Installer et configurer un serveur de mail

La configuration présentée ici permet d'héberger:

- les emails de plusieurs utilisateurs,
- sur plusieurs domaines différents,
- dotés d'un filtre antispam et antivirus, efficace et collaboratif,

- Les utilisateurs auront:
  - la possibilité de filtrer les messages côté serveur,
  - la possibilité de modifier leur mot de passe,
  - accès à des listes de discussion et de diffusion,

- Pour consulter leurs emails, ils disposerons :
  - d'un webmail moderne,
  - d'un accés IMAP sécurisé,

- Pour envoyer leurs emails, il auront accès à un serveur SMTP,
- l'administrateur pourra fixer un quota pour chaque boîte email, et des quota pour chaque domaine,
- L'ensemble sera sécurisé par SSL/TLS, avec un certificat délivré par l'autorité de certification collaborative CAcert.

Pour faire cela, nous utilisons, sur un serveur en Debian Stable, Postfix, Dovecot, GNU Mailman, MySql, Amavis, Clamav, Spamassassin, et Roundcube.

Il s'agit d'une configuration désormais ancienne (2013), mais les grands principes restent les mêmes. Si vous cherchez une configuration moderne et plus automatisée, je vous suggère de regarder du côté de mailu ou mail-in-a-box

Nous déconseillons vivement de reproduire trait pour trait cette configuration sans comprendre ce que vous faites.

La configuration d'un serveur email nécessite une bonne compréhension des principes de fonctionnement d'un système UNIX, la capacité d'analyser et de comprendre les difficultés qui se présenteront nécessairement, la capacité d'adapter cette documentation à votre propre environnement.

Nous vous conseillons de lire cette documentation en intégralité avant de tenter de l'appliquer.

Mal configuré, un serveur email peut produire des catastrophes : Vous risquez d'offrir un relais pour du spam et d'être blacklisté par les principaux serveurs emails de la planète.

Commencez dans un environnement de test et/ou faites vous aider par quelqu'un d'expérimenté.

Postfix : la colonne vertébrale du système de courrier

Postfix est un MTA, un *Mail Transfert Agent*, soit la brique de base du fonctionnement des emails sur Internet.

Nous avons fait le choix de le configurer pour utiliser des `virtual_domains`[1], et des `virtual_mailbox`. Cela signifie que nous **n'utilisons pas les comptes UNIX traditionnels** pour distribuer le courrier, mais des utilisateurs virtuels.

Il y a différentes façon de configurer Postfix pour qu'il utilise des utilisateurs virtuels. De notre côté, nous avons choisi d'utiliser **Mysql** pour cela, d'abord parceque cette solution est très souple (elle vous permettra de faire grossir facilement votre serveur de mails, d'héberger plusieurs domaines différents, notamment), et ensuite parce qu'elle est largement documentée.

Utiliser Postfix avec Mysql (du moins dans la configuration choisie ici), cela signifie que tout les utilisateurs, leurs mots de passe, leurs quota, etc, sont stockés dans une base de données, qui sera également utilisée par les autres applications qui ont besoin de ces données (Dovecot en particulier).

Pour créer cette base de donnée, et ensuite disposer d'une interface web facile à utiliser pour administrer les domaines et les utilisateur, nous utilisons Postfixadmin, dont nous verrons la configuration en préambule, avec l'installation de Postfix.

Intaller Postfix et Postfixadmin

Mysql

Pour plus de simplicité, vous devriez commencer par installer `mysql-server` et le configurer, cela nous éviteras d'avoir à la faire à l'installation de Postfix. N'oubliez pas de fixer un mot de passe fort pour l'utilisateur `root`.

Créez ensuite la bse de donnée qui accueillera toutes les informations nécessaires au bon fonctionnement de Postfix et Dovecot. Nous l'administrerons ensuite *via* PostfixAdmin[2] :

```
root@debian:~# mysql -u root -p
Enter password:
```

```
mysql> create database postfix;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON postfix.* TO 'postfixadmin'@'localhost'
IDENTIFIED BY 'postfixadmin-password';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SELECT ON postfix.* TO 'postfix'@'localhost' IDENTIFIED BY
'postfix-password';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
```

Postfix

Sur une distribution debian, vous pouvez installer simplement postfix avec la commande :

```
apt-get install postfix postfix-mysql
```

Si ce n'est pas déjà le cas, la suite `exim4`[3] sera supprimée. Une série de question vous sera posée, essayez d'y répondre de façon adaptée à votre environnement. Cependant, tout ces paramètres pourront être modifiés par la suite.

PostfixAdmin

PostfixAdmin ne fait pas parti des dépôts de Debian, nous allons donc devoir l'installer à la main. Par ailleurs, nous avons besoin d'un serveur web (tel que `Apache`) pour le faire fonctionner. La configuration d'Apache est hors du champs de cette documentation. Veillez néanmoins à sécuriser l'accés à PostfixAdmin par SSL (`https`), et je vous conseille également de dédier un sous-domaine à cet usage.

Vous devez télécharger PostfixAdmin ici. Prenez, évidement, la dernière version stable disponible (2.3.6 à ce jour). Puis décompressez l'archive là ou il vous semble que c'est judicieux.

Ajustez les droits sur le dossier `postfixadmin/` pour qu'il puisse être utilisable avec l'utilisateur qui fait tourner le serveur web, puis allez modifier le fichier `postfixadmin/config.inc.php`, pour obtenir ceci :

```
# Vous devez chercher ces variables dans le fichier

$CONF['configured'] = true;
...
```

```
$CONF['default_language'] = 'fr';
...
$CONF['database_type'] = 'mysqli';
$CONF['database_host'] = 'localhost';
$CONF['database_user'] = 'postfixadmin';
$CONF['database_password'] = 'postfixadmin-password';
$CONF['database_name'] = 'postfix';
...
$CONF['encrypt'] = 'md5crypt';
```

Vous pouvez désormais vous rendre sur l'interface de configuration web de PostfixAdmin[4] : https://postfixadmin.example.net/setup.php

Entrez un mot de passe de *setup* dans les champs prévu à cet effet, pour obtenir un *hash* de ce mot de passe, que vous irez ensuite reporter dans le fichier de configuration de postfixadmin.

Ouvrez ensuite le fichier `postfixadmin/config.inc.php` et repérez la ligne :

```
$CONF['setup_password'] = 'changeme';
```

Remplacez simplement `changeme` par le *hash* que vous avez obtenu.

