et à grande échelle entre – dans le premier cas – les personnes noires et le comportement criminel, et – dans le deuxième cas – l'homosexualité et les agressions sexuelles ?

Peu importe la raison, peu importe la façon répréhensible dont ces corrélations codifiées peuvent exister, elles révèlent une

autre face de la culture algorithmique. Quand un seul individu fait explicitement une erreur de jugement, il ne peut jamais affecter qu'un nombre fini de personnes. Un algorithme, quant à lui, a le potentiel d'influer sur un nombre de vies exponentiellement plus grand.



Clochix  
@clochix



Abonné

C'est pour cela que nous devons exiger l'ouverture des algorithmes, pour savoir à quelle sauce nous sommes dévorés

Pour aller plus loin, 4 articles en français sur le même sujet :

- [Surveiller les algorithmes](#)
- [Ces algorithmes qui vous nous gouvernent](#)
- [Ouvrir les modèles, pas seulement les données](#)
- [Le jaguar et le bus scolaire](#)

## Note

[1] Pour une définition plus élaborée voir [Qu'est-ce qu'un algorithme](#)

---

# Plus rien ne marche, qu'est-ce qu'on fait ?

*Désormais conscients et informés que nos actions et nos données en ligne sont faciles à espionner et l'enjeu de monétisation en coulisses, il nous restait l'espoir que quelques pans des technologies de sécurité pouvaient encore faire échec à la surveillance de masse et au profilage*

*commercial. Pas facile pour les utilisateurs moyens d'adopter des outils et des pratiques de chiffrement, par exemple, cependant de toutes parts émergent des projets qui proposent de nous aider à y accéder sans peine.*

*Mais quand les experts en sécurité, quittant un moment leur regard hautain sur le commun des mortels à peine capables de choisir un mot de passe autre que 123AZERTY, avouent qu'ils savent depuis longtemps que tout est corrompu directement ou indirectement, jusqu'aux services soi-disant sécurisés et chiffrés, le constat est un peu accablant parce qu'il nous reste tout à reconstruire...*

## **Plus rien ne fonctionne**

*article original : [Everything is broken](#) par [Quinn Norton](#)*

*Traduction Framalang : Diab, rafiote, Omegax, Scailyna, Amine Brikci-N, EDGE, r0u, fwix, dwarfpower, sinma, Wan, Manu, Asta, goofy, Solarus, Lumi, mrtino, skhaen*

Un beau jour un de mes amis a pris par hasard le contrôle de plusieurs milliers d'ordinateurs. Il avait trouvé une faille dans un bout de code et s'était mis à jouer avec. Ce faisant, il a trouvé comment obtenir les droits d'administration sur un réseau. Il a écrit un script, et l'a fait tourner pour voir ce que ça donnerait. Il est allé se coucher et il a dormi environ quatre heures. Le matin suivant, en allant au boulot, il a jeté un coup d'œil et s'est aperçu qu'il contrôlait désormais près de 50 000 ordinateurs. Après en avoir pratiquement vomi de trouille, il a tout arrêté et supprimé tous les fichiers associés. Il m'a dit que finalement il avait jeté le disque dur au feu. Je ne peux pas vous révéler de qui il s'agit, parce qu'il ne veut pas finir dans une prison fédérale ; et c'est ce qui pourrait lui arriver s'il décrivait à qui que ce soit la faille qu'il a découverte. Cette faille a-t-elle été corrigée ? Sans doute... mais pas par lui. Cette histoire n'est en rien exceptionnelle. Passez quelque temps dans le monde des

hackers et de la sécurité informatique, et vous entendrez pas mal d'histoires dans ce genre et même pires que celle-là.

Il est difficile d'expliquer au grand public à quel point la technologie est chancelante, à quel point l'infrastructure de nos vies ne tient qu'avec l'équivalent informatique de bouts de ficelle. Les ordinateurs et l'informatique en général sont détraqués.

## Quand c'est codé avec les pieds, bonjour les vautours

Pour un bon nombre d'entre nous, en particulier ceux qui ont suivi l'actualité en matière de sécurité et les questions d'écoutes sauvages, rien de surprenant dans toutes les dernières révélations. Si nous ne connaissons pas les détails, nous savions tous, dans le monde de la sécurité, que la technologie est vacillante et malade. Depuis des années nous voyons tourner les vautours qui veulent profiter de cet état de fait. La NSA n'est pas et n'a jamais été le grand prédateur unique fondant sur Internet. C'est simplement le plus gros de ces charognards. S'ils arrivent à aller aussi loin, ce n'est pas parce que leurs employés sont des dieux des maths.

Si la NSA s'en sort si bien, c'est parce que les logiciels en général sont merdiques.

Huit mois avant que Snowden ne fasse ses révélations, j'ai twitté ça :



*« alerte de sécu : tout a une faille 0 day, tout le monde est suivi à la trace, toutes les données fuient, tout est vulnérable, tout est compromis jusqu'à l'os. »*

J'en étais arrivée à cette conclusion un peu désespérée : chercher des logiciels de qualité est un combat perdu d'avance. Comme ils sont écrits par des gens n'ayant ni le temps ni l'argent nécessaires, la plupart des logiciels sont publiés dès qu'ils fonctionnent assez bien pour laisser leurs auteurs rentrer chez eux et retrouver leur famille. Pour nous le résultat est épouvantable.

