RFC 6973 : Privacy Considerations for Internet Protocols

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 juillet 2013

Date de publication du RFC : Juillet 2013

https://www.bortzmeyer.org/6973.html

La question de la vie privée est désormais une question de première importance sur l'Internet. La sortie de ce RFC en plein scandale PRISM est une coïncidence mais elle tombe bien. Pendant longtemps, la vie privée avait pourtant été plutôt négligée par les ingénieurs et cette négligence se retrouvait dans les normes techniques, par exemple les RFC, qui ne contenaient pas grand'chose sur la question. Désormais, l'importance du problème est reconnue à l'IETF et ce nouveau RFC vise à expliquer à tous les auteurs de RFC ce qu'il faut connaître de la vie privée et de sa protection, lorsqu'on conçoit un protocole réseau.

Ce n'est donc pas un texte général sur la vie privée (il en existe déjà beaucoup) même s'il explique les principes de base (le membre typique de l'IETF peut être très ignorant à ce sujet). Son objectif est de faire en sorte que les protocoles de la famille TCP/IP prennent en compte les problèmes de vie privée dès leur conception. Cela a été un long chemin. Peu de RFC traitaient explicitement cet aspect (une exception intéressante est le RFC 4941 [1]). Mais, petit à petit, entre autre grâce au travail d'Alissa Cooper, une des auteures de ce document, l'IETF est passé de « bof, c'est de la politique, ça ne nous concerne pas, et d'ailleurs la technique est neutre » à un atelier sur la protection de la vie privée (le compte-rendu figure dans le RFC 6462) puis à ce RFC 6973 qui dit en substance que tout nouveau protocole de la famille TCP/IP devrait avoir, lors de sa conception, une réflexion sur les risques qu'il pose pour la vie privée et les moyens de les limiter. S'il reste encore à l'IETF quelques partisans du « on s'en fiche, que toutes les données soient accessibles et que tout le monde soit à poil <http://www.slate.fr/monde/74668/prism-resignation> », ils se font nettement moins entendre désormais. Sans aller jusqu'à imposer une section "*Privacy considerations*" dans chaque RFC (sur le modèle de l'obligatoire section "*Security considerations*", cf. RFC 3552), ce RFC l'encourage et aide à sa rédaction.

Le problème n'est pas trivial : la vie privée est quelque chose de complexe. Et tout le monde n'a pas la même opinion sur l'importance de son respect. Sans compter que le cadre légal est très différent d'un

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4941.txt

pays à l'autre, alors que les RFC sont censés avoir une portée mondiale. Si certains RFC ne semblent pas poser de problème de vie privée (notre RFC cite le RFC 6716...) d'autres sont ou seront entièrement consacrés à ce sujet (RFC 3325) et d'autres nécessiteront un paragraphe ou une section dédiée.

La section 2 détaille les responsabilités des concepteurs de protocoles Internet. Ces protocoles sont souvent utilisés dans une grande variété de cas, parfois non prévus lors de leur conception originelle. Bref, on ne peut pas demander au rédacteur du RFC sur un nouveau protocole de prévoir tous les usages et il y a donc des limites à l'analyse qu'il peut faire sur les conséquences de « son » protocole sur la vie privée. D'autre part, une grande partie, peut-être la majorité des problèmes de vie privée, prennent naissance dans un comportement de l'utilisateur, mal ou peu guidé par des mauvaises interfaces utilisateur. La conception de ces interfaces est typiquement hors-sujet pour l'IETF, qui ne se préoccupe que de protocoles.

Un bon exemple est fourni par HTTP (RFC 7230). Quand on voit la variété des usages de HTTP aujourd'hui, de l'accès à des pages Web statiques à l'échange de données médicales, on se dit qu'une analyse des risques qu'il pose pour la vie privée, faite à l'époque de son développement, aurait été bien limitée.