Vous pouvez ensuite recharger la page de setup, et créer un administrateur.

Revenons sur le fichier `postfixadmin/config.inc.php`, pour finir de configurer PostfixAdmin

Cette opération n'est pas nécessaire à ce stade, mais faites le avant d'utiliser PostfixAdmin pour insérer vos premiers domaines et utilisateurs!

config.inc.php

```
...

$CONF['configured'] = true;

...
// Ici, j'ai forcé le https pour postfixadmin
// je vous conseille de faire de même
$CONF['postfix_admin_url'] = 'https://postfixadmin.monserveur.tld/';

...

$CONF['default_language'] = 'fr';
```

```
...

// Site Admin
// Define the Site Admins email address below.
// This will be used to send emails from to create mailboxes.
$CONF['admin_email'] = 'postmaster@monserveur.tld';

...

$CONF['smtp_server'] = 'localhost';
$CONF['smtp_port'] = '25';

...

$CONF['dovecotpw'] = "/usr/sbin/dovecotpw";

...

$CONF['min_password_length'] = 8;

...

// Default Aliases
// The default aliases that need to be created for all domains.
$CONF['default_aliases'] = array (
     'abuse' => 'abuse@monserveur.tld',
     'hostmaster' => 'hostmaster@monserveur.tld',
     'postmaster' => 'postmaster@monserveur.tld',
     'webmaster' => 'webmaster@monserveur.tld'
);

...
// Mailboxes
// If you want to store the mailboxes per domain set this to 'YES'.
// Examples:
//   YES: /usr/local/virtual/domain.tld/username@domain.tld
//   NO:  /usr/local/virtual/username@domain.tld
$CONF['domain_path'] = 'YES';
// If you don't want to have the domain in your mailbox set this to 'NO'.
// Examples:
//   YES: /usr/local/virtual/domain.tld/username@domain.tld
//   NO:  /usr/local/virtual/domain.tld/username
// Note: If $CONF['domain_path'] is set to NO, this setting will be forced
to YES    .
$CONF['domain_in_mailbox'] = 'NO';

...
```

```
// Default Domain Values
// Specify your default values below. Quota in MB.
$CONF['aliases'] = '10';
$CONF['mailboxes'] = '10';
$CONF['maxquota'] = '1000';

...

// Specify '' for Dovecot and 'INBOX.' for Courier.
$CONF['create_mailbox_subdirs_prefix']='';

...

$CONF['used_quotas'] = 'YES';
// if you use dovecot >= 1.2, set this to yes.
// Note about dovecot config: table "quota" is for 1.0 & 1.1, table
"quota2" is f    or dovecot 1.2 and newer
$CONF['new_quota_table'] = 'YES';
```

Voila, PostfixAdmin est configuré et opérationnel… mais nous n'avons pas encore de serveur mail qu'il puisse administrer!

Postfix : configuration de base

Un utilisateur dédié

Toute notre architecture de courrier est fondé sur des utilisateurs virtuels. Nous allons donc avoir besoin d'un utilisateur (et d'un groupe) UNIX qui "prêtera son identité" aux différents processus. Nous allons en créer un qui sera dédié à cet usage :

```
# Vous devez remplacer /srv/vmail par le chemin vers le dossier
# qui accueillera les boîtes mail de vos utilisateurs
groupadd -g 3000 vmail
useradd -d /srv/vmail -m -u 3000 -g 3000 vmail
```

Connecter Postfix à Mysql

Nous allons désormais créer, dans `/etc/postfix`, quelques fichiers qui indiquerons à Postfix comment se connecter à la base de données, afin d'y obtenir les information dont il a besoin :

mysql_virtual_mailbox_domains.cf

```
hosts = 127.0.0.1
```

```
user = postfix
password = postfix-password
dbname = postfix
query = SELECT domain FROM domain WHERE domain='%s' and backupmx = 0 and
active = 1
```

mysql_relay_domains.cf

```
hosts = 127.0.0.1
user = postfix
password = postfix-password
dbname = postfix
query = SELECT domain FROM domain WHERE domain='%s' and backupmx =
```

mysql_virtual_alias_maps.cf

```
hosts = 127.0.0.1
user = postfix
password = postfix-password
dbname = postfix
query = SELECT goto FROM alias WHERE address='%s' AND active = 1
```

mysql_virtual_mailbox_maps.cf

```
hosts = 127.0.0.1
user = postfix
password = postfix-password
dbname = postfix
query = SELECT maildir FROM mailbox WHERE username='%s' AND active = 1
```

Nous devons ensuite éditer `main.cf`[5], pour y faire apparaitre l'utilisateur et les fichiers connecteurs que nous venons de créer :

main.cf

```
# ceci n'est pas un fichier complet
# uniquement les variables qui nous intéressent

virtual_uid_maps = static:3000
virtual_gid_maps = static:3000
virtual_mailbox_base = /srv/vmail
virtual_mailbox_domains =
mysql:/etc/postfix/mysql_virtual_mailbox_domains.cf
```

```
virtual_mailbox_maps = mysql:/etc/postfix/mysql_virtual_mailbox_maps.cf
virtual_alias_maps = mysql:/etc/postfix/mysql_virtual_alias_maps.cf

relay_domains = mysql:/etc/postfix/mysql_relay_domains.cf
```

Postfix SASL & TLS

Pour permettre a Postfix d'authentifier les utilisateur, nous allons le configurer pour utiliser la couche SASL fournis par Dovecot, donc nous verrons la configuration ensuite.

main.cf

```
# SASL
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_tls_security_options = $smtpd_sasl_security_options
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_local_domain =
smtpd_sasl_authenticated_header = yes
```

Pour que cette authentification sois sécurisé, nous allons configurer TLS. J'inclus ici aussi bien la connexion de nos utilisateurs que celle des serveurs de courrier distants:

main.cf

```
# TLS
tls_random_source = dev:/dev/urandom

smtpd_use_tls = yes
#####################################################
# J'emploi ici un certificat fourni par cacert.org #
smtpd_tls_cert_file=/etc/ssl/cacert/cert.pem
smtpd_tls_key_file=/etc/ssl/cacert/privatekey.pem
smtpd_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
#####################################################
smtpd_tls_security_level = may
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtpd_tls_ask_ccert = yes
smtpd_tls_auth_only = yes
```

```
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtp_tls_security_level = may
smtp_tls_note_starttls_offer = yes
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtp_use_tls = yes
smtp_tls_loglevel = 1
```