Si les logiciels sont aussi mauvais, c'est parce qu'ils sont très complexes, et qu'il cherchent à parler à d'autres logiciels, soit sur le même ordinateur, soit au travers du réseau. Même votre ordinateur ne peut plus être considéré comme unique : c'est une poupée russe, et chaque niveau est fait de quantité d'éléments qui essaient de se synchroniser et de parler les uns avec les autres. L'informatique est devenue incroyablement complexe, alors que dans le même temps les gens sont restés les mêmes, pétris de la même boue grise originelle pleine d'une prétention à l'étincelle divine.

Le merdier qu'est votre ordinateur sous Windows est tellement complexe que personne sur Terre ne sait tout ce qu'il fait vraiment, ni comment.

Maintenant imaginez des milliards de petites boîtes opaques qui essaient en permanence de discuter les unes avec les autres, de se synchroniser, de travailler ensemble, partageant des bouts de données, se passant des commandes... des tous petits bouts de programmes aux plus gros logiciels, comme les navigateurs – c'est ça, Internet. Et tout ça doit se passer quasi-simultanément et sans accrocs. Sinon vous montez sur vos grands chevaux parce que le panier de la boutique en ligne a oublié vos tickets de cinéma.

On n'arrête pas de vous rappeler que le téléphone avec lequel

vous jouez à des jeux stupides et que vous laissez tomber dans les toilettes au troquet du coin est plus puissant que les ordinateurs utilisés pour la conquête de l'espace il y a de cela quelques décennies à peine. La NASA dispose d'une armée de génies pour comprendre et maintenir ses logiciels. Votre téléphone n'a que vous. Ajoutez à cela un mécanisme de mises à jour automatiques que vous désactivez pour qu'il ne vous interrompe pas au beau milieu d'une séance de Candy Crush...

À cause de tout ça, la sécurité est dans un état effrayant. En plus d'être truffés de bugs ennuyeux et de boîtes de dialogue improbables, les programmes ont souvent un type de faille piratable appelée *0 day* (« zéro jour ») dans le monde de la sécurité informatique. Personne ne peut se protéger des *0 days*. C'est justement ce qui les caractérise : *0* représente le nombre de jours dont vous disposez pour réagir à ce type d'attaque. Il y a des *0 days* qui sont anodins et vraiment pas gênants, il y a des *0 days* très dangereux, et il y a des *0 days* catastrophiques, qui tendent les clés de la maison à toute personne qui se promène dans le coin. Je vous assure qu'en ce moment même, vous lisez ceci sur une machine qui a les trois types de *0 days*. Je vous entends d'ici me dire : « Mais, Quinn, si personne ne les connaît comment peux-tu savoir que je les ai ? » C'est parce que même un logiciel potable doit avoir affaire avec du code affreux. Le nombre de gens dont le travail est de rendre le logiciel sûr peut pratiquement tenir dans un grand bar, et je les ai regardé boire. Ce n'est pas rassurant. La question n'est pas : « est-ce que vous allez être attaqué ? » mais : « quand serez-vous attaqué ? »

Considérez les choses ainsi : à chaque fois que vous recevez une mise à jour de sécurité (apparemment tous les jours avec mon ordi sous Linux), tout ce qui est mis à jour a été cassé, rendu vulnérable depuis on ne sait combien de temps. Parfois des jours, parfois des années. Personne n'annonce vraiment cet aspect des mises à jour. On vous dit « Vous devriez installer

cela, c'est un patch critique ! » et on passe sous silence le côté « ...parce que les développeurs ont tellement merdé que l'identité de vos enfants est probablement vendue en ce moment même à la mafia estonienne par des script kiddies accros à l'héro ».

Les bogues vraiment dangereux (et qui peut savoir si on a affaire à eux lorsqu'on clique sur le bouton « Redémarrer ultérieurement » ?) peuvent être utilisés par des hackers, gouvernements, et d'autres horreurs du net qui fouillent à la recherche de versions de logiciels qu'ils savent exploiter. N'importe quel ordinateur qui apparaît lors de la recherche en disant « Hé ! Moi ! Je suis vulnérable ! » peut faire partie d'un botnet, en même temps que des milliers, ou des centaines de milliers d'autres ordinateurs. Souvent les ordinateurs zombies sont possédés à nouveau pour faire partie d'un autre botnet encore. Certains botnets patchent les ordinateurs afin qu'ils se débarrassent des autres botnets, pour qu'ils n'aient pas à vous partager avec d'autres hackers. Comment s'en rendre compte si ça arrive ? Vous ne pouvez pas ! Amusez-vous à vous demander si votre vie en ligne va être vendue dans l'heure qui suit ! La prochaine fois que vous penserez que votre grand-mère n'est pas cool, pensez au temps qu'elle a passé à aider de dangereux criminels russes à extorquer de l'argent à des casinos offshore avec des attaques DDoS.