Le RFC commence vraiment avec la section 3, sur la terminologie. Section bien nécessaire pour ce sujet, où le vocabulaire utilisé est souvent trop flou. Un RFC précédent sur la terminologie de la sécurité avait été le RFC 4949 mais il parlait peu de vie privée. Donc, quelques termes nouveaux (d'autres sont dans le RFC) :

— Assistant ("*enabler*") : une entité qui n'est pas dans le chemin direct de communication mais qui facilite la communication (et peut donc apprendre des choses). Par exemple, un serveur DNS faisant autorité va être un assistant pour une requête HTTP.
— Intermédiaire ("*intermediary*") : une entité située, elle, sur le chemin de communication, par exemple un relais SIP.
— Observateur ("*observer*") : une entité qui est techniquement en mesure d'observer le trafic, du fait de sa position dans le réseau. L'observateur est une entité légitime du réseau, dont l'existence est connue et au moins tacitement autorisée. Par exemple, un serveur de courrier électronique qui relaie les messages est un observateur pour les courriers. L'observateur est, par exemple, un intermédiaire ou un assistant (définitions précédentes) mais il faut se rappeler que les deux extrémités de la communication sont aussi des observateurs..
— Attaquant ("*attacker*") : toute entité qui va essayer de violer la vie privée, de manière non autorisée.
— Écoutant ("*eavesdropper*") : un cas particulier d'attaquant, purement passif (il ne fait qu'écouter). Contrairement à l'observateur, il n'est **pas** autorisé.
— Corrélation ("*correlation*") : combiner plusieurs informations pour en tirer une information plus complète.
— Empreinte numérique ("*fingerprint*") : un ensemble de caractéristiques d'une machine ou d'un programme qui permettent de l'identifier. Par exemple, le champ "*User-Agent :*", la liste de polices et quelques autres informations permettent de prendre l'empreinte du navigateur (cf. le Panopticlick <`http://panopticlick.eff.org/`>).
— Point intéressant (ou IOI pour "*Item Of Interest*") : toute information que l'attaquant peut avoir envie de connaître. Ainsi, le contenu de la communication est un point intéressant mais aussi, souvent, le seul fait qu'une communication ait lieu entre Alice et Bob est un point intéressant, même si on ne sait pas ce qu'ils se sont dit.
— Donnée personnelle ("*personal data*") : donnée qui permet d'identifier un individu physique. Ce concept est à la base de beaucoup de lois dans l'Union européenne comme la loi Informatique & Libertés en France. Contrairement à ce que croient beaucoup de gens, une donnée personnelle ne contient pas forcément le nom de l'individu. Par exemple, dans beaucoup de cas, une adresse IP est une donnée personnelle.

— Analyse de trafic ("*traffic analysis*") : déduire de l'information uniquement à partir de métadonnées sur les communications (qui appelle qui, quand, etc) sans avoir accès au contenu des communications. Par exemple, la simple augmentation du trafic radio d'une armée ennemie peut permettre de prévoir une action de sa part, même si le contenu des communications est chiffré et donc incompréhensible.
— Anonyme ("*anonymous*") : alors là, c'est un des termes les plus utilisés et les moins bien compris quand on parle de vie privée. On est anonyme si on ne peut pas être distingué des autres membres de l'**ensemble d'anonymat** ("*anonymity set*"). L'anonymat n'est jamais absolu, car on diffuse toujours des informations, l'important est que l'ensemble d'anonymat autour de ces informations soit suffisamment large.
— Identité ("*identity*") : un ensemble d'attributs qui identifient un individu. On a en général plusieurs identités, selon le contexte. (Voir l'excellent livre d'Olivier Iteanu <https://www.bortzmeyer.org/identite-numerique-en-question.html>.)
— Aujourd'hui, les informations d'identité sur l'Internet sont souvent stockés chez des **fournisseurs d'identité** spécialisés (si vous vous connectez à un service sur le Web avec votre compte Facebook, alors Facebook est votre fournisseur d'identité). Elles sont ensuite utilisées par des "*relying parties*" (le site Web où on se connecte), qui ont donc sous-traité la gestion de l'identité à un tiers.
— Nom officiel ("*official name*") : celui qui apparait sur les papiers d'identité étatiques. Il n'est en général pas unique. Notez que l'identité, définie plus haut, n'inclut pas forcément le nom officiel (mon identité sur un canal IRC donné peut être mon pseudo `canari95`). Notez aussi que l'anonymat n'est pas uniquement le fait de garder son nom officiel secret. Cela peut être aussi de vouloir empêcher le rapprochement de deux de ses identités.
— Nom personnel ("*personal name*") : nom par lequel on désigne un individu. Cela peut être le nom officiel mais pas toujours. Du point de vue technique, un logiciel ne sait pas en général s'il manipule des noms personnels ou des noms officiels.
— Pseudonyme ("*pseudonym*") : un nom qu'on va utiliser pour ne pas divulguer un nom personnel.