Confier la distribution à Dovecot

Dovecot est un MDA, *Mail Delivery Agent* sécurisé, puissant et léger. Nous allons lui déléguer (outre le SASL, que nous avons vu plus haut), la distribution locale du courrier. Aprés réception, Postfix passera les courriers à Dovecot, qui se chargera de les délivrer dans les boites emails, aprés quelques menus traitement.

main.cf

```
mailbox_command = /usr/lib/dovecot/deliver
mailbox_transport = dovecot
dovecot_destination_recipient_limit = 1
```

Quelques options complémentaires

Vous trouverez dans excellente documentation de postfix (traduite en français) toutes les informations nécessaires pour comprendre ces options, plus triviales. Bien sur, n'hésitez à fouillez cette documentation : ce que nous utilisons ne vous conviendras peut-être pas.

main.cf

```
#spécifique à Debian
myorigin = /etc/mailname
#########################
smtpd_banner = $myhostname Mail Server
biff = no
readme_directory = no
append_dot_mydomain = no
delay_warning_time = 4h
myhostname = monserveur.tld
mydestination = machine.monserveur.tld, localhost.localdomain, localhost
relay_domains =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
```

```
message_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = ipv4
```

informer le processus master

Postfix est configuré de tel sorte que son processus maître, lancé avec le fichier `master.cf`, informe et organise de nombreux processus fils, avec la configuration principale. Nous devons donc reporter un certain nombre d'information dans le fichier `master.cf`

Voici notre fichier `master.cf` complet. Veuillez noter qu'il contient également des imformation pour Postscreen, amvis et Mailman, que nous aborderons plus tard.

master.cf

```
# Postfix master process configuration file.  For details on the format
# of the file, see the master(5) manual page (command: "man 5 master").
#
# Do not forget to execute "postfix reload" after editing this file.
#
#
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
#
# ==========================================================================
#smtp      inet  n       -       -       -       -       smtpd
smtp      inet  n       -       -       -       1       postscreen
smtpd     pass  -       -       -       -       -       smtpd
dnsblog   unix  -       -       -       -       0       dnsblog
tlsproxy  unix  -       -       -       -       0       tlsproxy
submission inet n       -       -       -       -       smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
#smtps     inet  n       -       -       -       -       smtpd
#  -o syslog_name=postfix/smtps
#  -o smtpd_tls_wrappermode=yes
```

```
#  -o smtpd_sasl_auth_enable=yes
#  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
#  -o milter_macro_daemon_name=ORIGINATING
#628       inet  n       -       -       -       -       qmqpd
pickup    fifo  n       -       -       60      1       pickup
cleanup   unix  n       -       -       -       0       cleanup
qmgr      fifo  n       -       n       300     1       qmgr
#qmgr     fifo  n       -       n       300     1       oqmgr
tlsmgr    unix  -       -       -       1000?   1       tlsmgr
rewrite   unix  -       -       -       -       -       trivial-rewrite
bounce    unix  -       -       -       -       0       bounce
defer     unix  -       -       -       -       0       bounce
trace     unix  -       -       -       -       0       bounce
verify    unix  -       -       -       -       1       verify
flush     unix  n       -       -       1000?   0       flush
proxymap  unix  -       -       n       -       -       proxymap
proxywrite unix -       -       n       -       1       proxymap
smtp      unix  -       -       -       -       -       smtp
relay     unix  -       -       -       -       -       smtp
#       -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq     unix  n       -       -       -       -       showq
error     unix  -       -       -       -       -       error
retry     unix  -       -       -       -       -       error
discard   unix  -       -       -       -       -       discard
local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       -       -       -       lmtp
anvil     unix  -       -       -       -       1       anvil
scache    unix  -       -       -       -       1       scache
#
# ====================================================================
# Interfaces to non-Postfix software. Be sure to examine the manual
# pages of the non-Postfix software to find out what options it wants.
#
# Many of the following services use the Postfix pipe(8) delivery
# agent.  See the pipe(8) man page for information about ${recipient}
# and other message envelope options.
# ====================================================================
#
# maildrop. See the Postfix MAILDROP_README file for details.
# Also specify in main.cf: maildrop_destination_recipient_limit=1
#
maildrop  unix  -       n       n       -       -       pipe
  flags=DRhu user=vmail argv=/usr/bin/maildrop -d ${recipient}
#
# ====================================================================
#
# Recent Cyrus versions can use the existing "lmtp" master.cf entry.
```

```
#
# Specify in cyrus.conf:
#   lmtp    cmd="lmtpd -a" listen="localhost:lmtp" proto=tcp4
#
# Specify in main.cf one or more of the following:
#  mailbox_transport = lmtp:inet:localhost
#  virtual_transport = lmtp:inet:localhost
#
# ====================================================================
#
# Cyrus 2.1.5 (Amos Gouaux)
# Also specify in main.cf: cyrus_destination_recipient_limit=1
#
#cyrus     unix  -       n       n       -       -       pipe
#  user=cyrus argv=/cyrus/bin/deliver -e -r ${sender} -m ${extension}
${user}
#
# ====================================================================
# Old example of delivery via Cyrus.
#
#old-cyrus unix  -       n       n       -       -       pipe
#  flags=R user=cyrus argv=/cyrus/bin/deliver -e -m ${extension} ${user}
#
# ====================================================================
#
# See the Postfix UUCP_README file for configuration details.
#
uucp      unix  -       n       n       -       -       pipe
  flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail
($recipient)
#
# Other external delivery methods.
#
ifmail    unix  -       n       n       -       -       pipe
  flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
bsmtp     unix  -       n       n       -       -       pipe
  flags=Fq. user=bsmtp argv=/usr/lib/bsmtp/bsmtp -t$nexthop -f$sender
$recipient
scalemail-backend unix -    n    n    -    2    pipe
  flags=R user=scalemail argv=/usr/lib/scalemail/bin/scalemail-store
${nexthop} ${user} ${extension}
mailman   unix  -       n       n       -       -       pipe
  flags=FR user=list
  argv=/usr/lib/mailman/bin/postfix-to-mailman.py ${nexthop} ${user}
dovecot   unix  -       n       n       -       -       pipe
  flags=DRhu user=vmail:vmail argv=/usr/lib/dovecot/deliver -f ${sender} -
d ${user}@${nexthop}
amavis unix - - - - 1 smtp
```

```
        -o smtp_data_done_timeout=1200
        -o smtp_send_xforward_command=yes

127.0.0.1:10025 inet n - - - - smtpd
        -o content_filter=
        -o local_recipient_maps=
        -o relay_recipient_maps=
        -o smtpd_restriction_classes=
        -o smtpd_client_restrictions=
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o mynetworks=127.0.0.0/8
        -o strict_rfc821_envelopes=yes
        -o
receive_override_options=no_unknown_recipient_checks,no_header_body_checks
#        -o smtpd_bind_address=127.0.0.1
```

