RFC 4366 : Transport Layer Security (TLS) Extensions

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 19 mars 2009

Date de publication du RFC : Avril 2006

https://www.bortzmeyer.org/4366.html

Ce RFC décrivait les premières extensions au protocole cryptographique TLS. Il décrit le mécanisme général d'extension (qui a depuis été déplacé dans le RFC 5246 [1]) puis certaines extensions spécifiques (résumées en section 1). Il est désormais remplacé par le RFC 6066.

Ces extensions ne changent pas le protocole : un client ou un serveur TLS qui ne les comprend pas pourra toujours interagir avec ceux qui les comprennent. Le serveur doit en effet ignorer les informations contenues dans le `Hello Client` qu'il ne connait pas (section 7.4.1.2 du RFC 5246).

La section 2 décrit le mécanisme général des extensions. Elle a été remplacée par le RFC 5246, notamment la section 7 de ce dernier (voir la structure en 7.4.1.2). En gros, le principe est d'ajouter des données à la fin du paquet `Hello`, qui marque le début de la négociation TLS. Dans le langage de spécification propre à TLS, la liste des extensions possibles est un `enum` (section 2.3) mais elle est désormais en ligne à l'IANA <https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml>.

Les premières extensions normalisées sont décrites dans la section 3. J'en cite seulement certaines. Leur description officielle est désormais dans le nouveau RFC 6066.

SNI ("*Server Name Indication*", section 3.1) permet au client TLS d'indiquer au serveur le nom sous lequel il l'a trouvé. Cela autorise le serveur à présenter des certificats différents selon le nom sous lequel on y accède, fournissant ainsi une solution au problème récurrent de l'authentification avec TLS lorsqu'un serveur a plusieurs noms <https://www.bortzmeyer.org/auth-x509-plusieurs-noms.html>. À noter que SNI fonctionne avec les IDN, le RFC décrivant le mécanisme à suivre.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc5246.txt`

"*Maximum Fragment Length*", section 3.2, permet au client d'indiquer qu'il souhaiterait des fragments de données de taille plus réduite. Cela peut être utile pour un client contraint en mémoire ou en capacité réseau, par exemple un téléphone portable. Certains de ces clients contraints apprécieront également "*Truncated HMAC*" (section 3.5) qui autorise à réduire la taille du MAC utilisé pour protéger l'intégrité des paquets.

La section 3.6 décrit ("*Certificate Status Request*"), une extension qui permet d'indiquer la volonté d'utiliser OSCP, un protocole d'interrogation de certificats X.509 (plus léger que l'habituelle liste de certificats révoqués).

Qui dit nouvelles extensions, dit nouvelles erreurs. La section 4 est donc consacrée aux nouveaux codes d'erreur, pour toutes les extensions décrites. La plus évidente étant `unsupported_extension` où le client TLS reçoit une extension qu'il n'avait pas demandé (mise dans son `Hello`).

Enfin, la section 5 décrit la procédure pour l'enregistrement des nouvelles extensions mais, comme elle a été assouplie par le RFC 5246, il vaut mieux consulter la section 12 de ce dernier.

Notre RFC annule le RFC 3646 (le principal changement est un léger assouplissment des règles d'enregistrement de nouvelles extensions) et est lui-même remplacé par les RFC 5246 et RFC 6066.