Par exemple, avec IP, le routeur est un intermédiaire, et peut être un observateur (s'il est équipé pour cela, disons que si le routeur est une machine Unix, il devient un observateur dès qu'il lance tcpdump). Quelqu'un qui s'est branché sur le câble sans autorisation est un écoutant.

À noter que l'analyse de sécurité du RFC 3552 supposait que les deux extrémités de la communication étaient sûres et que seuls les intermédiaires représentaient un danger. En matière de vie privée, ce n'est évidemment pas le cas (section 4 de notre RFC). Comme l'a montré l'affaire PRISM, ou comme le montrent les pratiques des gros silos commerciaux du Web 2.0 comme Facebook, le danger est souvent chez l'une des parties en train de communiquer, pas chez un tiers...

Et quelles sont exactement les menaces qui pèsent sur la vie privée ? La section 5 les détaille, suivant en partie le livre de Solove <http://docs.law.gwu.edu/facweb/dsolove/Understanding-Privacy/> et la recommandation du Conseil de l'Europe <https://wcd.coe.int/ViewDoc.jsp?Ref=CM/Rec%282010%2913>. D'abord, leurs conséquences. Les atteintes à la vie privée peuvent provoquer de la gêne mais aussi une perte de dignité, ou des pertes financières. Dans des cas extrêmes (violence conjugale, par exemple), ces atteintes peuvent mettre la vie des victimes en danger. D'autre part, même s'il n'y a pas eu accès à des informations privées, le seul fait d'être sous surveillance peut mettre très mal à l'aise, voir créer un sentiment d'angoisse. Il peut aussi entraîner un changement de comportement (on hésitera à commettre un acte légal, de peur des conséquences s'il était révélé) pouvant aller jusqu'à l'auto-censure. La vie privée est donc une affaire sérieuse, pas un simple détail.

Première attaque contre la vie privée envisagée, la surveillance des communications. Le RFC rappelle qu'elle n'est pas limitée au cas où l'attaquant a accès au contenu des communications. La seule analyse de trafic peut déjà en révéler beaucoup. Le chiffrement n'est pas une protection parfaite : le type de trafic reste visible, via des indicateurs comme la taille des paquets ou leur fréquence. Pour déjouer sérieusement la surveillance, il faudrait des protocoles avec des tailles de paquets variables, n'ayant pas de chaîne de bits prévisibles à un endroit fixe dans le paquet, etc. Pour sérieusement gêner l'analyse de

trafic, il faut des systèmes comme Tor. Parmi les entitées présentées plus haut, aussi bien les écoutants que les observateurs peuvent avoir une activité de surveillance.

Une fois les messages arrivés à bon port, tout n'est pas terminé. Les données stockées à destination peuvent faire l'objet d'une compromission, si le serveur qui les stocke n'est pas suffisamment protégé et peut être piraté. Sans compter le cas, traité plus loin, où le serveur de destination est lui-même un attaquant. Ceci dit, ces problèmes ne sont pas du ressort de l'IETF qui s'occupe normalement uniquement des réseaux.

