La tête dans les nuages mais les pieds sur terre

GMail, Google Apps, Zoho, Flickr, Del.icio.us, Box.net, Wuala, DropBox, Plaxo… Ça vous parle ? Vous savez, ces applications en ligne (ou Web Apps) pratiques et séduisantes qui poussent comme des champignons aux quatre coins de la Toile, souvent accompagnées de la mention *beta* pour faire hi-tech. On a même trouvé un terme pour englober tout ce petit monde, le très à la mode Cloud Computing, soit l'informatique *dans les nuages* ou *dématérialisée*[1].

Que celui qui n'a pas un compte chez l'un de ces services en ligne me jette la première pierre. Reconnaissons qu'il est fort commode, notamment pour qui travaille sur plusieurs postes ou de façon nomade, de pouvoir accéder à ses courriels, à ses documents et à certaines données depuis n'importe quel PC (voire téléphone mobile) à condition de disposer d'une connexion Internet.

Mais n'est-on pas en droit de s'inquiéter de savoir que tant de nos données se baladent on ne sait trop où dans l'espace virtuel ? Plaxo, par exemple, avait été critiqué à ses débuts car assez porté sur le spam et l'exploitation peu scrupuleuse des carnets d'adresses qu'on lui confiait (même si ce service a depuis redressé la barre). Quid des documents créés avec Google Docs, des fichiers conservés chez Box.net, des mails échangés avec GMail ? De plus en plus de grands groupes ou de start-ups se lancent sur ce marché apparemment juteux et se battent pour posséder nos données.

Il y a quelque temps déjà, plusieurs voix s'élevaient contre contre ces services en ligne : Larry Ellison, le fondateur

d’Oracle, qualifiait le Cloud Computing de mode, et Richard Stallman, dans un entretien accordé au quotidien anglais The Guardian, allait plus loin en taxant ces services en ligne de pièges. Stallman mettait en garde les utilisateurs contre ces Web Apps et le stockage de données personnelles sur les serveurs d’entreprises commerciales : selon lui, confier ses données à de tels services revient à en perdre le contrôle et pose donc les mêmes problèmes que l’utilisation des logiciels propriétaires.

Stallman, qui n’a pas l’habitude de faire dans la nuance, recommandait donc de n’utiliser aucun de ses services et de leur préférer nos bonnes vieilles applis *en dur* sur lesquelles nous gardons tout contrôle. (Nul doute qu’il pensait par là à la Framakey…)

Tim O’Reilly, dans son blog, s’est lui aussi penché sur la question et a publié un billet assez fourni, dans lequel il estime qu’il faudrait appliquer au Cloud Computing les principes de l’Open Source et rappelle qu’il avait déjà mis en garde contre le verrouillage du Web, pourtant basé sur des programmes et outils Open Source, par des applications Web 2.0.

Cette question de la confidentialité, du contrôle des données et de l’indépendance de l’utilisateur face au logiciel concerne tous ceux qui ont un usage intensif du Web et de l’outil informatique, mais semble cruciale pour les adeptes du logiciel libre, très sensibles à ces questions. Comme dans le software classique, certains acteurs du Cloud Computing proposent des solutions libres. C’est le cas de Clipperz, qui a développé un gestionnaire de mots de passe et d’informations personnelles sous licence GPL.

Sur le blog de Clipperz , un des auteurs appelle les utilisateurs et les développeurs à agir pour préserver la liberté et la confidentialité *dans les nuages*, et propose quelques mesures pour que les applications Web 2.0 soient en

accord avec les valeurs du libre, du point de vue des licences et du comportements des navigateurs Internet par exemple. On y retrouve par billet interposé Richard Stallman, avec qui l'auteur s'est entretenu et qui y va lui aussi de ses conseils.

Histoire de redescendre un peu sur terre après tant de temps passé dans les nuages, nous vous présentons donc la traduction de ce billet, réalisée par notre équipe Framalang.

