Postfix

Autoriser le point pour faire du *plus addressing*

Le *plus addressing* est une solution simple pour donner des adresses mail différentes selon les sites auquels vous donnez votre adresse. Cela permet de faire des filtres selon la destinataire et éventuellement de retrouver l'origine de la fuite si vous commencez à recevoir du spam sur une adresse *plus-adressée*.

Concrètement, vous pouvez donner `foo+truc@exemple.org` à la place de `foo@exemple.org`, vous recevrez les mails envoyés à l'adresse avec un `+`.

Certains sites mal codés considèrent malheureusement qu'une adresse mail avec un `+` est invalide. Pour contourner ces sites de gougnafiers, on peut utiliser le point (`.`) à la place du caractère `+`.

N'autoriser que le point

Là, c'est tout simple : il suffit de changer le paramètre `recipient_delimiter` dans `/etc/postfix/mail.cf` et de recharger `postfix`.

Si vous n'avez pas encore utilisé le *plus addressing*, je vous conseille de faire ça, c'est le plus simple.

Autoriser les deux : le point et le plus

Il semblerait qu'on puisse, depuis Postfix 2.11, mettre plusieurs caractères dans `recipient_delimiter` (genre `recipient_delimiter = +-`). Mais ça ne fonctionne pas chez moi, sans doute à cause de mon groupware (Bluemind) qui fait plein de trucs tout seul (et ça me va très bien) mais qui, du coup, n'aime pas trop certaines modifications manuelles. Il semblerait aussi qu'il faille bidouiller Dovecot si vous l'utilisez derrière postfix.

Donc on va faire une réécriture de l'adresse de destination des mails.

Créez un fichier `/etc/postfix/point_addressing` qui contiendra ceci :

```
/^(.*)\.(.*)@(.*)$/ $1+$2@$3
```

> NB : Depuis ma mise à jour de Bluemind en version 4, il a fallu que j'ajoute ça à ce même fichier :

```
/^(.*)\+(.*)@(.*)$/ $1@$3
```

Dans `/etc/postfix/main.cf`, ajoutez `regexp:/etc/postfix/point_addressing` au début du paramètre `virtual_alias_maps` (l'ordre est important : la recherche d'alias va regarder les fichiers dans l'ordre). Chez moi, c'est passé de

```
virtual_alias_maps = hash:/etc/postfix/virtual_alias
```

à

```
virtual_alias_maps = regexp:/etc/postfix/point_addressing, hash:/etc/postfix/virtual_alias
```

Rechargez `postfix`, profitez □□

Si vous avez des utilisatrices dont l'identifiant comporte un point… là, j'avoue que c'est un peu compliqué. Il faudrait certainement adapter l'expression rationnelle, ou faire un autre fichier pour le `virtual_alias_maps`.

Permettre l'utilisation de Postfix par un serveur distant avec authentification

Si le serveur en face a une adresse IP fixe, on peut se contenter de l'ajouter au paramètre `mynetworks`, de relancer postfix et hop, le serveur est autorisé à utiliser le serveur postfix comme relais. Mais parfois, on veut une authentification avec login et mot de passe : serveur mutualisé, adresse IP non fixe, etc.

Tout d'abord : postfix n'est pas capable de faire de l'authentification. Il délègue ça à un service externe via `SASL`. On utilise dovecot ou cyrus pour ça.

Dovecot

Installation

```
apt install dovecot-core
```

Configuration

Dans le fichier `/etc/dovecot/conf.d/10-auth.conf`, changer le `auth_mecanisms` pour :

```
auth_mechanisms = plain login
```

Dans le même fichier, décommenter cette ligne à la fin du fichier :

```
!include auth-passwdfile.conf.ext
```

Et commenter celle-ci :

```
#!include auth-system.conf.ext
```

Dans `/etc/dovecot/conf.d/10-master.conf`, décommenter / ajouter ceci dans le bloc `service auth {}` :

```
unix_listener /var/spool/postfix/private/auth {
  mode = 0666
  user = postfix
  group = postfix
}
```

Et on relance le service :

```
systemctl restart dovecot.service
```

Gestion des utilisateurs

Avec la configuration qu'on a choisi plus haut (la doc vous tend les bras si vous voulez explorer d'autres pistes), les utilisateurs sont gérés dans le fichier `/etc/dovecot/users`.

Le format est le suivant (le format est complexe, je ne traite que les champs qui nous intéressent, encore une fois la doc…) :

```
login:{FORMAT DU MOT DE PASS}mot de passe::::::
```

Pour le login `foo`, le mot de passe `bar`, le tout sans chiffrement, ça donne :

```
foo:{PLAIN}bar::::::
```

Si vous voulez (et vous le voulez) chiffrer le mot de passe :

```
doveadm pw -s sha256-crypt
```

Tapez le mot de passe deux fois, la commande vous donnera un truc du genre de :

```
{SHA256-CRYPT}$5$bMeZKE.YWD8D2F6q$JpGqMfx4G6lyRuOkN2uKdRvexzrwJXNo6dWkUuZZjV/
```

Il suffit alors d'utiliser ça en lieu et place de `{PLAIN}bar` dans le fichier `/etc/dovecot/users`.

Pour voir les algorithmes de chiffrement disponible, faites `doveadm pw -l`.

Normalement, pas besoin de recharger dovecot quand on modifie le fichier.

Pensez à modifier les permissions du fichier :

```
chown root:dovecot /etc/dovecot/users
chmod 640 /etc/dovecot/users
```

NB : si vous utilisez Rspamd pour la signature DKIM, créez des utilisateurs avec une adresse mail contenant le domaine (ex : `foo@example.org:{PLAIN}bar::::::`) ou mettez `allow_username_mismatch = true;` dans `/etc/rspamd/local.d/dkim_signing.conf` et rechargez le service rspamd, sinon la signature DKIM ne sera pas ajoutée au mail.

Postfix

Ajouter ceci à `/etc/postfix/main.cf` :

```
#### SASL ####
## specify SASL type ##
smtpd_sasl_type = dovecot

## path to the SASL socket relative to postfix spool directory i.e. /var/spool/postfix ##
```

```
smtpd_sasl_path = private/auth

## postfix appends the domain name for SASL logins that do not have the domain part ##
smtpd_sasl_local_domain = example.org

## SASL default policy ##
smtpd_sasl_security_options = noanonymous

## for legacy application compatibility ##
#broken_sasl_auth_clients = yes

## enable SMTP auth ##
smtpd_sasl_auth_enable = yes

## smtp checks ##
## these checks are based on first match, so sequence is important ##
smtpd_recipient_restrictions = permit_mynetworks, permit_sasl_authenticated,
reject_unauth_destination
```

Et on relance le service, bien sûr :

```
systemctl reload postfix
```

Notez le `permit_mynetworks` : on n’utilise l’authentification SASL que si le serveur n’est pas dans la liste des serveurs autorisés par adresse IP.

Révision #11
Créé 5 mai 2021 08:53:23 par Luc
Mis à jour 21 août 2023 09:10:56 par Luc