Avec des logiciels pareils sur l'App Store, plus besoin de virus

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 9 décembre 2010

https://www.bortzmeyer.org/malware-iphone.html

Dave Winer raconte sur son excellent blog <http://scripting.com/stories/2010/11/15/theTechIndustryIsAVirus.html>, une curieuse aventure qui lui est arrivée avec son iPhone. Il a installé une nouvelle application (de partage de photos) sur son "*smartphone*" et découvert que celle-ci lisait son carnet d'adresses avant de l'envoyer au site maître (le tout, évidemment, sous couvert d'améliorer "*the user experience*"). Par delà l'immoralité de l'entreprise Path.com, cet incident donne à réfléchir sur le modèle de sécurité des "*smartphones*".

L'auteur s'indigne en effet de ce qu'aucun avertissement ne soit apparu. On installe une application non libre censée permettre de partager des photos et elle accède au carnet d'adresses, et elle l'envoie sans demander d'autorisation, sans même qu'on soit prévenu ! Tout logiciel qui fait cela est habituellement nommé "*malware*" mais, ici, comme il est distribué sur l'App Store, il échappe à cette qualification, pourtant bien méritée.

Comment l'utilisateur d'un "*smartphone*" est-il protégé contre ce genre d'attaques ? L'essentiel de la (passionnante) discussion dans les commentaires a porté sur ce point. Sur iPhone, le système installe une application sans limiter ses privilèges, sans dire à l'utilisateur ce que cette application va faire. C'est pareil sur un Unix normal, me direz-vous. Certes, mais la plupart des logiciels (la totalité, dans mon cas) qu'on utilise sur Unix sont du logiciel libre et le fait que le code source soit accessible (même s'il n'est pas toujours lu) limite sérieusement les tentations pour les auteurs. Au contraire, avec le logiciel privateur, pas de code source et l'entreprise qui l'écrit peut donc avoir envie d'en profiter, d'en abuser.

Normalement, tout logiciel distribué sur l'App Store est validé par Apple. Ce pouvoir exorbitant a mené à bien des dérapages (croisade à l'iranienne <http://www.wired.com/gadgetlab/2009/06/apple-no-porn-allowed-in-iphones-app-store/> contre la pornographie, censure des documentations <http://mediawatch.dk/artikel/apple-bans-mag-app-android-app-store> sur les systèmes concurrents) mais, en théorie, il devait protéger l'utilisateur de tas de choses désagréables qu'on a sur un ordinateur Windows et qu'on ne voudrait pas trouver sur son téléphone, notamment les virus. L'expérience malheureuse de Dave Winer montre que cela ne marche pas. Il est probable que

l'évaluation que fait Apple des logiciels ne se base que sur des considérations business et certainement pas sur le respect de la vie privée des futurs utilisateurs.

Existe-t-il des meilleurs mécanismes pour prévenir l'utilisateur de ce que va faire l'application qui l'installe ? Sans doute mais, comme souvent en matière de sécurité, il n'y a pas de solution idéale, ledit utilisateur étant souvent le maillon faible. Sur Android, lors de l'installation d'une application (qu'elle vienne du "*Market*" officiel ou pas), l'utilisateur est prévenu des privilèges que demandera l'application. Trouvé sur un article sur la sécurité d'Android <http://securitymusings.com/article/2078/security-threats-in-android-or-not>, voici un exemple de ce qu'affiche le téléphone : Ce mécanisme n'est pas parfait. D'abord, comme souvent dès qu'on demande à l'utilisateur, celui-ci cliquera souvent Oui sans même réfléchir, quelle que soit la longueur de la liste des privilèges demandés. Et, même s'il ne faut pas être informaticien pour comprendre qu'un logiciel qui affiche les horaires des prochains bus n'a pas besoin d'avoir accès au carnet d'adresses, en pratique, le temps manque et le cerveau ralentit dès qu'il est confronté à un ordinateur. À la décharge de l'utilisateur, il faut ajouter que le mécanisme d'approbation est binaire : soit on accepte toute la liste (et, avec certaines applications, elle est longue), soit on ne peut pas utiliser l'application du tout. Il n'y a hélas pas moyen de dire « Je veux bien que tu lises le GPS mais ne touche pas au carnet d'adresses ». (Il existe des solutions non-standard comme celle de CyanogenMod <http://www.frandroid.com/actualites-generales/71353_cyanogenmod-permet-maintenant-de-supprimer-des-permissions-aux-applications/>.)

Autre limitation de ce mécanisme de sécurité : il ne limite pas ce que fait l'application qui avait un accès. Par exemple, il est tout à fait normal qu'un client SIP ait accès au carnet d'adresses mais ensuite, je ne veux pas qu'il l'envoie à un tiers, ce que font pourtant certains logiciels <http://www.lemonde.fr/technologies/article/2010/12/20/ces-donnees-privees-que-les-applications-mobiles-t
1455982_651865.html> (voir une bonne étude à ce sujet <http://blogs.wsj.com/wtk-mobile/>).

Au fait, s'il y a des utilisateurs de MeeGo qui lisent ceci, comment cela se passe-t-il sur ce système ? Ces problèmes de donner à des applications des privilèges, avec une certaine granularité, me rappelle mon travail passé comme ingénieur système sur VMS : contrairement au modèle Unix binaire (root / pas root, avec root qui a tous les droits), VMS avait un système de permissions très riche, où on pouvait ne donner à une application que certains droits. Une des faiblesses de ce mécanisme était la longueur de la liste des droits possibles, que peu de gens maîtrisaient. Une autre faiblesse était que certains droits, sans que cela soit clairement documenté, permettait d'en acquérir d'autres (je me souviens bien du privilège « "*Change Mode to Kernel*" » qui permettait d'écrire dans les métadonnées du processus... où la liste des droits était stockée). Android a-t-il de telles faiblesses ? Si une chose est sûre en matière de sécurité, c'est que concevoir un système invulnérable est fort difficile : des failles inattendues surgissent toujours (pour Android, voir par exemple Soundminer <http://www.lemondeinformatique.fr/actualites/lire-soundminer-un-malware-espionne-et-vole-les-donnees-des-smartphones-android-32698
html>).

Sur la question de limiter les privilèges d'une application inconnue et peut-être malveillante, il y a aussi :

- Les solutions à base de machines virtuelles comme Qubes <https://www.bortzmeyer.org/qubes.html>,
- Les mécanismes à base de privilèges pour les applications, comme celui de VMS cité plus haut ou comme cela existe sur Linux <http://linux-attitude.fr/post/capabilities> (voir une bonne discussion <http://lwn.net/Articles/421671/> suite à un très intéressant article de Brad Spengler <http://forums.grsecurity.net/viewtopic.php?f=7&t=2522>, sur les limites de ces capacités).
- Le modèle de sécurité d'Android est très bien décrit dans la documentation officielle <http://source.android.com/tech/security/index.html>.

Merci à Vincent-Xavier Jumel pour ses suggestions.

Notez qu'Apple n'a nullement corrigé le problème qui reste, début 2012, aussi préoccupant qu'avant <http://thenextweb.com/insider/2012/02/15/what-ios-apps-are-grabbing-your-data-why-they-d>.