Les risques ci-dessus sont une combinaison de risques pour la vie privée avec les risques plus généraux de sécurité. Mais il y a aussi des risques très spécifiques à la question de la vie privée. Par exemple, la corrélation, qui permet d'acquérir des informations normalement privées en reliant des données qui, séparément, semblaient inoffensives. Par exemple, si des identificateurs stables sont utilisés, les protocoles réseaux facilitent la corrélation. Une adresse IP stable sur le long terme présente un certain nombre d'avantages techniques mais est aussi un danger pour la vie privée (cf. RFC 4941). Autre exemple, le protocole TLS permet de reprendre une session cryptographique existante, pour gagner du temps et éviter de recommencer la négociation de tous les paramètres (cela peut se faire côté serveur, RFC 5246 ou côté client, RFC 5077). Mais comme cette reprise de session se fait avant la négociation de ces paramètres de cryptographie, elle a lieu en clair. Un écoutant peut donc voir que le second client TLS qu'il écoute est en fait le même que le premier, s'il utilise cette fonction.

Autre attaque contre la vie privée, celle passant par l'identification. Savoir que la machine `192.0.2.67` accède au site Web du gouvernement, c'est une chose. Pouvoir accéder à des informations sur la personne qui utilise cette machine, c'est bien mieux, pour l'attaquant. Cela peut se faire facilement avec les protocoles qui identifient une personne (une adresse SIP ou XMPP par exemple) mais aussi de manière indirecte même si le protocole permet l'anonymat. Prenons par exemple un site Web qui n'identifie pas ses utilisateurs. Aucun risque qu'un attaquant qui l'observe retrouve qui a accédé à la page `/page-sensible.html`, non ? Sauf que, si le même attaquant peut écouter le trafic d'un autre site Web qui authentifie ses utilisateurs, il peut faire une corrélation (via, par exemple, le champ `User-Agent:`) et ainsi identifier l'utilisateur.

Dans beaucoup de cas, l'utilisateur sait qu'on récolte des données personnelles sur lui. Si je me crée un compte sur amazon.com, il est clair qu'Amazon va savoir des choses sur moi, par exemple mon adresse postale pour me livrer les produits achetés. Il peut aussi y avoir des usages secondaires, des cas où le détenteur des données s'en sert pour autre chose que ce qu'il a annoncé. L'usage secondaire est techniquement indétectable et, comme cela se passe en dehors de la communication standardisée par l'IETF, notre RFC estime que ce problème, si grave qu'il soit en pratique, n'est pas de la responsabilité de l'IETF.

Encore plus sérieux, la révélation de données à des tiers. Comme il est trivial de copier des données numériques, il est techniquement impossible de savoir si les données personnelles qu'on a accepté de confier à l'entreprise X ne vont pas être copiées chez son partenaire Y. Là encore, c'est en dehors de la sphère IETF mais c'est quand même une des menaces les plus sérieuses. Le système PRISM est un exemple d'une telle copie, où les données qu'on avait accepté de confier à Google ou Facebook sont accessibles à la NSA. (Cet exemple n'est pas mentionné dans ce RFC qui, pour ne vexer personne et surtout pas les grosses entreprises états-uniennes qui contribuent beaucoup à l'IETF, passe rapidement sur ce risque, pourtant l'un des plus fréquents.)

Certains protocoles IETF permettent à l'utilisateur d'indiquer ses préférences en matière de protection de la vie privée, et notamment d'interdire la révélation de ces données à un tiers. C'est le cas par exemple du système décrit dans le RFC 6280. Évidemment, il n'existe aucun moyen technique de savoir si ces préférences sont respectées...

Notez aussi que la révélation peut être accidentelle : certains administrateurs systèmes sont assez négligents avec les données personnelles (ou bien ne se rendent pas compte qu'elles sont personnelles) et les laissent parfois accessibles par accident.

Dernière menace envisagée, l'exclusion. C'est le fait d'interdire à l'utilisateur de savoir ce qu'on sait sur lui. (La loi Informatique & Libertés appelle cela le droit d'accès aux données.) C'est une attaque tentante pour les gens qui veulent utiliser vos données personnelles. Cela limite le contrôle que l'individu peut faire.