Liberté et protection de la vie privée en ligne : agissez !

Freedom and Privacy in the Cloud – a call for action

Marco – 30 mai 2008 – Clipperz
(Traduction Framalang : Olivier, Burbumpa et Don Rico)

Ce message traite de la liberté. La liberté de posséder vos données et la liberté d'utiliser des logiciels libres. Vous devriez aussi pouvoir exiger ces libertés et en jouir quand vous utilisez des applications web.

Si vous soutenez le mouvement du logiciel libre, vous pouvez facilement opter pour Gimp au lieu de Photoshop, pour Firefox au lieu d'Internet Explorer. Vous pouvez également protéger le caractère privé de vos données en utilisant les outils de cryptage disponibles (GPG, TrueCrypt…). Mais dès qu'il s'agit d'applications web, tout se complique.

Les avantages des applications web – ou web apps – (accessibles partout et tout le temps, mises à jour transparentes, stockage fiable, …) sont nombreux, mais bien souvent les utilisateurs perdent la liberté d'étudier, de modifier et de discuter du code source qui fait tourner ces web apps.

De plus, nous sommes contraints de confier nos données aux fournisseurs de ces web apps (marque-pages, documents rédigés,

copies des discussions, informations financières et désormais… dossiers médicaux) qui ne résident alors plus sur nos disques durs mais qui sont rangés quelque part *dans les nuages*. Ce n'est pas vraiment une situation confortable de devoir choisir entre aspect pratique et liberté.

Que l'on soit clair : les web apps sont formidables et je les adore. Mais je pense que le moment est venu de réclamer plus de liberté et de confidentialité. Voilà comment nous pouvons obtenir ces deux résultats en trois étapes.

1. Choisissez l'AGPL

Quelle est l'importance de l'AGPL ? Si vous êtes un fournisseur de services et que vos services s'appuient sur des logiciels placés sous licence AGPL, vous devez rendre le code source disponible à toute personne utilisant ce service. La FSF suggère dans ses directives de placer un lien *Source* qui renvoie à une archive contenant le code source directement dans l'interface de l'application web.

(Ne me demandez pas pourquoi la communauté des logiciels libres a mis tant de temps à réagir !)

Mesures

- Aider Clipperz à mettre au point une *suite AGPL* : un ensemble d'applications web répondant aux besoins les plus courants.
- Cette suite devrait comprendre : un traitement de texte, un logiciel de discussion, un gestionnaire de mots de passe, un carnet d'adresses, un pense-bête, un calendrier, un gestionnaire de marque-pages … Et chaque web app devra être soumise à la licence AGPL ! Vous pourrez alors oublier Google, del.icio.us, Plaxo, Meebo … à moins qu'ils ne se mettent à l'AGPL aussi.
- Nous avons déjà deux candidats pour certains postes (Ajax Chat pour les discussions en ligne et, bien sûr, Clipperz pour le gestionnaire de mots de passe), mais la

plupart des places sont encore à pourvoir !
- Aider Clipperz à diffuser les bienfaits de l'AGPL auprès des développeurs de projets web open-source. Demandez-leur de se convertir à l'AGPL.

2. Ajoutez-y une pointe de divulgation nulle de données

Les développeurs Web, comme les utilisateurs, connaissent encore assez peu les possibilités offertes par le chiffrage via un navigateur pour rendre les applications web aussi sécurisées et confidentielles que les logiciels classiques.

Chez Clipperz, nous voulons apporter une nouvelle vision que nous appelons *les web apps à divulgation nulle de données* (description plus détaillée ici) qui associe l'idée d'un hébergement auquel même l'hébergeur n'a pas accès et un ensemble de règles basées sur le credo *confidentialité absolue*.

Ce nom est aussi bien un hommage au chiffrage (une *garantie de divulgation nulle de données* est un protocole de chiffrage standard) que la promesse d'une relation particulière entre l'utilisateur et le fournisseur d'application. Le serveur hébergeant la web app peut ne rien savoir sur ses utilisateurs, pas même leurs identifiants ! Clipperz applique cette vision pour mettre en œuvre son gestionnaire de mots de passe en ligne.

