RFC 7590 : Use of Transport Layer Security (TLS) in the Extensible Messaging and Presence Protocol (XMPP)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 juin 2015

Date de publication du RFC : Juin 2015

https://www.bortzmeyer.org/7590.html

Le groupe de travail UTA <https://tools.ietf.org/wg/uta> de l'IETF produit des recommandations pour un usage correct de TLS par les applications. En effet, indépendamment des forces et faiblesses propres de TLS, plusieurs problèmes de sécurité sont survenues en raison d'une mauvaise utilisation. Ce nouveau RFC traite le cas spécifique de l'utilisation de TLS par le protocole XMPP.

Les problèmes généraux identifiés par le groupe UTA avaient été documentés dans le RFC 7457[1] et les solutions généralistes, applicables à toutes les applications, dans le RFC 7525. Et pour le protocole XMPP, normalisé dans le RFC 6120 ? XMPP utilise TLS depuis au moins 1999. Les sections 5, 9 et 13 du RFC 6120 expliquent déjà comment faire du TLS avec XMPP. Mais notre nouveau RFC va plus loin et, dans l'esprit du manifeste XMPP/TLS <http://lwn.net/Articles/599647/> , décide que XMPP doit suivre les recommandations plus strictes du RFC 7525, notamment concernant le choix des algorithmes de chiffrement.

La section 3 du RFC répète les recommandations du RFC 7525, que doivent désormais suivre les mises en œuvre de XMPP, notamment :

— Toute mise en œuvre de XMPP doit avoir l'option <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls' qui indique qu'elle est prête à démarrer TLS (RFC 6120, section 5.4.1 et quelques autres). Mais, comme cette option n'est pas elle-même protégée par TLS, elle peut être supprimée par un homme du milieu (attaque dite de "*stripping*", RFC 7457, section 2.1). XMPP doit donc tenter de faire du TLS avec son partenaire, que cette option soit présente ou pas (le manifeste cité plus haut impose TLS, de toute façon).
— La compression TLS étant désormais rejetée (RFC 7525, section 3.3), XMPP peut se rabattre sur la compression XMPP du XEP-0138 <http://xmpp.org/extensions/xep-0138.html>.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7457.txt

— XMPP a un mécanisme de reprise rapide des sessions (XEP-0198 <http://xmpp.org/extensions/xep-0198.html>), on peut encore l'améliorer en le couplant avec la reprise de sessions de TLS.
— En théorie (RFC 6125), un client XMPP devrait authentifier le serveur (et, de préférence, le serveur authentifier les autres serveurs). En pratique, ce n'est pas toujours le cas (avec Pidgin, un certificat à problèmes est signalé mais un seul clic suffit à l'accepter). Une des raisons pour lesquelles on ne peut pas imposer immédiatement une authentification généralisée est que les serveurs XMPP sont souvent hébergés dans un environnement multi-clients, chaque client hébergé ayant son propre nom de domaine et qu'un tel serveur devrait donc avoir un certificat pour chaque client, ou un certificat couvrant tous les clients. DANE résoudra peut-être le problème. En attendant, le RFC recommande fortement de préférer une connexion chiffrée et non authentifiée à une connexion en clair (se replier sur du trafic en clair parce que le certificat est invalide est absurde, mais courant). C'est par exemple ce que recommande le RFC 5386 pour le cas d'IPsec. Bref, ≪ TLS tout le temps, authentification si possible ≫ est le principe.
— SNI ("*Server Name Indication*", RFC 6066, section 3) est inutile en XMPP, l'attribut `to` suffit à indiquer le domaine concerné (l'attribut `to` envoyé avant TLS n'indique que le domaine, pas le destinataire).
— En sécurité, il est bien connu que le point faible est l'utilisateur humain. La section 3.6 fait donc des recommandations pour les développeurs d'interface utilisateur : indiquer si la connexion avec le serveur est chiffrée par TLS, indiquer si l'authentification a réussi et, si oui, comment, permettre d'afficher l'algorithme de chiffrement utilisé et le certificat présenté, être averti si un certificat change (ce qu'on nomme parfois l'épinglage - "*pinning*"). Je viens de tester avec un Pidgin 2.10.10 et la seule de ces recommandations qui semble mise en œuvre est la possibilité d'afficher les certificats (menu "*Tools -¿ Certificates*").

La section 5 rappelle que XMPP sur TLS chiffre aussi les informations de routage (contrairement à SMTP) et limite donc les fuites de métadonnées. Elle revient aussi sur quelques limites de TLS : XMPP passe par plusieurs serveurs et, si ceux-ci sont piratés ou indiscrets, le chiffrement TLS, qui n'est pas de bout en bout, ne protège pas contre ceux qui ont le contrôle des serveurs. En termes moins gentils, si vous utilisez Google Talk, le chiffrement TLS avec Google Talk (qui marche bien) ne vous protège pas de PRISM. (À noter que notre RFC ne cite pas la solution de bout-en-bout OTR, qui existe mais est mal intégrée à XMPP.)

Je n'ai pas trouvé de liste de toutes les implémentations XMPP, avec leur degré de conformité à ce RFC.