Dovecot : distribuer le courrier aux utilisateurs

Comme dit plus haut, Dovecot est un MDA, *Mail Delivery Agent*, léger et sécurisé. Il est par ailleurs très facile d'emploi.

Installation

Dovecot est packagé dans Debian, pour l'installer, il suffit d'utiliser la commande :

```
apt-get install dovecot-common dovecot-core dovecot-antispam dovecot-imapd
dovecot-managesieved dovecot-mysql dovecot-pop3d dovecot-sieve
```

Sa configuration se trouve dans `/etc/dovecot` :

```
ls /etc/dovecot
conf.d/           dovecot-db.conf.ext        dovecot.pem           master-
users
dovecot.conf      dovecot-dict-sql.conf.ext  dovecot-sql.conf      private/
dovecot.conf.bak  dovecot-ldap.conf.ext      dovecot-sql.conf.ext  README
```

et l'essentiel de ce qui nous préoccupe dans `/etc/dovecot/conf.d/` :

```
ls conf.d
10-auth.conf        20-imap.conf                auth-ldap.conf.ext
10-director.conf    20-managesieve.conf         auth-master.conf.ext
```

```
10-logging.conf     20-pop3.conf                  auth-passwdfile.conf.ext
10-mail.conf        90-acl.conf                   auth-sql.conf.ext
10-master.conf      90-plugin.conf                auth-static.conf.ext
10-ssl.conf         90-quota.conf                 auth-system.conf.ext
10-tcpwrapper.conf  90-sieve.conf                 auth-vpopmail.conf.ext
15-lda.conf         auth-checkpassword.conf.ext
15-mailboxes.conf   auth-deny.conf.ext
```

Connecter Dovecot à Mysql

Nous avons besoins que Dovecot puise ses informations à la même source que Postfix : nous allons donc le connecter à la même base de donnée que Postfix :

- dans `etc/dovecot/conf.d/10-auth.conf`, on décommente la ligne suivante :

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

pour obtenir

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

- puis on va configurer le fichier `conf.d/auth-sql.conf.ext` :

auth-sql.ext.conf

```
# Authentication for SQL users. Included from auth.conf.
#
# <doc/wiki/AuthDatabase.SQL.txt>

passdb {
  driver = sql

  # Path for SQL configuration file, see example-config/dovecot-
sql.conf.ext
  args = /etc/dovecot/dovecot-sql.conf
}

# "prefetch" user database means that the passdb already provided the
# needed information and there's no need to do a separate userdb lookup.
# <doc/wiki/UserDatabase.Prefetch.txt>
#userdb {
#  driver = prefetch
#}

userdb {
  driver = sql
```

```
  args = /etc/dovecot/dovecot-sql.conf
}

# If you don't have any user-specific settings, you can avoid the
user_query
# by using userdb static instead of userdb sql, for example:
# <doc/wiki/UserDatabase.Static.txt>
#userdb {
  #driver = static
  #args = uid=vmail gid=vmail home=/var/vmail/%u
#}
```

- puis on écris le contenu du fichier `/etc/dovecot/dovecot-sql.conf` qui lui contient véritablement les informations de connexions à la base de donnée :

dovecot-sql.conf

```
driver = mysql
connect = host=127.0.0.1 dbname=postfix user=postfix password=postfix-
password
default_pass_scheme = MD5-CRYPT
user_query = SELECT '/srv/vmail/%d/%n' AS home, 3000 AS uid, 3000 AS gid,
CONCAT('*:bytes=', CAST(quota AS CHAR)) AS quota_rule FROM mailbox WHERE
username = '%u' AND active='1'
password_query = SELECT password FROM mailbox WHERE username = '%u'
```

Revue des fichiers de configuration de dovecot

Je passe sur les fichiers qui ne nous sont pas utile dans cette installation générique. Cependant, n'hésitez pas à ouvrir tout ces fichiers : ils sont trés bien documentés, et permettent de configurer finement Dovecot

Je ne reprends pas nécessairement ici l'ensemble des fichiers, mais seulement les options qui sont utiles pour notre cas

10-mail.conf

On configure ici le chemin d'accés de chaque boite mail.

10-auth.conf

```
...
mail_location = maildir:/srv/vmail/%d/%n:INDEX=/srv/vmail/%d/%n/indexes
...
namespace inbox {
  inbox = yes
}
...
```

10-master.conf

On active ici les services et les protocoles servis par Dovecot

10-master.conf

```
...
service imap-login {
  inet_listener imap {
    port = 143
  }
  inet_listener imaps {
    port = 993
    ssl = yes
  }
...
## J'ai choisi de ne pas activer pop3
## si on souhaite l'utiliser, décommenter ceci

#service pop3-login {
#  inet_listener pop3 {
#    port = 110
#  }
#  inet_listener pop3s {
#    port = 995
#    ssl = yes
#  }
#}
service lmtp {
  unix_listener lmtp {
```

```
    mode = 0666
  }

  # Create inet listener only if you can't use the above UNIX socket
  #inet_listener lmtp {
    # Avoid making LMTP visible for the entire internet
    #address =
    #port =
  #}
}
service auth {
  # auth_socket_path points to this userdb socket by default. It's
typically
  # used by dovecot-lda, doveadm, possibly imap process, etc. Users that
have
  # full permissions to this socket are able to get a list of all
usernames and
  # get the results of everyone's userdb lookups.
  #
  # The default 0666 mode allows anyone to connect to the socket, but the
  # userdb lookups will succeed only if the userdb returns an "uid" field
that
  # matches the caller process's UID. Also if caller's uid or gid matches
the
  # socket's uid or gid the lookup succeeds. Anything else causes a
failure.
  #
  # To give the caller full permissions to lookup all users, set the mode
to
  # something else than 0666 and Dovecot lets the kernel enforce the
  # permissions (e.g. 0777 allows everyone full permissions).
  unix_listener auth-userdb {
    #mode = 0666
    #user =
    #group =
  }

  # Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
  mode = 0666
  }
  # Auth process is run as this user.
  #user = $default_internal_user
}
service dict {
  # If dict proxy is used, mail processes should have access to its
socket.
  # For example: mode=0660, group=vmail and global
```

```
mail_access_groups=vmail
  unix_listener dict {
    mode = 0600
    user = vmail
    #group =
  }
}
```

