Sécurité, facilité d'usage, et les utilisateurs non-informaticiens

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 août 2014

https://www.bortzmeyer.org/securite-facilite.html

La question de la facilité d'usage des solutions de sécurité informatique revient régulièrement sur le devant de la scène. S'il y a si peu d'utilisateurs de l'Internet qui se servent de la cryptographie ou d'autres mesures techniques qui les protégeraient, ne serait-ce pas parce que les techniques en question sont trop complexes, amenant les utilisateurs à préférer prendre des risques plutôt que de tenter de les maîtriser ?

La question revient souvent, mais elle est très ancienne. L'article qui avait été le premier à pointer ce problème de l'utilisabilité est « "*Why Johnny Can't Encrypt*" <http://www.gaudior.net/alma/johnny.pdf> » d'Alma Whitten et J. D. Tygar1, en... 1999 Peu de progrès dans le débat depuis, à part une large reconnaissance, par la communauté des gens qui travaillent dans la sécurité informatique, que l'utilisabilité est en effet cruciale : la meilleure solution de sécurité du monde ne servira à rien si elle n'est pas utilisée car trop compliquée. Or, les révélations du héros Snowden ont rappelé l'importance de se protéger, face à l'espionnage massif qui a lieu sur l'Internet. La cryptographie fait donc l'objet d'un regain d'intérêt. Et il y a également une reprise du débat critique. Il y a deux semaines, Matthew Green avait écrit « "*What's the matter with PGP?*" <http://blog.cryptographyengineering.com/2014/08/whats-matter-with-pgp.html> » où il estimait que ce système devait disparaître (attention, son article contient bien des confusions, notamment entre le logiciel GPG et la norme OpenPGP, et des malhonnêtetés intellectuelles comme de comparer la longueur d'une clé avec celle d'un certificat).

Plus récemment, en France, Okhin à Pas Sage en Seine <http://www.passageenseine.org/> s'est attaqué aux informaticiens <https://about.okhin.fr/posts/MakeDataloveNotCyberwar/> en critiquant vigoureusement « les barbus auto proclamés gourous des internets, cyber hactivistes, hackers, sysadmin et autre » et en annonçant que « Ce qui est cassé ce sont nos égos, nos réactions de sociopathes nihilistes face à un problème politique et social. » Ce thème a ensuite été repris par Barbayellow dans « Sécurité : pourquoi ça ne marche pas <http://blog.barbayellow.com/2014/08/16/securite-pourquoi-ca-ne-marche-pas/> » et par Numendil en « Chers nous... <http://pixellibre.net/2014/08/chers-nous/> » ainsi que par Tris Acatrinei dans « Une histoire de privilèges <http://www.hackersrepublic.org/culture-du-hacking/une-histoire-de-privileges> ».

Je vous laisse lire tous ces articles avant de voir le mien. Je ne vais pas essayer de réfuter la thèse « L'utilisabilité est essentielle pour la sécurité » car elle est exacte. La grande majorité des attaques réussies ne viennent pas d'une brillante réussite technique contre un algorithme de cryptographie, elles viennent d'erreurs commises par les utilisateurs. Mais je voudrais discuter un peu d'autres affirmations, comme de prétendre que la balle est à 100 % dans le camp des informaticiens/hackers/programmeurs, ou comme l'idée selon laquelle la difficulté d'utilisation des logiciels de sécurité est essentiellement due au manque d'intérêt des techniciens pour l'utilisabilité.

Alors, commençons par un problème pratique : clamer qu'il faut améliorer les logiciels, notamment du point de vue de l'ergonomie, c'est bien mais c'est vain. Encore faudrait-il dire en quoi on peut améliorer l'ergonomie existante. J'ai lu attentivement tous les articles cités plus haut et il y a beaucoup plus de temps passé à critiquer les développeurs (orgueilleux, autistes, méprisants, privilégiés) qu'à discuter de solutions concrètes. Il faut dire que le problème n'est pas trivial. Même un informaticien professionnel a du mal à configurer HTTPS sur son serveur Web, ou à utiliser PGP (et, plus encore, à l'utiliser tous les jours sans jamais faire d'erreur). C'est qu'il ne s'agit pas uniquement de compétences : même quand on a les compétences de base, on n'a pas forcément le temps. Donc, si on pouvait rendre ces logiciels plus faciles d'accès, ne nécessitant pas un week-end de travail <https://www.bortzmeyer.org/nouvelle-cle-pgp.html> pour générer une clé PGP conforme aux exigences de la cryptographie moderne, cela profiterait à tout le monde <https://www.bortzmeyer.org/accessibilite.html>, de M. Michu au gourou informaticien. La question est « comment ? » Certaines améliorations sont assez évidentes (pour PGP, avoir des choix par défaut corrects, au lieu d'obliger les utilisateurs à demander à des experts en cryptographie comment les améliorer). D'autres le sont nettement moins. J'aimerai voir moins d'articles « les informaticiens sont méchants, ils font des logiciels peu utilisables » et plus d'articles « proposition concrète d'amélioration / de nouveau logiciel ». (Je ne suis pas exigeant, je ne demande pas de logiciel fini.) Donc, oui, les logiciels d'aujourd'hui peuvent et doivent être considérablement améliorés. Mais tout le monde en est convaincu ! Ce qu'il faudrait faire désormais, c'est avancer un peu et proposer des améliorations précises, et je n'en ai pas encore vu. Et c'est un problème difficile, loin des Yakafokon. L'article « "*Why Johnny Can't Encrypt*" », cité plus haut, explique bien pourquoi le problème des interfaces utilisateur en sécurité est différent des problèmes d'ergonomie habituels. (Il explique également bien pourquoi le manque d'interface graphique est un faux problème : « "*All this failure is despite the fact that PGP 5.0 is attractive, with basic operations neatly represented by buttons with labels and icons, and pull-down menus for the rest*" ».)