Maintenant, assez déprimé sur les menaces, place aux solutions. La section 6 envisage les différentes méthodes connues pour limiter les dégâts. Améliorer la protection de la vie privée dans les protocoles réseau n'est pas facile car une protection sérieuse dépend de très nombreux facteurs, qui ne relèvent pas de la seule responsabilité des protocoles (ni même des logiciels). Je pense personnellement (et le RFC a tort de ne pas le rappeler) que le problème requiert avant tout des solutions politiques et juridiques. Mais ce RFC 6973 est un RFC, pas une loi, et se focalise donc surtout sur les aspects techniques : lorsqu'on est membre de l'IETF, que peut-on faire de concret aujourd'hui ?

Première chose évidemment, récolter **moins** de données ("*data minimization*" ou "*No data, no privacy problem*"). Les protocoles devraient être conçues de façon à ne transmettre que les données strictement nécessaires à l'accomplissemnt de la tâche. Les choix vont être délicats car les informations envoyées, même pas strictement nécessaires, peuvent avoir une utilité (le `User-Agent:` en HTTP, énorme risque pour la vie privée, mais si rigolo pour faire des statistiques). Le RFC se focalise donc surtout sur un point, l'identifiabilité. Il faut tout faire pour éviter qu'on identifie un utilisateur. Cela peut passer par l'absence d'identificateurs dans le protocole (ce que permet HTTP mais pas SMTP) ou par l'utilisation d'identificateurs de durée de vie limitée, choisis aléatoirement et changés régulièrement (les adresses IP temporaires du RFC 4941).

La meilleure protection de la vie privée est quand l'utilisateur peut rester anonyme. Cela implique qu'il existe un ensemble d'anonymat assez vaste pour qu'on soit réellement protégé. Un `User-Agent:` HTTP à lui seul vous place dans un ensemble d'anonymat qui peut être très réduit (j'ai vu une fois dans mes journaux un `User-Agent:` qui proclamait que le navigateur tournait sur OpenBSD, ce qui diminue considérablement le nombre de « suspects »). Par exemple, pour SIP, le RFC 3325 permet de choisir une adresse anonyme (notez que, comme souvent en matière d'anonymat, l'information sur votre identité peut venir d'ailleurs comme le champ `Via:`).

Proche de la notion d'anonymat, celle de pseudonymat. Dans ce cas, on a une identité stable et réutilisable, elle n'est simplement pas liée aux identités qu'on veut protéger (comme son nom officiel, par exemple). Dans l'exemple SIP ci-dessus, l'adresse utilisée est toujours `anonymous@anonymous.invalid` et est la même pour tout le monde. C'est de l'anonymat. Si par contre je me crée un compte `aucbbf51n` chez un fournisseur SIP `example.net` qui ne garde pas trace de mes informations personnelles, mon adresse `aucbbf51n@example.net` est un pseudonyme. Un pseudonyme a l'avantage, par rapport à l'anonymat, de permettre de construire une réputation.

Beaucoup d'identificateurs sur l'Internet vont être des pseudonymes puisque les protocoles Internet n'obligent pas (et heureusement, comparez avec ce que réclame l'UIT qui voudrait obliger à utiliser des « vraies » identités) à se servir d'un nom officiel. Ainsi, mon adresse de courrier pour ce blog est `stephane+blog@bortzmeyer.org` mais vous ne savez pas si c'est mon nom officiel ou un pseudonyme. Parfois, tout de même, des protocoles ou formats Internet transportent des identités qui peuvent être des noms officiels. Un exemple typique est le format vCard du RFC 6350.

Les pseudonymes ne sont pas parfaits, deux pseudonymes utilisés dans des contextes différents peuvent parfois être reliés par exemple via une information que vous communiquez. Si vous publiez

sur l'informatique avec une adresse professionnelle `monsieursérieux@example.com` et que vous tenez un blog sur la sexualité des hamsters avec l'adresse `lol-hamster218@example.net`, ne faites pas les mêmes fautes d'orthographe, n'utilisez pas les mêmes tournures de phrase dans les deux contextes ou bien vous serez démasqué. Un exemple de pseudonymes impossibles à lier est fourni par Romain Gary et Émile Ajar : étant donné la différence des styles, personne n'avait jamais suspecté que c'était le même écrivain.