Récemment un hacker anonyme a écrit un script qui prenait le contrôle d'appareils embarqués Linux. Ces ordinateurs possédés scannaient tout le reste d'Internet et ont créé un rapport qui nous en a appris beaucoup plus que ce que nous savions sur l'architecture d'Internet. Ces petites boîtes hackées ont rapporté toutes leurs données (un disque entier de 10 To) et ont silencieusement désactivé le hack. C'était un exemple délicieux et utile d'un individu qui a hacké la planète entière. Si ce malware avait été véritablement malveillant, nous aurions été dans la merde.

Et ceci parce que les ordinateurs sont tous aussi



inévitablement défectueux : ceux des hôpitaux et des gouvernements et des banques, ceux de votre téléphone, ceux qui contrôlent les feux de signalisation et les capteurs et les systèmes de contrôle du trafic aérien. Chez les industriels, les ordinateurs destinés à maintenir l'infrastructure et la chaîne de fabrication sont encore pires. Je ne connais pas tous les détails, mais ceux qui sont les plus au courant sont les personnes les plus alcooliques et nihilistes de toute la sécurité informatique. Un autre de mes amis a accidentellement éteint une usine avec un "ping" malformé au début d'un test d'intrusion. Pour ceux qui ne savent pas, un "ping" est seulement la plus petite requête que vous pouvez envoyer à un autre ordinateur sur le réseau. Il leur a fallu une journée entière tout faire revenir à la normale.

Les experts en informatique aiment prétendre qu'ils utilisent des logiciels d'un genre complètement différent, encore plus géniaux, qu'eux seuls comprennent, des logiciels faits de perfection mathématique et dont les interfaces semblent sortir du cul d'un âne colérique. C'est un mensonge. La forme principale de sécurité qu'ils offrent est celle que donne l'obscurité – il y a si peu de gens qui peuvent utiliser ces logiciels que personne n'a le moindre intérêt à concevoir des outils pour les attaquer. Sauf si, comme la NSA, vous voulez prendre le contrôle sur les administrateurs systèmes.

## **Une messagerie chiffrée et bien codée, il ne peut rien nous arriver, hein ?**

Prenons un exemple que les experts aiment mettre sous le nez des gens normaux qui ne l'utilisent pas : OTR. OTR, ou *Off The Record messaging*, ajoute une couche de chiffrement aux échanges via messagerie instantanée. C'est comme si vous utilisiez AIM ou Jabber et que vous parliez en code sauf que c'est votre ordinateur qui fait le code pour vous. OTR est bien conçu et robuste, il a été audité avec attention et nous

sommes bien sûrs qu'il ne contient aucune de ces saloperies de vulnérabilités zéro jour.

Sauf que OTR n'est pas vraiment un programme que vous utilisez tel quel.

Il existe un standard pour le logiciel OTR, et une bibliothèque, mais elle ne fait rien par elle-même. OTR est implémentée dans des logiciels pour des neuneus par d'autres neuneus. À ce stade, vous savez que ça va se terminer dans les pleurs et les grincements de dents.

La partie principale qu'utilise OTR est un autre programme qui utilise une bibliothèque appelée "libpurple". Si vous voulez voir des snobs de la sécurité aussi consternés que les ânes qui ont pondu leur interface, apportez-leur "libpurple". "Libpurple" a été écrit dans un langage de programmation appelé C.

Le C est efficace dans deux domaines : l'élégance, et la création de vulnérabilités jour zéro critiques en rapport avec la gestion de la mémoire.

*Heartbleed*, le bogue qui a affecté le monde entier, permettant la fuite de mots de passe et de clés de chiffrement et qui sait quoi encore ? – Du classique et superbe C.

La "libpurple" a été écrite par des gens qui voulaient que leur client de discussion *open source* parle à tous les systèmes de messagerie instantanée du monde, et se foutaient complètement de la sécurité ou du chiffrement. Des gens du milieu de la sécurité qui en ont examiné le code ont conclu qu'il y avait tellement de façons d'exploiter la "libpurple" que ça n'était probablement pas la peine de la patcher. Elle doit être jetée et réécrite de zéro. Ce ne sont pas des bugs qui permettent à quelqu'un de lire vos messages chiffrés, ce sont des bugs qui permettent à n'importe qui de prendre le contrôle total de votre ordinateur, regarder tout ce que vous tapez ou lisez et même probablement vous regarder vous mettre

les doigts dans le nez devant la webcam.

Ce magnifique outil qu'est OTR repose sur la "libpurple" dans la plupart des systèmes où il est utilisé. Je dois éclaircir un point, car même certains geeks n'en ont pas conscience : peu importe la force de votre chiffrement si celui qui vous attaque peut lire vos données par-dessus votre épaule, et je vous promets que c'est possible. Qu'il sache le faire ou pas encore, cela reste néanmoins possible. Il y a des centaines de bibliothèques comme "libpurple" sur votre ordinateur : des petits bouts de logiciels conçus avec des budgets serrés aux délais irréalistes, par des personnes ne sachant pas ou ne se souciant pas de préserver la sécurité de votre système.

Chacun de ces petits bugs fera l'affaire quand il s'agit de prendre le contrôle de tout le reste de votre ordinateur. Alors on met à jour, on remet à jour, et peut-être que ça mettra les intrus dehors, ou peut-être pas. On n'en sait rien ! Quand on vous dit d'appliquer les mises à jour, on ne vous dit pas de réparer votre navire. On vous dit de continuer à écopier avant que l'eau n'atteigne votre cou.



