RFC 7435 : Opportunistic Security: Some Protection Most of the Time

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 3 janvier 2015

Date de publication du RFC : Décembre 2014

http://www.bortzmeyer.org/7435.html

Les révélations de Snowden ont sérieusement relancé l'intérêt pour la sécurité informatique, et notamment sur les risques d'espionnage. Mais, en sécurité, le mieux est parfois l'ennemi du bien. Certaines exigences de sécurité peuvent mener à des solutions techniquement parfaites mais tellement compliquées à utiliser qu'elles ne seront que peu ou pas déployées. C'est par exemple le cas de l'authentification pour utiliser le chiffrement. Si on impose l'authentification forte du pair avant de chiffrer, on risque de ne pas pouvoir chiffrer, dans des cas où ce serait pourtant bien utile. Ce nouveau RFC, par le mainteneur de Postfix, définit un concept utile, cela de **sécurité opportuniste** ("*opportunistic security*") : on chiffre quand on peut, même sans authentification, et on authentifie si c'est possible. L'idée est d'augmenter sérieusement la part de trafic chiffré dans l'Internet.

Un petit détour technique, avant de commencer le RFC. Pourquoi chiffrer sans authentifier est-il dangereux ? Si on ne veut que la confidentialité, mais qu'on se moque de l'identité du pair avec qui on communique, pourquoi s'embêter à authentifier, surtout lorsqu'on sait que chiffrer est si simple et authentifier si compliqué ? C'est parce que chiffrer sans authentifier vous rend vulnérable à l'Homme du Milieu. Celui-ci, tapi dans le réseau, entre Alice et Bob, va recevoir les messages d'Alice et les transmettre à Bob (et réciproquement) et cela, évidemment, après lecture. Chiffrer ne sert à rien s'il y a un homme du milieu : Alice croit chiffrer pour Bob alors qu'elle chiffre en fait pour l'espion qui pourra donc lire le texte en clair avant de le chiffrer et de le passer à Bob.

À noter que l'homme du milieu est forcément **actif** : il doit maintenir un canal de communication avec Alice et un avec Bob. Parfois, pour des raisons pratiques ou bien juridiques (un attaquant actif peut relever de lois pénales plus sévères), l'attaquant reste **passif** et, dans ce cas, le chiffrement seul, sans authentification, protège parfaitement. Attention, toutefois : pas mal de personnes semblent croire que seul le FAI ou bien la NSA peuvent faire des attaques actives : or, celles-ci sont plus faciles que ne le croit M. Toutlemonde (usurpation ARP par exemple).

La solution correcte aux attaques de l'homme du milieu est l'authentification : si Alice vérifie qu'elle parle bien à Bob, et uniquement à lui, alors, elle pourra détecter quand un homme du milieu sera présent, et éviter ainsi de lui passer ses secrets. C'est le mode « tout ou rien » (soit on authentifie, soit on avorte la communication) qui était traditionnel dans les protocoles IETF (à part quelques exceptions comme la sécurité « mieux que rien » du RFC 5386 [1]). Mais l'authentification n'est pas facile. Prenons un exemple, celui de HTTPS, qui utilise la plupart du temps des certificats X.509 pour l'authentification. Des tas de choses peuvent aller mal : certificats expirés <http://www.bortzmeyer.org/tester-expiration-certifs.html>, signés par une autorité inconnue de ce navigateur <http://www.bortzmeyer.org/cacert.html>, auto-signés pour éviter de payer des AC à l'utilité douteuse, et tous les autres malheurs qui ponctuent la vie de l'internaute qui essaie de se connecter en HTTPS. L'utilisateur se rabat donc souvent sur du simple HTTP non chiffré. De peur de l'homme du milieu, il fait tout passer en clair ! Cet état de chose est évidemment absurde. Lorsqu'on a un des fameux avertissements de sécurité « ce certificat a un problème incompréhensible pour vous, qu'est-ce que vous décidez ? », c'est dans la grande majorité des cas une erreur du serveur et pas une attaque de l'homme du milieu. Il vaudrait donc mieux, lorsque le site est également accessible en HTTP ordinaire, chiffrer sans authentification : cela ne peut jamais être pire que de ne pas chiffrer du tout. (Les sites HTTPS à plus haut niveau de sécurité, comme une place de marché Bitcoin <http://www.bortzmeyer.org/bitcoin-marches.html>, imposent HTTPS, et, en cas d'erreur, il n'y a qu'à pleurer en attendant la réparation.)

Ce débat sur l'authentification obligatoire est aussi ancien que la cryptographie. Je me souviens des vieux dinosaures de la sécurité regardant SSH avec méfiance, lors de son introduction, car l'authentification n'était pas obligatoire. Mais les solutions que prônaient ces sympathiques tétrapodes étaient tellement complexes à déployer que les gens ne chiffraient pas, et utilisaient telnet. Il est donc clair que SSH, même avec son modèle TOFU ("*Trust On First Use*") d'authentification plutôt faible, a amélioré la sécurité de l'Internet. Le TOFU n'est pas parfait (il est vulnérable si l'homme du milieu est présent dès la première connexion, et il ne permet pas de savoir si un changement de clé est normal ou pas) mais il a permis de remplacer massivement l'ancien telnet par un protocole chiffré.

Outre HTTPS et SSH, plusieurs protocoles utilisent du chiffrement, et ont des solutions plus ou moins bonnes pour l'authentification. Des solutions comme DANE (RFC 6698) ont été proposées pour résoudre une partie des problèmes d'authentification mais, à l'heure actuelle, il est clair que l'Internet n'a pas de solution d'authentification généraliste techniquement correcte, politiquement sûre, et effectivement déployée.

