RFC 3207 : SMTP Service Extension for Secure SMTP over Transport Layer Security

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 novembre 2009

Date de publication du RFC : Février 2002

http://www.bortzmeyer.org/3207.html

Le courrier électronique, on le sait, n'offre guère de sécurité, que cela soit contre la lecture illégitime des messages ou bien contre l'injection de faux messages. Vue la complexité de son architecture (cf. RFC 5598 [1]), fruit d'une histoire longue et agitée, il ne faut pas espérer trouver **la** solution parfaite à tous les problèmes de sécurité du courrier. Aussi, ce RFC 3207 se contente d'apporter juste une brique à l'édifice, l'utilisation de TLS pour protéger le canal de communication entre deux MTA.

La communication normale entre deux de ces serveurs de messagerie se fait, avec le protocole SMTP, en clair. Aucune protection contre l'écoute ou la modification de données (section 1). Il n'y a pas non plus de moyen d'être sûr de l'authenticité du serveur en face. TLS, normalisé aujourd'hui dans le RFC 5246, fournit ces services de confidentialité (via le chiffrement) et d'authentification (via les certificats). Il faut noter que TLS ne sécurise ici que le **canal** entre deux serveurs, pas le **message** qui est transporté. Un message passe typiquement par plusieurs serveurs et toutes les communications ne seront pas forcément protégées par TLS. Même si elles le sont, si un des serveurs intermédiaires est malhonnête, TLS ne protégera pas (voir la section 6 du RFC qui détaille ce point de manière très complète). Au contraire, les solutions de sécurité **de bout en bout** comme PGP protégeront le message (mais sont peut-être plus difficiles à déployer).

Donc, le principe de ce RFC est de faire tourner SMTP au dessus de TLS. Cela se fait par le biais d'une nouvelle extension SMTP, `STARTTLS`, décrite en section 2 (contrairement à HTTP, il n'y a pas de port spécial où on utiliserait toujours TLS).

Cette extension est composée d'une option qu'annonce le serveur distant, `STARTTLS` :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc5598.txt`

```
220 mail.generic-nic.net ESMTP Postfix (Debian/GNU)
EHLO chezmoi.example.net
250-mail.generic-nic.net
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
...
```

et d'une nouvelle commande, également nommée `STARTTLS`, décrite en section 4 :

```
STARTTLS
220 2.0.0 Ready to start TLS
...
```

À cette commande, le serveur distant peut parfois répondre 454 (*"TLS not available due to temporary reason"*) auquel cas l'initiateur de la connexion devra décider s'il continue sans TLS (et est donc vulnérable à une **attaque par repli**, où un attaquant actif essaie de faire croire que la sécurisation n'est pas possible, voir la section 6 du RFC) ou de renoncer. Voir par exemple l'option `smtp_tls_security_level = encrypt"` de Postfix. Le RFC suggère (section 6) d'enregistrer le fait qu'un pair donné utilise TLS et de refuser les connexions ultérieures si TLS a été abandonné mais je ne connais aucune mise en œuvre de SMTP sur TLS qui fasse cela.

Pour que le courrier puisse continuer à fonctionner dans l'Internet, le RFC impose qu'un serveur annoncé publiquement (par exemple, placé en partie droite d'un enregistrement MX) accepte les connexions sans TLS. En revanche, l'option TLS est particulièrement intéressante (section 4.3) si l'utilisateur humain est authentifié et donc si on utilise le port de soumission (RFC 4409).

Si la commande `STARTTLS` est un succès (code 220), la négociation TLS habituelle commence. Les deux parties examinent ensuite ses résultats (par exemple l'algorithme de cryptographie négocié, ou bien le certificat du pair distant) et décident de continuer ou pas (section 4.1).

Parlant du certificat, faut-il vérifier celui du voisin ? En pratique, quasiment personne ne le fait, en raison du prix des certificats et de la difficulté à les obtenir et les gérer. De nos jours, SMTP sur TLS procure donc la confidentialité par rapport à des écoutants passifs mais aucune authentification. À la rigueur, on peut toujours regarder les en-têtes du message pour avoir quelques informations supplémentaires :

```
Received: from mail.bortzmeyer.org (unknown [IPv6:2001:4b98:41::d946:bee8:232])
        (using TLSv1 with cipher DHE-RSA-AES256-SHA (256/256 bits))
        (Client CN "mail.bortzmeyer.org", Issuer "ca.bortzmeyer.org" (not verified))
        by mx1.nic.fr (Postfix) with ESMTP id 964A71714088
        for <echo@nic.fr>; Thu, 26 Nov 2009 12:08:08 +0100 (CET)
```

Si le MTA vérifie l'authenticité de son partenaire, le RFC lui laisse le choix des détails, recommandant seulement d'accepter un serveur situé dans le même domaine que celui qu'on cherche à contacter.

Une fois TLS négocié avec auccès, les machines doivent ignorer tout ce qui a été tapé avant et reprendre la négociation SMTP (`EHLO` et la suite), puisque l'information obtenue sans TLS n'est pas fiable (section 4.2).

Ce RFC est une mise à jour de la première norme, le RFC 2487, en bien plus détaillé, notamment sur les attaques possibles (une annexe résume les changements).

La plupart des MTA aujourd'hui ont la capacité de faire du TLS, comme décrit pour Postfix dans mon article <http://www.bortzmeyer.org/postfix-tls.html>. Parmi les auto-répondeurs <http://www.bortzmeyer.org/repondeurs-courrier-test.html> de test, certains acceptent TLS.