(Crédit image :

[sridgway](#), licence CC BY 2.0)

Pour prendre un peu de recul par rapport à cette scène d'horreur et de désolation, je dois vous dire que la situation est tout de même meilleure que par le passé. Nous disposons aujourd'hui d'outils qui n'existaient pas dans les années 90, comme le "sandboxing", qui permet de confiner des programmes écrits stupidement là où ils ne peuvent pas faire beaucoup de dégâts. (Le « sandboxing » consiste à isoler un programme dans une petite partie virtuelle de l'ordinateur, le coupant ainsi de tous les autres petits programmes, ou nettoyant tout ce que ce programme essaie de faire avant que d'autres puissent y accéder).

Des catégories entières de bugs horribles ont été éradiqués comme la variole. La sécurité est prise plus au sérieux que jamais, et il y a tout un réseau de personnes pour contrer les logiciels malveillants 24h sur 24. Mais ils ne peuvent pas vraiment garder la main. L'écosystème de ces problèmes est tellement plus vaste qu'il ne l'était ne serait-ce qu'il y a dix ans, qu'on ne peut pas vraiment dire que l'on fait des progrès.

## **Les gens, eux aussi, sont cassés**

« Je vous fais confiance... » est ce que j'aime le moins entendre de la part des mes sources Anonymous. C'est invariablement suivi de bribes d'informations qu'ils n'auraient jamais dû me confier. Il est naturel de partager quelque chose de personnel avec quelqu'un en qui on a confiance. Mais c'est avec exaspération que je dois rappeler aux Anons qu'avant d'être connectés à un autre être humain ils sont d'abord connectés à un ordinateur, relayé à travers un nombre indéterminé de serveurs, switches, routeurs, câbles, liaisons sans fil, et en bout de chaîne, mon ordinateur parfaitement ciblé par les attaques. Tout ceci se déroule le temps d'une longue inspiration. Cela semble une évidence, mais il est bon de le rappeler : les humains ne sont pas conçus pour penser de cette manière.

Personne n'arrive à utiliser les logiciels correctement. Absolument tout le monde se plante. OTR ne chiffre pas avant le premier message, un fait que des éminents professionnels de la sécurité et des hackers qui subissent une chasse à l'homme dans une vingtaine de pays oublient en permanence. Gérer toutes les clés de chiffrement et de déchiffrement dont vous avez besoin pour garder vos données en sûreté sur plusieurs appareils, sites, et comptes est théoriquement possible, de la même façon que réaliser une appendicectomie sur soi-même est théoriquement possible. *Il y a un gars qui a réussi à le faire en Antarctique, pourquoi pas moi, hein ?*

Tous les experts en programmes malveillants que je connais ont un jour oublié ce que faisait là un certain fichier, ont cliqué dessus pour le voir et ensuite compris qu'ils avaient exécuté un quelconque logiciel malveillant qu'ils étaient censés examiner. Je sais cela parce que ça m'est arrivé une fois avec un PDF dans lequel je savais qu'il y avait quelque chose de mauvais. Mes amis se sont moqués de moi, puis m'ont tous confessé discrètement qu'ils avaient déjà fait la même chose. Si quelques-uns des meilleurs spécialiste de rétro-ingénierie de logiciels malveillants ne peuvent surveiller leurs fichiers malveillants, qu'espérer de vos parents avec cette carte postale électronique qui est prétendument de vous ?

Les pièces jointes exécutables (ce qui inclut les documents Word, Excel, et les PDF) des emails que vous recevez chaque jour peuvent provenir de n'importe qui (on peut écrire à peu près ce que l'on veut dans le champ « De : » d'un email) et n'importe laquelle de ces pièces jointes pourrait prendre le contrôle de votre ordinateur aussi facilement qu'une vulnérabilité jour zéro. C'est certainement de cette façon que votre grand-mère s'est retrouvée à travailler pour des criminels russes, ou que vos concurrents anticipent tous vos plans produits. Mais dans le monde d'aujourd'hui, vous ne pourrez sûrement pas conserver un emploi de bureau si vous

refusez d'ouvrir des pièces jointes. Voilà le choix qui s'offre à vous : prendre en permanence le risque de cliquer sur un dangereux programme malveillant, ou vivre sous un pont, laissant sur la pelouse de votre ancienne maison des messages pour dire à vos enfants combien vous les aimez et combien ils vous manquent.

Les experts de la sécurité et de la vie privée sermonnent le public à propos des métadonnées et des réseaux d'échange de données, mais prendre en compte ces choses est aussi naturel que de se faire une batterie de tests sanguins tous les matins, et à peu près aussi facile. Les risques sur le plan sociétal de renoncer à notre vie privée sont énormes. Et pourtant, les conséquences pour chacun de ne pas y renoncer sont immédiatement handicapantes. Il s'agit au final d'un combat d'usure entre ce que l'on veut pour nous-mêmes et nos familles, et ce que l'on doit faire pour vivre dans notre communauté en tant qu'humains – un champ de mines monétisé par les entreprises et monitoré par les gouvernements.