Ce débat a mené au terme de **sécurité opportuniste** ("*opportunistic security*"), qui date de bien avant le RFC. Mais il n'était pas défini rigoureusement. Parfois, il désignait le fait de faire de la crypto sans que l'utilisateur l'ait explicitement demandé (ce que fait HTTPS Everywhere en transformant d'autorité des sessions HTTP en HTTPS), parfois il désignait le chiffrement sans authentification (c'est le sens qu'il a dans le RFC 5386), parfois la capacité de basculer automatiquement en mode non-chiffré (la définition du Wikipédia anglophone) et parfois enfin du chiffrement avec authentification mais sans avoir de configuration spécifique par pair (c'est ainsi que le RFC 4322 utilise le terme). Notre RFC lui donne une définition stable, qui sera utilisée dans les documents ultérieurs de l'IETF, comme le futur HTTP 2 ou comme DANE pour SMTP.

Bref, on change de perspective : d'un modèle où on authentifie systématiquement, protégeant contre les attaques passive **et** actives (et où l'absence d'authentification n'est citée que comme « mode dégradé », voire coupe la communication complètement), on passe à un modèle plus réaliste où l'état de base est l'absence de toute protection (la réalité de l'Internet d'aujourd'hui) et où tout gain en sécurité, même

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5386.txt

imparfait, est une amélioration par rapport à cet état de base. Avec la **sécurité opportuniste**, on chiffre quand c'est possible et on authentifie quand on peut. « Opportuniste » est donc un terme positif dans ce RFC. Il repose sur l'idée d'un « plancher de sécurité » en dessous duquel on ne descendra pas, et du « au moins aussi sûr » où on déploie tout système (comme le chiffrement) qui va améliorer les choses, ou en tout cas ne jamais les dégrader. Ce nouveau modèle ne change rien au cas où l'on imposait une sécurité minimale (par exemple avec HSTS, RFC 6797), il concerne les communications qui se font actuellement en clair.

La section 3 de notre RFC définit ce qu'est la sécurité opportuniste. Elle a (en simplifiant) **trois niveaux**, le niveau de base, où tout est en clair (c'est la majorité du trafic Internet aujourd'hui), le niveau où on chiffre, mais sans authentifier, et le niveau le plus sûr, où on chiffre après authentification. Ce dernier niveau est le meilleur mais la devise de la sécurité opportuniste est « mieux vaut le deuxième niveau que de rester bêtement sans sécurité du tout ». (Ce modèle est simplifié car il peut y avoir plusieurs niveaux de chiffrement ; par exemple, on va essayer de chiffrer en mode PFS mais, si cela échoue, on chiffre quand même, dans un mode moins sûr.) Ce modèle de sécurité opportuniste repose sur quatre principes :

— Les politiques de sécurité existantes continuent comme avant. Si vous avez mis `StrictHostKeyChecking yes` dans la configuration d'OpenSSH, la sécurité opportuniste ne vous concerne pas, OpenSSH imposera une authentification de toutes les machines, même la première fois que vous vous connectez.
— Priorité à la communication. En cas de problème d'authentification, on considère qu'il vaut mieux continuer de manière non sûre plutôt que de renoncer (c'est de fait la politique de nombreux utilisateurs <http://www.bortzmeyer.org/rational-security.html>).
— S'adapter au pair et donc utiliser, pour chaque couple de machines qui communiquent, la sécurité maximale possible pour ce couple. Si un pair ne parle que DES, c'est nul, mais on l'utilise quand même, sans pour autant descendre le niveau de sécurité de la communication avec les autres pairs. (Il existe plusieurs méthodes pour déterminer les capacités du pair, certaines dans la session elle-même, comme la négociation des algorithmes de chiffrement dans TLS, d'autres externes, comme DANE.)
— Franchise. Dans les messages aux utilisateurs, ou dans les journaux, on ne prétend pas qu'on a obtenu un niveau de sécurité qui ne correspond pas à la réalité. Si on n'a pas pu authentifier, on le dit (cela va poser d'intéressants problèmes d'interface utilisateur...)

La section 4 du RFC illustre le concept de sécurité opportuniste avec SMTP (domaine de compétence principal de l'auteur). Une extension à SMTP, `STARTTLS`, normalisée dans le RFC 3207, permet d'utiliser TLS pour chiffrer la session. Elle est aujourd'hui largement déployée, par exemple par Facebook <https://www.facebook.com/notes/protect-the-graph/the-current-state-of-smtp-starttls-depl1453015901605223> ou par Google <https://www.google.com/transparencyreport/saferemail/>. À noter que cette extension n'est pas protégée cryptographiquement, et donc vulnérable aux attaques par repli (*"downgrade attacks"*) où l'attaquant actif va essayer de faire croire qu'un des pairs ne gère pas la sécurité maximale, menant à l'emploi de solutions de sécurité inférieures (ces attaques, un des points faibles de la sécurité opportuniste, sont traitées également dans la section 6). `STARTTLS` ne protège donc guère contre un attaquant actif. Donc, utiliser le texte en clair lorsqu'il y a un problème TLS (mauvais certificat, par exemple, ce qui est très fréquent avec SMTP sur TLS) n'a guère de sens : de toute manière, un attaquant actif pourrait supprimer le `STARTTLS`. Bref, SMTP ne devrait pas se rabattre sur le texte en clair alors que du chiffrement sans authentification est possible.

Ne manquez pas également de lire la section 6, qui résume le problème de sécurité, et qui note que la sécurité opportuniste peut aussi aider contre des attaquants ayant de grands moyens, genre NSA (RFC 7258), en les obligeant à utiliser des attaques actives (comme les attaques QUANTUM).