Un problème supplémentaire survient fréquemment de nos jours : beaucoup de services sur l'Internet ne font plus la gestion de l'identité eux-mêmes mais la délèguent à un **fournisseur d'identité**, utilisant des techniques comme OpenID ou des protocoles privés. Selon le protocole utilisé, les risques pour la vie privée peuvent être plus ou moins grands. Par exemple, si le service voulant authentifier (RP pour "*relying party*", celui qui compte sur le fournisseur d'identité) communique directement avec le fournisseur d'identité (au lieu que tout passe via le client final), le fournisseur d'identité peut apprendre à quels services se connectent ses clients. Autre exemple, le fournisseur d'identité ne devrait pas envoyer au RP la totalité des informations dont il dispose sur un client.

Et, pour clore cette section 6 sur les solutions techniques améliorant la protection de la vie privée, le RFC note que le protocole doit fournir des mécanismes permettant à l'utilisateur de contrôler ses propres données (savoir ce que les autres savent sur lui, et pouvoir exprimer des préférences).

Plus pratique, la section 7 transforme ces bons conseils techniques en une liste de choses à vérifier lorsqu'on conçoit un nouveau protocole réseau. Une fois qu'on a développé un modèle du nouveau protocole (RFC 4101), lire cette liste et se poser les questions qu'elle contient devrait aider à limiter les risques pour la vie privée.

D'abord, minimiser la quantité de données distribuées. La meilleure protection des données est quand il n'y a pas de données. Quels sont les identificateurs utilisés par le protocole ? Est-ce qu'ils permettent la corrélation entre des sessions différentes (c'est souvent le cas s'ils sont stables sur le long terme) ? Ne pourrait-on pas limiter leur utilisation ? Leur mettre une durée de vie limitée ? Permettre aux utilisateurs d'en changer facilement ? Ensuite les données qui ne sont pas officiellement des identificateurs ? Qu'est ce qui est échangé entre les parties qui communiquent ? N'est-ce pas trop de données ? Ces données ne permettraient-elles pas de retrouver les identificateurs (pensez au Panopticlick <`http://panopticlick.eff.org/`>) ? À qui sont envoyés ces identificateurs et ces données (rappelez-vous qu'il y a typiquement plus que deux parties impliquées dans une communication) ? Par exemple, lors d'une connexion HTTP à `http://www.example.com/`, la requête DNS est envoyéee à des assistants (cf. le vocabulaire au début), les serveurs de la racine et ceux de `.com`. La racine apprend donc qu'on se connecte à `www.example.com`. Et le risque d'empreinte numérique ? Par exemple, si le protocole permet N opérations au début d'une connexion, sans spécifier leur ordre, un observateur peut apprendre quel logiciel vous utilisez en regardant l'ordre choisi. Et enfin est-ce que le protocole permet, voire impose, la conservation de données ou d'identificateurs sur le long terme, en dehors des sessions de communication ?

Ça, c'était pour minimiser les données. Pour le contrôle par l'utilisateur, les questions à se poser sont : quels sont les mécanismes dans le protocole pour contrôler **quelles** données on diffuse et à **qui** ? Là encore, il ne faut pas penser qu'au destinataire mais aussi à tous les intermédiaires possibles. Un exemple, HTTP n'impose pas des en-têtes indiscrets comme `User-Agent:` et `Referrer:` et cela permet à certains navigateurs ou relais de les supprimer.