Je travaille en plein là-dedans, et je ne m'en sors pas mieux. J'ai dû une fois suivre un processus pour vérifier mon identité auprès d'un informateur méfiant. J'ai dû prendre une série de photos montrant où je me trouvais ainsi que la date. Je les ai mises en ligne, et on m'a permis de procéder à l'interview. Au final, il se trouve qu'aucune de ces vérifications n'avait été envoyées, parce que j'avais oublié d'attendre la fin du chargement avant d'éteindre nerveusement mon ordinateur. « Pourquoi m'avez-vous quand même permis de vous voir ? » demandais-je à ma source. « Parce qu'il n'y a que vous qui pourrait faire une chose aussi stupide », m'a-t-il répondu.

Touché.

Mais si cela m'arrive à moi, une adulte relativement bien entraînée qui fait attention à ce genre de sujets systématiquement, quelle chance ont les gens avec de vrais

boulots et de vraies vies ?

## Enfin, c'est la culture qui est cassée.

Il y a quelques années, j'ai rencontré plusieurs personnes respectées qui travaillent dans la confidentialité et la sécurité logicielle et je leur ai posé une question. Mais d'abord j'ai dû expliquer quelque chose : « La plupart des gens n'ont pas de droits d'administration sur les ordinateurs qu'ils utilisent. »



(Crédit image :

[amelungc](#), licence CC BY 2.0)

C'est-à-dire que la plupart des gens qui utilisent un ordinateur dans le monde n'en sont pas propriétaires... Que ce soit dans un café, à l'école, au travail, installer une application bureautique n'est pas directement à la portée d'une grande partie du monde. Toute les semaines ou toutes les deux semaines, j'étais contacté par des gens prêts à tout pour améliorer la sécurité et les options de confidentialité, et j'ai essayé de leur apporter mon aide. Je commençais par « Téléchargez le... » et on s'arrêtait là. Les gens me signalaient ensuite qu'ils ne pouvaient pas installer le logiciel sur leur ordinateur. En général parce que le

département informatique limitait leurs droits dans le cadre de la gestion du réseau. Ces gens avaient besoin d'outils qui marchaient sur ce à quoi ils avaient accès, principalement un navigateur.

Donc la question que j'ai posée aux hackers, cryptographes, experts en sécurité, programmeurs, etc. fut la suivante : quelle est la meilleure solution pour les gens qui ne peuvent pas télécharger de nouveau logiciel sur leurs machines ? La réponse a été unanime : aucune. Il n'y a pas d'alternative. On me disait qu'ils feraient mieux de discuter en texte brut, « comme ça ils n'ont pas un faux sentiment de sécurité ». À partir du moment où ils n'ont pas accès à de meilleurs logiciels, ils ne devraient pas faire quoi que ce soit qui puisse déranger les gens qui les surveillent. Mais, expliquais-je, il s'agit d'activistes, d'organiseurs, de journalistes du monde entier qui ont affaire à des gouvernements et des sociétés et des criminels qui peuvent vraiment leur faire du mal, ces gens sont vraiment en danger. On me répondait alors que dans ce cas, ils devraient s'acheter leurs propres ordinateurs.

Et voilà, c'était ça la réponse : être assez riche pour acheter son propre ordinateur, ou bien littéralement tout laisser tomber. J'ai expliqué à tout le monde que ce n'était pas suffisant, j'ai été dénigrée lors de quelques joutes verbales sans conséquences sur Twitter, et je suis passée à autre chose. Peu de temps après, j'ai compris d'où venait l'incompréhension. Je suis retourné voir les mêmes experts et j'ai expliqué : dans la nature, dans des situations vraiment dangereuses – même quand les gens sont traqués par des hommes avec des armes – quand le chiffrement et la sécurité échouent, personne n'arrête de parler. Ils espèrent seulement ne pas se faire prendre.

La même impulsion humaine qui nous pousse vers le hasard et les loteries depuis des milliers d'années soutient ceux qui luttent même quand les chances sont contre eux. « Peut-être



bien que je m'en sortirai, autant essayer ! » Pour ce qui est de l'auto-censure des conversations dans une infrastructure hostile, les activistes non techniques s'en sortent de la même manière que les Anons, ou que les gens à qui l'on dit de se méfier des métadonnées, ou des réseaux d'échanges de données, ou de ce premier message avant que l'encodage OTR ne s'active. Ils foirent.

Cette conversation a été un signal d'alerte pour quelques personnes de la sécurité qui n'avaient pas compris que les personnes qui devenaient activistes et journalistes faisaient systématiquement des choses risquées. Certains ont rallié mon camp, celui où on perd son temps à des combats futiles sur Twitter et ils ont pris conscience que quelque chose, même quelque chose d'imparfait, pouvait être mieux que rien. Mais beaucoup dans le domaine de la sécurité sont toujours dans l'attente d'un monde parfait dans lequel déployer leur code parfait.