10-ssl.conf

Là encore, j'utilise un certificat fournis par cacert.org

10-ssl.conf

```
...
ssl_cert = </etc/ssl/cacert/cert.pem
ssl_key = </etc/ssl/cacert/privatekey.pem
...
ssl_ca = </etc/ssl/certs/cacert.org.pem
...
```

15-lda.conf

`lda` signifie *local delivery agent* c'est ici qu'on le configure.

15-lda.conf

```
...
postmaster_address = postmaster@domain.tld
...
hostname = domain.tld
...
# If user is over quota, return with temporary failure instead of
# bouncing the mail.
quota_full_tempfail = yes
...
protocol lda {
  # Space separated list of plugins to load (default is global
mail_plugins).
  mail_plugins = $mail_plugins sieve quota
}
```

```
...
```

15-mailboxes.conf

Le nom du fichier est explicite, non ?

15-mailboxes.conf

```
...
# NOTE: Assumes "namespace inbox" has been defined in 10-mail.conf.
namespace inbox {

  ...

  # These mailboxes are widely used and could perhaps be created
automatically:
  mailbox Drafts {
    special_use = \Drafts
  }
  mailbox Junk {
    special_use = \Junk
  }
  mailbox Trash {
    special_use = \Trash
  }

  # For \Sent mailboxes there are two widely used names. We'll mark both
of
  # them as \Sent. User typically deletes one of them if duplicates are
created.
  mailbox Sent {
    special_use = \Sent
  }
  mailbox "Sent Messages" {
    special_use = \Sent
  }
  ...
}
```

20-imap.conf

20 imap.conf

```
...
protocol imap {
  ...

  # Maximum number of IMAP connections allowed for a user from each IP
address.
  # NOTE: The username is compared case-sensitively.
  mail_max_userip_connections = 10

 ...

  mail_plugins = $mail_plugins quota imap_quota antispam

 ...
}
```

90-plugin.conf

Ce fichier contient la configuration de tout les plugins activés dans dovecot (et qui nécessitent une configuration). Nous verrons le fonctionnement du plugin `antispam` entrainer_spamassassin_collaborativement plus loin.

90-plugin.conf

```
plugin {

#Antispam
  antispam_debug_target = syslog
  antispam_verbose_debug = 1
  antispam_backend = pipe
  antispam_trash = Trash
  antispam_spam = Junk
  antispam_allow_append_to_spam = no
  antispam_pipe_program = /usr/bin/sa-learn
  antispam_pipe_program_args = --no-sync
  antispam_pipe_program_spam_arg = --spam
  antispam_pipe_program_notspam_arg = --ham
}
```

90-quota.conf

Ce fichier permet de configurer Dovecot pour qu'il consulte les quota de chaque utilisateur dans la base

de données, et l'avertisse au franchissement d'un seuil :

90-quota.conf

```
 ...

## Quota limits

plugin {
  quota = dict:%u::proxy::quota
  quota_rule = *:storage=10M:messages=1000
}

...

## Quota warning
plugin {
  quota_warning = storage=99%% quota-warning 99 %u
  quota_warning2 = storage=97%% quota-warning 97 %u
  quota_warning3 = storage=95%% quota-warning 95 %u
  quota_warning4 = storage=90%% quota-warning 90 %u
  quota_warning5 = storage=85%% quota-warning 85 %u
  quota_warning6 = storage=80%% quota-warning 80 %u
  quota_warning7 = storage=75%% quota-warning 75 %u
}

 ...

## quota warning service
service quota-warning {
  executable = script /usr/local/bin/quota-warning.sh
  user = vmail
  unix_listener quota-warning {
  user = vmail
  }
}

 ...
```

Nous avons défini un script pour le service `quota-warning`, le voilà :

quota-warning.sh

```
#!/bin/sh

PERCENT=$1
```

```
USER=$2
FROM="postmaster@monserveur.tld"
qwf="/tmp/quota.warning.$$"

echo "From: $FROM
To: $USER
Subject: Votre boite mail est pleine ($PERCENT%)

Votre boite mail est pleine à $PERCENT%+, et risque d'être bientôt
incapable de recevoir du courrier.
Veuillez effacer des messages et vider la corbeille.
En cas de besoin, vous pouvez contacter votre administrateur:
postmaster@monserveur.tld" >> $qwf

cat $qwf | /usr/sbin/sendmail -f $FROM "$USER"
rm -f $qwf

exit 0
```

Pour que ce script soit utilisable, ajuster les permissions dessus :

```
chown dovecot:vmail /usr/local/bin/quota-warning.sh
chmod 775 /usr/local/bin/quota-warning.sh
```

dovecot.conf

Il faut tout de même faire un petit tour dans `/etc/dovecot.conf` pour y configurer la récupération des quota en base de donnée :

/etc/dovecot/dovecot.conf

```
 ...

dict {
  quota = mysql:/etc/dovecot/dovecot-dict-sql.conf.ext
  #expire = sqlite:/etc/dovecot/dovecot-dict-sql.conf.ext
}

 ...
```

Puis modifier `/etc/dovecot/dovecot-dict-sql.conf.ext` :

dovecot-dict-sql.conf.ext

```
connect = host=127.0.0.1 dbname=postfix user=postfixadmin
password=postfixadmin-password

 ...

map {
  pattern = priv/quota/storage
  table = quota2
  username_field = username
  value_field = bytes
}
map {
  pattern = priv/quota/messages
  table = quota2
  username_field = username
  value_field = messages
}

 ...
```

Utiliser SIEVE pour filtrer les emails

SIEVE est un protocole permettant de filtrer les emails directement sur le serveur. Il va nous permettre plusieurs choses, notamment de classer automatiquement les emails marqué comme étant du spam dans un sous-dossier `Junk/` des boîtes mail, et de donner la possibilité aux utilisateurs de filtrer leurs emails, de configurer des redirections ou encore des réponses automatiques, directement sur le serveur.