Les attaques contre la vie privée ne seront pas menées que par les participants légitimes à une communication. On aura aussi des tiers qui essaieront d'accéder à une information qu'ils ne sont pas censés avoir. Le protocole doit donc prendre en compte la possibilité de surveillance. Qu'est-ce qui est prévu

contre l'écoute ? En général, c'est l'utilisation du chiffrement et le RFC contient donc une phrase « utilisez TLS si vous ne voulez pas être écouté ». Mais cela n'empêche pas l'analyse de trafic. Le protocole a-t-il des mécanismes qui facilitent ou au contraire compliquent cette analyse ? Par exemple, avec SSH, on voit facilement si on a affaire à un transfert de fichiers ou à une session interactive, par la quantité de données qui passe et leur cadencement. Lors d'un transfert de fichiers, on a beaucoup d'octets dans un sens et peu dans l'autre, les accusés de réception. Lors d'une connexion à un shell, on a peu d'octets de l'utilisateur vers le shell et une quantité moyenne (les réponses) en sens inverse. Le chiffrement ne dissimule pas cela. SSH a un mécanisme de remplissage de la communication avec des données bidon, pour brouiller les pistes mais je n'ai pas l'impression qu'OpenSSH l'utilise.

Toujours en sécurité classique, comment se comporte le protocole en cas de compromission des données stockées ? Ou contre une attaque active ?

Souvent, les préférences de protection de la vie privée sont réglables. Mais la plupart des utilisateurs ne modifieront jamais le réglage. Le RFC demande donc qu'on prête attention aux réglages par défaut. En l'absence d'intervention explicite, est-on protégé ou, au contraire, faut-il activer cette protection délibérement ? Par exemple, pour le cas des adresses IP temporaires, le RFC 4941 demandait qu'elles ne soient pas utilisées par défaut alors que le RFC 6724 a demandé le contraire. Écrit quatre ans après, à un moment où les préoccupations de protection de la vie privée sont devenues plus importantes, ce RFC 6724 avait des priorités différentes.

D'ailleurs, la fin de cette section 7 rappelle que, comme Schneier aime le répéter, la sécurité est **toujours** un compromis. Elle prend du temps, ralentit le réseau, complique les choses (pour les programeurs et pour les utilisateurs), bref a un coût. La protection de la vie privée n'échappe pas à cette nécessité de chercher un compromis.

Pour clore ce RFC, la section 8 fournit un exemple complet d'analyse de sécurité d'un service donné. Les auteurs ont choisi la difficulté, car cette analyse porte sur un service particulièrement indiscret et intrusif, le service de **présence** (faire connaître à l'extérieur si on est disponible ou pas, par exemple pour être contacté par messagerie instantanée). Ce service permet d'illustrer toute la palette des questions soulevées par la protection de la vie privée. Présentée dans le RFC 2778, la présence (« je suis dispo », « je déjeune », « je suis en réunion ») est un service très dynamique, nécessitant de nombreuses mises à jour. Ses implications pour la vie privée sont évidentes et, dès le début, l'IETF avait prévu un contrôle des utilisateurs sur cette information (RFC 3859). Les protocoles qui utilisent un système de présence comme SIMPLE (RFC 6914) ou XMPP (RFC 3922) s'appuient sur ce contrôle.

Voici la gestion de la présence par l'utilisateur dans le client de messagerie instantanée Pidgin :

Dans l'architecture standard, il y a un tiers, un assistant, le serveur de présence, qui reçoit les informations des clients et les transmet aux lecteurs autorisés. Ce tiers est nécessaire pour les cas où la/les machine(s) du client soient toutes éteintes ou injoignables. Il sert également à agréger de l'information envoyée par les différentes machines de l'utilisateur (son ordinateur peut être éteint mais l'utilisateur être toujours joignable via son "*smartphone*"). Mais il complique évidemment beaucoup la sécurité puisqu'il va être au courant de beaucoup de choses. Bien sûr, le serveur ne distribue l'information qu'aux lecteurs autorisés. Mais il n'y a pas de protection technique : il faut faire une confiance totale au serveur, confiance en son honnêteté et ses bonnes pratiques de sécurité. On pirate un serveur de présence et on peut suivre les gens à la trace. Le serveur oublie de rendre TLS obligatoire et les écoutants peuvent tout apprendre (et le client du serveur de présence n'a aucun moyen de vérifier si TLS est activé ou pas). Encore pire, c'est le serveur de présence qui authentifie les lecteurs, pas le client final. L'anonymat n'est ici pas une solution puisque le but d'un service de présence est justement d'obtenir des informations sur des personnes identifiées (« quelqu'un est allé déjeuner » n'est pas une information intéressante).