Alors apparaît l'*Intelligence Community* (Communauté du renseignement), ils s'appellent entre eux le IC. Nous pourrions trouver ça sympathique s'ils arrêtaient d'espionner tout le monde en permanence, et eux aimeraient bien que l'on cesse de s'en plaindre. Après avoir passé un peu de temps avec eux, je pense savoir pourquoi ils ne se préoccupent pas de ceux qui se plaignent. Les IC font partie des humains les plus surveillés de l'histoire. Ils savent que tout ce qu'ils font est passé au peigne fin par leurs pairs, leurs patrons, leurs avocats, d'autres agences, le président, et parfois le Congrès. Ils vivent surveillés, et ne s'en plaignent pas.

Dans tous les appels pour augmenter la surveillance, les fondamentaux de la nature humaine sont négligés. Vous n'allez pas apprendre aux espions que ce n'est pas bien en faisant encore plus qu'eux. Il y aura toujours des failles, et tant qu'elles existeront ou pourront être utilisées ou interprétées, la surveillance sera aussi répandue que possible. Les humains sont des créatures généralement

égocentriques. Les espions, qui sont humains, ne comprendront jamais pourquoi vivre sans vie privée est mal aussi longtemps qu'ils le feront.

Et pourtant ce n'est pas cela le pire. La catastrophe culturelle qu'ils provoquent rend plus facile leur boulot d'épier le monde. Les aspects les plus dérangeants des révélations, ce sont le marché des failles *0 day*, l'accumulation des moyens de les exploiter, l'affaiblissement des standards. La question est de savoir qui a le droit de faire partie de ce « nous » qui est censé être préservé de ces attaques, écoutes et décryptages et profilages. Quand ils ont attaqué Natanz avec [Stuxnet](#) et laissé tous les autres centres nucléaires vulnérables, nous avons été tranquillement avertis que le « nous » en question commençait et finissait avec l'IC lui-même. Voilà le plus grand danger.

Quand le IC ou le [DOD](#) ou le pouvoir exécutif sont les seuls vrais Américains, et que le reste d'entre nous ne sommes que des Américains de deuxième classe, ou pire les non-personnes qui ne sont pas associées aux États-Unis, alors nous ne pouvons que perdre toujours plus d'importance avec le temps. À mesure que nos désirs entrent en conflit avec le IC, nous devenons de moins en moins dignes de droits et de considération aux yeux du IC. Quand la NSA accumule des moyens d'exploiter les failles, et que cela interfère avec la protection cryptographique de notre infrastructure, cela veut dire qu'exploiter des failles contre des gens qui ne sont pas de la NSA ne compte pas tellement. Nous sécuriser passe après se sécuriser eux-mêmes.

En théorie, la raison pour laquelle nous sommes si gentils avec les soldats, que nous avons pour habitude d'honorer et de remercier, c'est qu'ils sont supposés se sacrifier pour le bien des gens. Dans le cas de la NSA, l'inverse s'est produit. Notre bien-être est sacrifié afin de rendre plus aisé leur boulot de surveillance du monde. Lorsque cela fait partie de la culture du pouvoir, on est en bonne voie pour que cela

débouche sur n'importe quel abus.

Mais le plus gros de tous les problèmes culturels repose toujours sur les épaules du seul groupe que je n'aie pas encore pris à partie – les gens normaux, qui vivent leurs vies dans cette situation démentielle. Le problème des gens normaux avec la technologie est le même qu'avec la politique, ou la société en général. Les gens pensent être isolés et sans pouvoir, mais la seule chose qui maintient les gens seuls et sans pouvoir est cette même croyance. Ceux qui travaillent ensemble ont un énorme et terrible pouvoir. Il existe certainement une limite à ce que peut faire un mouvement organisé de personnes qui partagent un rêve commun, mais nous ne l'avons pas encore trouvée.

Facebook et Google semblent très puissants, mais ils vivent à peu près à une semaine de la ruine en permanence. Ils savent que le coût de départ des réseaux sociaux pris individuellement est élevé, mais sur la masse, c'est une quantité négligeable. Windows pourrait être remplacé par quelque chose de mieux écrit. Le gouvernement des États-Unis tomberait en quelques jours devant une révolte générale. Il n'y aurait pas besoin d'une désertion totale ou d'une révolte générale pour tout changer, car les sociétés et le gouvernement préféreraient se plier aux exigences plutôt que de mourir. Ces entités font tout ce qu'elles peuvent pour s'en sortir en toute impunité – mais nous avons oublié que nous sommes ceux qui les laissons s'en sortir avec ces choses.

Si les ordinateurs ne satisfont pas nos besoins de confidentialité et de communication, ce n'est pas en raison d'une quelconque impossibilité mathématique. Il existe un grand nombre de systèmes qui pourraient chiffrer nos données de façon sécurisée et fédérée, nous disposons de nombreuses façons de retrouver la confidentialité et d'améliorer le fonctionnement par défaut des ordinateurs. Si ce n'est pas ainsi que les choses se passent en ce moment c'est parce que nous n'avons pas exigé qu'il en soit ainsi, et non pas parce

que personne n'est assez malin pour que ça arrive.

C'est vrai, les geeks et les PDG et les agents et les militaires ont bousillé le monde. Mais en fin de compte, c'est l'affaire de tous, en travaillant ensemble, de réparer le monde.

---

## Tests contre Bogues : une guerre sans fin (Libres conseils 13/42)

Chaque jeudi à 21h, rendez-vous sur [le framapad de traduction](#), le travail collaboratif sera ensuite publié ici même.