Pour l'activer, nous allons utiliser le fichier `conf.d/90-sieve.conf` :

90-sieve.conf

```
...
plugin {
  # The path to the user's main active script. If ManageSieve is used,
this the
  # location of the symbolic link controlled by ManageSieve.
  sieve = ~/dovecot.sieve

  ...
```

```
  sieve_dir = ~/sieve

  ...

  # Path to a script file or a directory containing script files that need
to be
  # executed before the user's script. If the path points to a directory,
all
  # the Sieve scripts contained therein (with the proper .sieve extension)
are
  # executed. The order of execution within a directory is determined by
the
  # file names, using a normal 8bit per-character comparison. Multiple
script
  # file or directory paths can be specified by appending an increasing
number.
  sieve_before = /srv/vmail/global_sieve/
  #sieve_before2 =
  #sieve_before3 = (etc...)

  # Identical to sieve_before, only the specified scripts are executed
after the
  # user's script (only when keep is still in effect!). Multiple script
file or
  # directory paths can be specified by appending an increasing number.
  #sieve_after =
  #sieve_after2 =
  #sieve_after2 = (etc...)

  ...

}
```

Grâce à la directive `sieve_before =/srv/vmail/global_sieve/` , tout les scripts placés dans ce dossier seront exécuté avant que le courrier soit distribué à l'utilisateur. C'est là que nous allons placer la directive permettant de trier le spam automatiquement vers un dossier spécial :

/srv/vmail/global_sieve/spam.sieve

```
require ["fileinto","imap4flags"];

# rule:[Tri des spams]
if allof (header :contains "X-Spam-Flag" "YES",
        # La ligne suivante n'est pas indispensable, elle sert à ne pas
        # générer d'erreur lorsque le destinataire est légitime mais
        # ne dispose pas d'une mailbox (ici, une liste mailman)
```

```
          not address :contains "To" "@listes.monserveur.tld",
{
        fileinto "Junk";
}
```

Il faudra compiler ce script avec la commande : `sievec /srv/vmail/global_sieve/spam.sieve srv/vmail/global_sieve/spam.svbin`

Nous verrons plus loin comment configurer Roundcube pour permettre aux utilisateurs de créer leurs propres filtres.

Combattre le spam : Postscreen, Amavis, Spamassassin et Clamav

Tel que configuré plus haut, votre serveur de courrier est opérationnel.

Cependant, si l'on s'arrête là, **le serveur va très rapidement s'effondrer sous le poids du spam**, et quand bien même il serait assez puissant pour tenir le coup, vos utilisateurs ne s'y retrouveraient plus. On estime qu'en décembre 2006, en france, 95% du trafic mail était du spam. Je doute que les choses se soient améliorées depuis[6] . Je vous présente ici une stratégie de lutte contre le spam qui est efficace, et qui permet aux utilisateurs de signaler quand un spam n'est pas détecté, ou bien quand un mail est classé comme spam par erreur, afin d'améliorer automatiquement le filtre Antispam.

Pour lutter contre le spam, nous allons déployer une stratégie en plusieurs étapes, à chaque étape de la récupération et de la distribution du courrier. Nous utiliserons, dans l'ordre :

- Postscreen : Un mécanisme intégré dans Postfix, qui filtre l'essentiel du spam, les plus grossiers, si l'on peut dire, avant même que le courrier ne soit accepté par le serveur, en testant le respect par le serveur émetteur des protocoles standardisés d'envoi d'emails. En effet, la plupart des spammeurs misent sur la quantité plutôt que la qualité, et préfèrent utiliser des serveurs extrêmement rapides et performant plutôt que de consommer des ressources à respecter les standards. Cette technique, outre qu'elle filtre une part immense du spam, à l'avantage d'économiser nos propres ressources, en coupant la communication très tôt, avant même d'avoir reçu le courrier.
- Postfix lui-même, qui fera également des vérifications sur le respect des protocoles, un peu plus tard dans la queue,

- Le trio Amavis/Spamassassin/Clamav, qui traitera tout les emails qui auront passés cette double barrière, fera une analyse de différent paramètres, et attribuera un score a chaque email. À partir d'un certain score, le courrier reçoit un en-tête (**X-Spam-Flag: YES**), qui nous permet de le ranger directement dans le dossier `Junk/` de chaque boîte email. À ce stade, nous ne rejetons aucun email, et les transmettons tous aux utilisateurs, afin d'éviter de perdre des emails faussement classé comme spam.
- Enfin, nous utilisons le plugin `antispam` de dovecot pour permettre aux utilisateurs de signaler quand un email est faussement classé comme spam, ou au contraire quand un spam passe au travers des mailles du filet.

Postscreen

Vous trouverez des pages détaillant l'usage de `postscreen` ici et là. Je ne peux que vous conseiller de lire la documentation ici, parce que si le mécanisme est très puissant, il est aussi susceptible d'avoir des effets de bord non-désirables, notamment l'introduction d'un délai dans la réception des emails.

Nous avons fait le choix d'utiliser une configuration assez *soft* de postscreen pour éviter ce délai.

Voila la configuration de postscreen que nous utilisons :

main.cf

```
# Postscreen
postscreen_access_list = permit_mynetworks

postscreen_dnsbl_sites = zen.spamhaus.org*3
                         b.barracudacentral.org*2
                         bl.spameatingmonkey.net*2
                         dnsbl.ahbl.org*2
                         bl.spamcop.net
                         dnsbl.sorbs.net
                         psbl.surriel.com
                         bl.mailspike.net
                         swl.spamhaus.org*-4
                         list.dnswl.org=127.[0..255].[0..255].0*-2
                         list.dnswl.org=127.[0..255].[0..255].1*-3
                         list.dnswl.org=127.[0..255].[0..255].[2..255]*-4
postscreen_dnsbl_threshold = 3
postscreen_dnsbl_action = enforce
```

```
postscreen_greet_banner = You have to respect RFCs
postscreen_greet_action = enforce
```

En pratique, cela signifie qu'avant d'accepter un mail, Postfix va :