Mesures

- Appliquer les techniques *divulgation nulle de données* à chaque composant de la *suite AGPL*. Convertir une application web à l'architecture *divulgation nulle* n'est pas simple, mais chez Clipperz nous avons développé un savoir-faire important et nous serons heureux de partager aussi bien ces connaissances que le code de base.

Nous pourrons ainsi finalement jouir d'un traitement de texte en ligne qui ne pourra pas lire nos documents, un logiciel de discussion qui n'enregistrera pas nos conversations, un wiki sur lequel on pourra conserver sans crainte des données importantes, etc.

- Établir et maintenir à jour une liste des Fournisseurs de Service d'Application (*NdT : ASP pour Application Service Provider en anglais*) qui hébergent la suite complète sous AGPL. Cette référence sera utile à tous ceux qui attachent de l'importance aux logiciels libres et à la confidentialité mais ne possèdent pas les compétences et les ressources pour faire tourner des web apps sur leur propre serveur.

3. Créer un navigateur plus intelligent

On y est presque, mais il nous reste encore à fournir aux utilisateurs de web apps un environnement encore plus flexible et sécurisé. Dans la pratique, du fait de l'architecture des web apps à *divulgation nulle de données*, le serveur réalise de façon générale les tâches suivantes :

- charger le code Javascript dans le navigateur de l'utilisateur (charger le programme) ;
- authentifier l'utilisateur (optionnel et par un protocole à *divulgation nulle de données*)
- rapatrier et stocker les données chiffrées demandées par le navigateur de l'utilisateur.

Logiciel libre est synonyme de contrôle total de ce qui se passe sur mon ordinateur. Se posent alors deux questions :

- Comment faire tourner une version modifiée du code Javascript à la place de celui chargé par le serveur ?
- Comment être alerté des modifications apportées au code Javascript que le serveur envoie à mon navigateur ?

J'ai récemment eu l'immense honneur d'échanger mes idées avec

Richard Stallman lui-même au sujet de ces problèmes, et il a suggéré une solution futée pour les résoudre tous les deux.

Stallman propose d'ajouter une fonctionnalité au navigateur qui permette à l'utilisateur de dire : « Quand tu charges l'URL X, utilise le code Javascript de l'URL Y comme s'il venait de l'URL X ». Si l'utilisateur fait appel à cette fonctionnalité, il peut utiliser sa propre copie du code Javascript et peut toujours échanger des données avec le serveur hébergeant l'application web.

Un navigateur possédant cette capacité pourrait aussi facilement vérifier si le script Java de l'URL X est différent du script Java sauvegardé à l'URL Y. Si l'utilisateur fait confiance à la version courante du code Javascript de l'URL X, il peut en faire une copie à l'URL Y et sera ainsi alerté de tout changement. Cette solution protège l'utilisateur du code malveillant qu'il pourrait exécuter sans le savoir dans son navigateur, du code qui pourrait voler ses données et détruire l'architecture à divulgation nulle d'information.

Mesures :

- Écrire des extensions pour les principaux navigateurs libres (Mozilla, Webkit, …) qui mettent en œuvre l'idée de Stallman.

Militer pour l'adjonction de la "suite AGPL" et des navigateurs améliorés pré-cités dans les distributions GNU/Linux.

- Continuez à lire ce blog où je posterai de nouveaux articles régulièrement.
- Faites-moi part de vos commentaires et suggestions.
- Faites passez le message au travers de vos blogs, de vos messages sur les forums, …
- Faites un don.

Et le meilleur pour la fin : comment nommeriez-vous cet

ambitieux projet ? Faites-moi part de vos idées dans les commentaires !

Notes

[1] Crédit photo : Nicholas T (Creative Commons By)