Traduction Framalang : Floxy, ga3lig, goofy, Astalaseven, Slystone, okram, KoS, Lycoris, 4nti7rust, peupleLa, Luc Didry, + Julius22

## Même en multipliant les regards, les bogues ne sautent pas aux yeux.

### Ara Pulido

*Ara Pulido est ingénieure d'essais pour Canonical, d'abord comme membre de l'équipe assurance qualité d'Ubuntu (QA team), et maintenant dans le cadre de l'équipe de certification du matériel. Même si elle a commencé sa carrière en tant que développeuse, elle a vite découvert que ce qu'elle aimait*

*vraiment, c'était tester les logiciels. Elle est très intéressée par les nouvelles techniques d'analyse et tente d'utiliser son savoir-faire pour améliorer Ubuntu.*

## **Les tests maison ne suffisent pas**

Je me suis impliquée dans le logiciel libre dès le début de mes études à l'Université de Grenade. Là-bas, avec des amis, nous avons fondé un groupe local d'utilisateurs de Linux et organisé plusieurs actions pour promouvoir le logiciel libre. Mais, depuis que j'ai quitté l'université, et jusqu'à ce que je commence à travailler chez Canonical, ma carrière professionnelle s'est déroulée dans l'industrie du logiciel propriétaire, d'abord comme développeuse puis comme testeuse.

Lorsque l'on travaille au sein d'un projet de logiciel propriétaire, les ressources pour tester sont très limitées. Une petite équipe reprend le travail initié par les développeurs avec les tests unitaires, utilisant leur expérience pour trouver autant de bogues que possible afin de mettre à disposition de l'utilisateur final un produit aussi abouti que possible. Dans le monde du logiciel libre, en revanche, tout est différent.

Lors de mon embauche chez Canonical, hormis la réalisation de mon rêve d'avoir un travail rémunéré au sein d'un projet de logiciel libre, j'ai été émerveillée par les possibilités des activités de test dans le cadre d'un tel projet. Le développement du produit s'effectue de manière ouverte, et les utilisateurs ont accès au logiciel dès son commencement, ils le testent et font des rapports de bogues dès que c'est nécessaire. C'est un nouveau monde rempli de beaucoup de possibilités pour une personne passionnée par les tests. Je voulais en profiter au maximum.

Comme beaucoup de personnes, je pensais que les tests « maison », c'est-à-dire l'utilisation par soi-même du logiciel que l'on envisage de mettre à disposition, était

l'activité de test la plus importante qu'on puisse mener dans le logiciel libre. Mais si, selon la formule de Raymond dans [La cathédrale et le bazar](#) « avec suffisamment d'observateurs, tous les bogues sautent aux yeux », alors comment se fait-il qu'avec ses millions d'utilisateurs Ubuntu comporte encore des bogues sérieux à chaque nouvelle version ?

La première chose dont je me suis aperçue quand j'ai commencé à travailler chez Canonical c'est que les activités de test organisées étaient rares ou inexistantes. Les seules sessions de test qui étaient d'une certaine façon organisées se présentaient sous la forme de messages électroniques envoyés à une liste de diffusion, manière de battre le rappel pour tester un paquetage dans la version de développement d'Ubuntu. Je ne pense pas que cela puisse être considéré comme une vraie procédure de test, mais simplement comme une autre forme de « test maison ». Cette sorte de test génère beaucoup de doublons, car un bogue facile à débusquer sera documenté par des centaines de personnes. Malheureusement le bogue potentiellement critique, vraiment difficile à trouver, a de bonnes chances de passer inaperçu en raison du bruit créé par les autres bogues, et ce même si quelqu'un l'a documenté.

## En progrès

La situation s'améliore-t-elle ? Sommes-nous devenus plus efficaces pour les tests au sein des projets de développement libre ? Oui, j'en suis convaincue.

Pendant les derniers cycles de développement d'Ubuntu, nous avons commencé bon nombre de sessions de test. La gamme des objectifs pour ces sessions est large, elle comprend des domaines comme de nouvelles fonctionnalités de bureau, des tests de régression, des tests de pilotes X.org ou des tests de matériel d'ordinateur portable. Les résultats sont toujours suivis et ils s'avèrent vraiment utiles pour les développeurs, car ils leur permettent de savoir si les nouveautés

fonctionnent correctement, au lieu de supposer qu'elles fonctionnent correctement à cause de l'absence de bogues.

En ce qui concerne les outils d'assistance aux tests, beaucoup d'améliorations ont été apportées :

- Apport(1) a contribué à augmenter le niveau de détail des bogues signalés concernant Ubuntu : les rapports de plantage incluent toutes les informations de débogage et leurs doublons sont débusqués puis marqués comme tels ; les utilisateurs peuvent signaler des bogues sur base de symptômes, etc.
- Le Launchpad(2), avec ses connexions en amont, a permis d'avoir une vue complète des bogues – sachant que les bogues qui se produisent dans Ubuntu se situent généralement dans les projets en amont, et permet aux développeurs de savoir si les bogues sont en cours de résolution.
- Firefox, grâce à son programme et à son extension Test Pilot, mène des tests sans qu'on ait à quitter le navigateur(3). C'est, à mon sens, une bien meilleure façon de rallier des testeurs qu'une liste de diffusion ou un canal IRC.
- L'équipe Assurance Qualité d'Ubuntu teste le bureau en mode automatique et rend compte des résultats toutes les semaines(4), ce qui permet aux développeurs de vérifier très rapidement qu'il n'y a pas eu de régression majeure pendant le développement.