- `postscreen_access_list` : Vérifier si le mail est issu de `mynetwork`, auquel cas le mail sera accepté sans autre forme de procès,
- `postscreen_dnsbl_*` : donner un score de "réputation" à l'expéditeur de l'email,
- `postscreen_greet_*` : Vérifier le bon respect de la norme SMTP,
- `_action = enforce` : En cas d'échec au test, rejeter le mail avec un code erreur *550 spam detected* (on pourrait utiliser `drop` pour couper brutalement la connexion et économiser un peu plus de ressources)

Pour que cette configuration fonctionne, il faut en informer le processus master. (Voir le fichier master.cf, plus haut)

Restrictions SMTPd dans Postfix

Une fois passé la barrière de Postscreen, le courrier est mis en queue pour être traité par Postfix. À ce stade, nous pouvons de nouveau effectuer quelques vérifications portant sur le respect de la norme SMTP :

main.cf

```
# Restrictions smtpd
smtpd_recipient_restrictions =
# ces régles sont pour tous:
    reject_non_fqdn_recipient,
    reject_non_fqdn_sender,
    reject_unknown_recipient_domain,
    reject_unknown_sender_domain,
    reject_unauth_pipelining,
# mes utilisateurs:
    permit_mynetworks,
    permit_sasl_authenticated,
# Bloquer quand il n'y a pas de reverse DNS
    reject_unknown_reverse_client_hostname,
# Bloquer quand le HELO/EHLO-hostnames est mal configuré
    reject_non_fqdn_hostname,
    reject_invalid_helo_hostname,
# Je ne suis pas responsable de ça :
    reject_unauth_destination,
# Si tous ça est ok,
    permit
```

Avec cette configuration, Postfix va effectuer des vérifications lors du dialogue avec le MTA distant. On peut effectuer ces vérifications à différents niveaux :

- `smtpd_helo_restrictions` : Dés le début de la *conversation* SMTP,
- `smtpd_sender_restrictions` : Un peut plus tard dans la *conversation*, lorsque le MTA distant annonce l'expéditeur du message
- `smtpd_recipient_restrictions` : Au moment ou le MTA distant annonce le destinataire du courrier. Nous plaçons nos vérifications à ce niveau pour bénéficier de toutes les informations précédentes, tout en ne passant pas à l'étape DATA, trop consomatrice de ressource si on fini par rejeter le courrier.
- `smtpd_data_restrictions` : Enfin, on peu utiliser à ce niveau des restrictions portant sur le contenu du mail (en cherchant par exemple des occurrences de termes couramment employés dans les spams. (Nous n'utilisons pas ces restrictions parce qu'elles nécessitent d'explorer le contenu du courrier, ce qui ne préserve pas nos propres ressources)

Voyons les restrictions utilisées :

Vérifications appliquées à tous les courriers

- `reject_non_fqdn_recipient` : Le destinataire de l'email doit être *fqdn (fully qualified domain name)*,
- `reject_non_fqdn_sender` : *idem* pour l'expéditeur,
- `reject_unknown_recipient_domain` : les domaines de destination inexistants seront rejetés,
- `reject_unknown_sender_domain` : *idem* pour les domaine de destination,
- `reject_unauth_pipelining` : Postfix utilise une technique, appelée *pipelining*, permettant d'envoyer plusieurs commande SMTP. Le protocole exige que le MTA distant interroge le MTA local sur ses capacitées à ce sujet. Si ce point n'est pas respecté (comme c'est souvent le cas des spammeurs), on rejette l'email.

Autoriser ses propres utilisateurs

- `permit_mynetworks` : Les utilisateurs locaux, une fois passé les tests précédents, sont autorisés à passer,
- `permit_sasl_authenticated` : Les utilisateurs authentifiés par SASL sont autorisés à passer

Vérifications appliquées aux MTA distants

- `reject_unknown_reverse_client_hostname` : rejeter si le *reverse DNS* est inexistant,
- `reject_non_fqdn_hostname` : rejeter si le domaine du MTA n'est pas *fqdn*,
- `reject_invalid_helo_hostname` : rejeter si la syntaxe du *HELO hostname* est invalide,
- `reject_unauth_destination` : rejeter tout les courriers pour lesquels notre serveur n'est pas la destination finale, ni un relais autorisé (type backup MX). **Cette restriction est essentielle : sans elle, nous devenons un OPEN RELAY**
- `permit` : enfin, accepter les courriers qui ont passés ces vérifications

vous trouverez ici une documentation détaillée sur ce qu'il est possible de faire avec les `smtpd_restrictions`.

Amavis, Spamassassin et Clamav

Les courriers qui ont passé ces premières barrières seront tous livrés aux utilisateurs.. En effet, nous allons désormais effectuer d'autres vérifications, avec notamment des techniques d'évaluation statistique, qui, si elles sont très efficaces, sont moins précises que les vérifications précédentes. Les spams que nous avons rejetés jusque là étaient grossiers, ceux qui restent sont bien mieux forgés.

Nous pourrions supprimer les emails que nous allons désormais identifier comme spams, mais nous risquerions d'avoir des *faux-positifs*, c'est à dire des emails marqué comme spams, mais qui en réalité n'en sont pas[7)]. Au contraire, nous allons ajouter des en-têtes aux email, indiquant si nous pensons qu'il s'agit de spam ou non, puis trier automatiquement le spam dans un dossier spécifique de la boîte mail de l'utilisateur.

Schéma général de fonctionnement

Aprés les vérifications précédentes, Postfix va transmettre le courrier à Amavis, dont la fonction sera de déléguer les opérations de vérifications antispam et antivirus à Spamassassin et Clamav, puis de restituer à Postfix les courriers vérifiés et dotés des fameux en-têtes à Postfix, pour qu'ils soient distribués localement.