Pour compléter le tableau, d'autres informations que la présence sont souvent transmises, par exemple les capacités du logiciel utilisé (accepte t-il les appels vidéo, par exemple). Elles peuvent permettre d'identifier l'appareil utilisé. En outre, les extensions ultérieures au service de présence ont permis de publier également la localisation physique. Un groupe de travail a été créé pour travailler sur la protection de la vie privée dans le cas où on distribue cette information, le groupe GEOPRIV <http://tools.ietf.org/wg/geopriv>, dont le premier RFC fut le RFC 4079 (mais il y en a d'autres </search?pattern=geopriv>).

Avec tellement d'information, l'analyse de sécurité du service de présence doit évidemment commencer par se demander si le jeu en vaut la chandelle. Déjà, on pourrait utiliser une autre architecture, plus pair à pair, où les utilisateurs se préviennent directement de leur présence. Il existe un format standard pour transporter cette information, PIDF ("*Presence Information Data Format*"), qu'on peut chiffrer pour empêcher toute écoute. Mais cette solution ne semble pas réaliste aux auteurs du RFC : elle ne fonctionne pas si la machine de publication est éteinte ou injoignable, chaque machine qui publie doit connaître les clés publiques de tous les abonnés, et enfin elle suscite un trafic important (la présence est très dynamique, avec des changements fréquents). Une variante est l'architecture où le serveur de présence ne sert que de redirecteur : il ne connait pas l'information, mais sait rediriger vers la machine actuelle de chaque personne qui publie de l'information de présence. Le serveur peut ainsi prévenir qu'une personne est injoignable et rediriger vers cette personne dans le cas contraire. Mais cette architecture, comme la précédente, a ses propres problèmes de vie privée, puisqu'elle permet à chaque participant d'apprendre les adresses IP des autres (normalement connues du seul serveur de présence). En outre, ces deux solutions, davantage pair à pair, souffrent du difficile problème de la connexion d'une machine à une autre, lorsqu'une des deux, ou les deux, sont derrière un pare-feu ou un routeur NAT. Avec le serveur de présence, il suffit aux clients d'une connexion sortante, ce qui est bien plus souvent possible.

Bref, cette intermédiaire dangereux qu'est le serveur de présence va être difficile à éliminer. D'où l'approche actuelle, qui repose plutôt sur des préférences quant à la divulgation de l'information, préférences (RFC 4745, RFC 5025 et RFC 6772) qui sont envoyées au serveur de présence, et à qui il faut faire confiance pour les respecter. Le RFC note bien que le succès est très limité : peu de logiciels exploitent cette information. Au moins, on peut documenter leurs limites, ce qui est fait ici.

On a donc là un bel exemple d'un compromis entre la protection de la vie privée (qui justifierait qu'on ne publie jamais rien sur sa présence) et le désir de faciliter la vie des utilisateurs en indiquant à l'avance si un appel a des chances d'aboutir ou pas.

Mon commentaire à une version préliminaire de ce RFC est disponible en ligne <http://trac.tools.ietf.org/group/iab/trac/ticket/258> (notez la note de cloture du ticket). Globalement, mon regret est que ce document (même après les changements) est très axé sur les risques dus aux tierces parties et parle peu des risques du méchant silo qui stocke les données personnelles, ou sur les risques étatiques type PRISM.

Si vous voulez approfondir ces questions, ce RFC cite trois sources : le « "*FAIR INFORMATION PRACTICES : A Basic History*" <http://bobgellman.com/rg-docs/rg-FIPShistory.pdf> » de Gellman, les directives de l'OCDE <http://www.oecd.org/internet/ieconomy/oecdguidelinesonthepro htm> et le « "*Privacy Indexes : A Survey of Westin[Caractère Unicode non montré [2]]s Studies*" <http://reports-archive.adm.cs.cmu.edu/anon/isri2005/CMU-ISRI-05-138.pdf> ».

2. Car trop difficile à faire afficher par LaTeX