Cependant, malgré l'amélioration des tests dans les projets de logiciel libre il reste encore beaucoup à faire.

## **Pour aller plus loin**

Les tests nécessitent une grande expertise, mais sont encore considérés au sein de la communauté du le logiciel libre comme une tâche ne demandant pas beaucoup d'efforts. L'une des raisons pourrait être que la manière dont on les réalise est

vraiment dépassée et ne rend pas compte de la complexité croissante du monde du logiciel libre durant la dernière décennie. Comment est-il possible que, malgré la quantité d'innovations amenées par les communautés du logiciel libre, les tests soient encore réalisés comme dans les années 80 ? Il faut nous rendre à l'évidence, les scénarios de tests sont ennuyeux et vite obsolètes. Comment faire grandir une communauté de testeurs supposée trouver des bogues avérés si sa tâche principale est de mettre à jour les scénarios de test ?

Mais comment améliorer la procédure de test ? Bien sûr, nous ne pouvons pas nous débarrasser des scénarios de test, mais nous devons changer la façon dont nous les créons et les mettons à jour. Nos testeurs et nos utilisateurs sont intelligents, alors pourquoi créer des scripts pas-à-pas ? Ils pourraient aisément être remplacés par une procédure de test automatique. Définissons plutôt une liste de tâches que l'on réalise avec l'application, et certaines caractéristiques qu'elle devrait posséder. Par exemple, « l'ordre des raccourcis dans le lanceur doit pouvoir être modifié », ou « le démarrage de LibreOffice est rapide ». Les testeurs trouveront un moyen de le faire, et créeront des scénarios de test en parallèle des leurs.

Mais ce n'est pas suffisant, nous avons besoin de meilleurs outils pour aider les testeurs à savoir ce qu'ils testent, où et comment. Pourquoi ne pas avoir des API (interfaces de programmation) qui permettent aux développeurs d'envoyer des messages aux testeurs à propos des nouvelles fonctionnalités ou des mises à jour qui doivent être testées ? Pourquoi pas une application qui nous indique quelle partie du système doit être testée ? en fonction des tests en cours ? Dans le cas d'Ubuntu, nous avons les informations dans le Launchpad (il nous faudrait aussi des données sur les tests, mais au moins nous avons des données sur les bogues). Si je veux démarrer une session de test d'un composant en particulier



j'apprécierais vraiment de savoir quelles zones n'ont pas encore été testées ainsi qu'une liste des cinq bogues comptant le plus de doublons pour cette version en particulier afin d'éviter de les documenter une fois de plus. J'aimerais avoir toutes ces informations sans avoir à quitter le bureau que je suis en train de tester. C'est quelque chose que Firefox a initié avec Test Pilot, bien qu'actuellement l'équipe rassemble principalement les données sur l'activité du navigateur.

La communication entre l'amont et l'aval et vice-versa doit aussi être améliorée. Pendant le développement d'une distribution, un bon nombre des versions amont sont également en cours de développement, et ont déjà une liste des bogues connus. Si je suis un testeur de Firefox sous Ubuntu, j'aimerais avoir une liste des bogues connus aussitôt que le nouveau paquet est poussé dans le dépôt. Cela pourrait se faire à l'aide d'une syntaxe reconnue pour les notes de versions, syntaxe qui pourrait ensuite être facilement analysée. Les rapports de bogue seraient automatiquement remplis et reliés aux bogues amont. Encore une fois, le testeur devrait avoir facilement accès à ces informations, sans quitter son environnement de travail habituel.

Les tests, s'ils étaient réalisés de cette manière, permettraient au testeur de se concentrer sur les choses qui comptent vraiment et font de la procédure de test une activité qualifiée ; se concentrer sur les bogues cachés qui n'ont pas encore été découverts, sur les configurations et environnements spéciaux, sur la création de nouvelles manières de casser le logiciel. Et, in fine, s'amuser en testant.

## Récapitulons

Pour ce que j'en ai vu ces trois dernières années, les tests ont beaucoup progressé au sein d'Ubuntu et des autres projets de logiciels libres dans lesquels je suis plus ou moins

impliquée, mais ce n'est pas suffisant. Si nous voulons vraiment améliorer la qualité du logiciel libre, nous devons commencer à investir dans les tests et innover dans la manière de les conduire, de la même façon que nous investissons dans le développement. Nous ne pouvons pas tester le logiciel du XXIe siècle avec les techniques du XXe siècle. Nous devons réagir. Qu'il soit open source ne suffit plus à prouver qu'un logiciel libre est de bonne qualité. Le logiciel libre sera bon parce qu'il est open source et de la meilleure qualité que nous puissions offrir.

1 <http://wiki.ubuntu.com/Appport>

2 <http://launchpad.net>

3 <http://testpilot.mozillalabs.com>

4 <http://reports.qa.ubuntu.com/reports/desktop-testing/natty>