Comme exemple de difficulté à concevoir une interface utilisateur simple et efficace, prenons **le** gros problème de la cryptographie, la gestion des clés. Une des difficultés de PGP est qu'il faut récupérer la clé publique de son correspondant (typiquement via un serveur de clés), la valider (c'est l'opération essentielle, sans laquelle il n'y a plus guère de sécurité) et la stocker. Ces opérations compliquées sont difficiles à expliquer et une erreur est vite arrivée (par exemple d'accepter une clé sans l'avoir vraiment vérifiée). OTR fonctionne sur un principe proche : comme OTR, contrairement à PGP, est limité aux communications synchrones, on n'a pas à récupérer la clé de son correspondant, mais il faut toujours la vérifier. Peut-on faire mieux ? Tout dépend de ce qu'on est prêt à abandonner en échange. Dans une discussion sur Twitter, Okhin <https://twitter.com/okhin> vantait le modèle de clés de X.509 (utilisé dans TLS et donc dans HTTPS) comme étant plus simple : l'utilisateur n'a aucune vérification à faire. Mais Okhin oublie de dire que le prix à payer est la sous-traitance complète de sa sécurité aux autorités de certification qui valent... ce qu'elles valent. Si on veut protéger sa vie privée, devoir faire confiance à ces entreprises n'est pas l'idée la plus géniale qui soit. Autre solution au problème de la gestion de clés, SSH et son principe TOFU ("*Trust On First Use*"). Il est amusant que SSH soit si rarement cité par ceux qui réclament des logiciels de sécurité plus faciles à utiliser. Car SSH est justement une réussite de l'interface utilisateur, puisqu'il est plus simple que ne l'était son concurrent non sécurisé, telnet. Si on veut un exemple de conception réussie, il ne faut pas regarder TLS ou OTR mais certainement plutôt le grand succès qu'est SSH (cf. annexe A.3 du RFC 5218 [1]). Mais, là encore, rien n'est parfait. SSH est facile

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5218.txt

grâce au TOFU : la première fois qu'on se connecte à un nouveau serveur, la clé est vérifiée (en théorie...) et elle est automatiquement mémorisée pour la suite. Ce principe est assez bon (c'est un compromis raisonnable entre utilisabilité et sécurité) mais il a aussi des inconvénients : vulnérabilité lors du premier usage, difficulté à changer les clés par la suite...

Outre la notion de **compromis**, essentielle en sécurité (mais trop rarement citée dans les articles réclamant des logiciels plus simples), il y a une autre notion essentielle, et souvent oubliée : l'**éducation**. En général, elle est balayée avec des phrases du genre « il n'y a pas besoin d'être ingénieur pour conduire une voiture ». Mais c'est une illusion : bien sûr qu'il faut apprendre beaucoup de choses pour conduire une voiture. Pas besoin d'être ingénieur, certes, mais n'importe quel conducteur a dû apprendre beaucoup, simplement, comme c'est aujourd'hui un savoir banalisé, il ne s'en rend pas compte (voir à ce sujet l'hilarant texte sur le support de General Motors <http://www.gluckman.com/harry/generalmotors.htm>). Dans une société où de plus en plus de choses dépendent de l'informatique, dire que les outils modernes doivent être accessibles sans aucune éducation, c'est **illusoire**. Il faut au contraire développer une **littératie numérique**, qui implique un minimum de choses sur la sécurité. L'apprentissage de cette littératie va nécessiter des efforts des deux côtés. Réclamer des systèmes informatiques qui soient utilisables sans aucune formation, ni effort de la part des utilisateurs, va entretenir des illusions.

Quand Barbayellow écrit « [Le métier des journalistes], ce n'est pas de savoir comment fonctionne un ordinateur, un serveur ou même Internet », il se trompe. Aujourd'hui, où une si grande partie des activités humaines (le journalisme, mais pas uniquement), se passe sur l'Internet, il **faut** y connaître quelque chose. Dire que les journalistes ne devraient pas connaître l'Internet, c'est comme s'il disait « connaître le droit n'est pas notre métier », alors que tant de métiers aujourd'hui nécessitent forcément une culture juridique minimale. Et dire qu'il n'est pas nécessaire d'apprendre l'Internet en 2014, c'est comme dire en 1450 qu'il n'est pas nécessaire d'apprendre à lire si on n'est pas un professionnel de l'édition...

"*A translation in english of this article was kindly done by Pete Dushenski (footnotes are not mine).*" <http://contravex.com/2014/08/26/infosec-education-because-stephane-bortzmeyer-is-lazy-and-im-n>