

# Configurer Postfix pour authentifier avec un mot de passe

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 juillet 2006. Dernière mise à jour le 17 février 2008

<https://www.bortzmeyer.org/postfix-sasl.html>

---

Autrefois, tout était simple, tout serveur SMTP relayait le courrier de n'importe qui vers n'importe qui. Cette possibilité ayant été largement abusée par les spammeurs, les serveurs bien gérés n'acceptent désormais de relayer que pour leur réseau local. Mais cela laisse ouvert le problème des utilisateurs distants. Une des solutions est d'authentifier leurs connexions avec un mot de passe.

Le RFC 6409<sup>1</sup> fait obligation aux serveurs SMTP d'authentifier les soumissions de courrier (section 4.3. "Require Authentication"). Il existe plusieurs méthodes pour cela :

- TLS, qui nécessite un certificat sur chaque client.
- SASL qui va utiliser un mot de passe qui peut être transmis en clair (il faudra alors chiffrer la communication, par exemple en utilisant TLS, cette fois pour la confidentialité) ou bien utilisé dans un système de défi/réponse.

Le choix va dépendre de :

- Si vous avez envie de gérer une PKI pour authentifier avec TLS.
- Du type de clients humains que vous avez (mettre en place une solution de sécurité va vous faire passer du temps au téléphone, avec les utilisateurs <<http://dwu.mu.org/>>).
- Des logiciels qu'utilisent vos clients.
- Des bases de données d'utilisateurs que vous avez (si vous comptez les réutiliser avec SASL).

Par exemple, je détaille ici la configuration du MTA Postfix sur une Debian. Un bon HOWTO est "*Postfix/SASL/TLS HowTo for Debian*" <<http://www.tribulaciones.org/docs/postfix-sasl-tls-howto.html>>. (Pour TLS seul, un autre article <<https://www.bortzmeyer.org/postfix-tls.html>> décrit sa configuration pour Postfix.)

D'abord, je sépare le service de réception normal du courrier (sur le port 25) et le service de soumission de courrier par les utilisateurs authentifiés (cette séparation est dans la logique du RFC 6409). Dans `master.cf`, je mets :

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc6409.txt>

---

```
smtp      inet n       -       -       -       smtpd
# smtpd_sasl_authenticated_header appeared only in Postfix 2.3
submission inet n       -       -       -       smtpd      -o smtpd_etrn_restrictions=reject -o smtpd
```

Ensuite, je dis à Postfix d'autoriser le relai de courrier pour les utilisateurs authentifiés en SASL. Dans `main.cf`, je mets :

```
smtpd_recipient_restrictions = permit_sasl_authenticated, ...

# SASL options: only secure methods, without plain text
smtpd_sasl_security_options = noanonymous, noplaintext
smtpd_sasl_local_domain = $myhostname
```

Avant de recharger Postfix (`postfix reload`), j'installe les logiciels nécessaires :

```
% sudo aptitude install postfix-tls libsasl2-modules sasl2-bin
```

Il reste à indiquer à SASL où trouver la base des utilisateurs. Dans `sasl/smtpd.conf` :

```
pwcheck_method: auxprop
mech_list: digest-md5 cram-md5
```

lui dit d'utiliser sa propre base (`/etc/sasl2`) et de ne publier que les méthodes d'authentification à base de défi/réponse. On remplit ensuite la base :

```
sudo saslpasswd2 -u mail.bortzmeyer.org -c stephane
sudo sasldblistusers2 # Pour vérifier
```

Faites **attention** au domaine donné par l'option `-u` : il doit coïncider avec la variable `myhostname` de Postfix. Sinon, on obtiendra des messages du genre *"no secret in database"*. **Attention** également au `chroot` : comme le paquetage Debian de Postfix utilise `chroot`, il faudra copier la base des utilisateurs dans la `chroot` (`sudo cp /etc/sasl2 /var/spool/postfix/etc`, cela n'est pas fait automatiquement (sauf par `cron`, de temps en temps), on risque donc des *"No such file or directory"* très déroutants dans le journal de Postfix).

On peut ensuite tester avec un client SASL. Cela peut être un Postfix mais je trouve `msmtp` `<http://msmtp.sourceforge.net/>` plus simple pour des tests. Avec cette configuration :

```
account bortzmeyer
host mail.bortzmeyer.org
port 587
from stephane@bortzmeyer.org
auth on
user stephane
password VRAIMENTSECRET
```

ma machine se connecte au port de soumission et s'authentifie. Postfix logue :

```
Jul 25 10:57:14 ariane postfix/submission[31542]: 25DFE240814: client=batilda.nic.fr[192.134.4.69], sasl_method=
```

Si on veut utiliser mutt, il peut parler directement à un tel serveur SMTP, en mettant dans le `.muttrc` :

```
set smtp_url="smtp://stephane@mail.bortzmeyer.org:587/"
```

Avec Thunderbird, cela serait *"Secure Authentication : yes"* et *"Connection Security : STARTTLS"*.

Autre façon de tester, depuis un programme Python, comme expliqué en <http://www.mkymong.com/python/how-do-send-email-in-python-via-smtplib/>.

Pour terminer de mettre en œuvre complètement le RFC 6409, il faudrait aussi, idéalement :

- Ne pas toucher aux en-têtes du message sur le port 25 mais seulement sur le 587, le port de soumission. Ce n'est pas évident avec Postfix [http://www.postfix.org/ADDRESS\\_REWRITING\\_README.html#william](http://www.postfix.org/ADDRESS_REWRITING_README.html#william) (voir aussi [http://www.postfix.org/postconf.5.html#local\\_header\\_rewrite\\_clients](http://www.postfix.org/postconf.5.html#local_header_rewrite_clients)).
- Loguer différemment les accès aux deux ports. Il n'existe malheureusement pas de moyen propre pour cela.

Merci beaucoup à Noel Jones pour son aide et ses explications.

Comme conclusion pessimiste, notons que, dans la réalité de l'Internet, il y aura toujours un logiciel bogué pour mettre en cause les plus belles configurations. Par exemple, Eudora sur MS-Windows a bien une option "Use port 587" mais, qu'elle soit cochée ou pas, Eudora utilise toujours le port 25 :-(. Heureusement, Thunderbird n'a pas ce problème.