Installation et configuration

Tout ces logiciels étant disponibles dans les dépots Debian, l'installation est très simple :

```
apt-get install amavis-new clamav clamav-daemon spamassassin
```

Néanmoins, un certains nombre de programmes externe seront nécessaires pour que clamav et spamassassin puissent analyser les emails :

```
apt-get install zoo unzip bzip2 arj nomarch lzop cabextract libnet-ldap-perl
libauthen-sasl-perl clamav-docs daemon libio-string-perl libio-socket-ssl-perl
libnet-ident-perl zip libnet-dns-perl p7zip unrar-free
```

Connecter Amavis à Postfix

On édite `master.cf` pour y ajouter la connection à Amavis[8)] :

master.cf

```
amavis unix - - - - 2 smtp
        -o smtp_data_done_timeout=1200
        -o smtp_send_xforward_command=yes

127.0.0.1:10025 inet n - - - - smtpd
        -o content_filter=
        -o local_recipient_maps=
        -o relay_recipient_maps=
        -o smtpd_restriction_classes=
        -o smtpd_client_restrictions=
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o mynetworks=127.0.0.0/8
        -o strict_rfc821_envelopes=yes
        -o
receive_override_options=no_unknown_recipient_checks,no_header_body_checks
        -o smtpd_bind_address=127.0.0.1
```

et on ajoutes dans `main.cf` :

main.cf

```
content_filter = amavis:[127.0.0.1]:10024
receive_override_options = no_address_mappings
```

Configurer Amavis

La configuration d'Amavis se trouve dans `/etc/amavis/conf.d/` :

```
ls conf.d
01-debian     15-av_scanners          25-amavis_helpers
05-domain_id  15-content_filter_mode  30-template_localization
05-node_id    20-debian_defaults      50-user
```

Dans la mesure du possible, nous ferons toutes nos manipulations dans le fichier `50-user`, les autres étant susceptibles d'être remplacés lors d'une mise à jour de Debian.

Cependant, nous allons éditer `15-content_filer_mode` pour activer les fonctions antispam et antivirus :

15-content_filer_mode

```
use strict;

 ...

# Supprimer les signes # devant les deux lignes
# suivantes pour activer l'antivirus

@bypass_virus_checks_maps = (
   \%bypass_virus_checks, \@bypass_virus_checks_acl,
\$bypass_virus_checks_re);

 ...

# Supprimer les signes # devant les deux lignes
# suivantes pour activer l'antispam

@bypass_spam_checks_maps = (
   \%bypass_spam_checks, \@bypass_spam_checks_acl,
\$bypass_spam_checks_re);

1;  # ensure a defined return
```

Voyons ce que nous avons à indiquer dans `50-user` :

50-user

```
use strict;

#
# Place your configuration directives here.  They will override those in
# earlier files.
#
# See /usr/share/doc/amavisd-new/ for documentation and examples of
# the directives you can use in this file

$QUARANTINEDIR =    '/srv/vmail/amavis-quarantine';             #
Quarantine Directory
$spam_admin =       'virusadmin@mondomain.tld';

$undecipherable_subject_tag = undef;
$sa_spam_subject_tag = undef;

@local_domains_maps = ( [ '.mondomain.tld', '.example.net', '.encore-un-
domaine.tld' ] );

$final_virus_destiny      = D_PASS;
```

```
$final_spam_destiny       = D_PASS;
$final_bad_header_destiny = D_PASS;

#$max_servers =1;

#------------ Do not modify anything below this line -------------
1;  # ensure a defined return
```

- `$QUARANTINEDIR` : cette directive indique ou seront stockés les emails mis en quarantaine,
- `$spam_admin` : cette boîte email recevra une notification pour les emails mis en quarantaine,
- `$undecipherable_subject_tag` : Par défaut, Amavis place un ***UNCHECKED*** dans le sujet des emails qu'il ne parvient pas à analyser, par exemple ceux chiffrés avec GPG. Cette option permet de laisser intact le sujet de ces emails.
- `sa_spam_subject_tag`: Par défaut, le sujet des email suspecté d'être du spam sera tagué avec la mention ***SPAM***. En indiquant undef, on s'assure que le sujet restera intact.
- `@local_domains_maps` : indiquer ici les domaines pour lesquels on accepte et on sert le courrier,
- `$final_virus_destiny`, `$final_spam_destiny`, `$final_bad_header_destiny` : Par défaut, Amavis rejette une partie des emails. Nous voulons qu'ils soinet tous distribué, avec les en-tête de spam. D'autres options sont possibles, néanmoins, soyez prudents, mal configurées, ces options peuvent vous faire passer pour un spammeur auprés d'autres serveurs.
- `$max_servers`: On peut réduire l'empreinte mémoire d'Amavis en ne lançant qu'une instance. Le processus sera alors un peu plus lent. Si vous utilisez cette option, il faudra reporter cette valeur dans `master.cf`

Amavis est un gros consommateur de mémoire vive. Si vous cherchez à en économiser un peu, vous pouvez désactiver la vérification antivirus et utiliser la directive `$max-servers` ci-dessus.

Configurer Spamassassin

Spamassassin est le logiciel qui va faire réellement le travail d'analyse antispam. Sa configuration se trouve dans `/etc/spamassassin/` :

```
17:46 root@effraie01 /etc/spamassassin # ls
65_debian.cf  local.cf            sa-update-keys/  v312.pre  v330.pre
init.pre      sa-update-hooks.d/  v310.pre         v320.pre
```

Là encore, effectuer nos modification dans le fichier `local.cf` permettra que nos modifications ne soient pas écrasées par une mise à jour.

local.cf

```
# This is the right place to customize your installation of SpamAssassin.
#
# See 'perldoc Mail::SpamAssassin::Conf' for details of what can be
# tweaked.
#
# Only a small subset of options are listed below
#
###########################################################################
#

bayes_path              /srv/vmail/bayes/bayes
bayes_file_mode         0777

razor_config            /etc/spamassassin/.razor/razor-agent.conf
pyzor_options           --homedir /etc/spamassassin

whitelist_from          *@mondomain.net

blacklist_from         *@domaine-de-spam.tld



# Add *****SPAM***** to the Subject header of spam e-mails
#
# rewrite_header Subject *****SPAM*****


#   Save spam messages as a message/rfc822 MIME attachment instead of
#   modifying the original message (0: off, 2: use text/plain instead)
#
report_safe 0


#   Set which networks or hosts are considered 'trusted' by your mail
#   server (i.e. not spammers)
#
# trusted_networks 212.17.35.


#   Set file-locking method (flock is not safe over NFS, but is faster)
#
lock_method flock


#   Set the threshold at which a message is considered spam (default: 5.0)
#
required_score 5.0
```

```
#   Use Bayesian classifier (default: 1)
#
use_bayes 1
use_bayes_rules 1

#   Bayesian classifier auto-learning (default: 1)
#
bayes_auto_learn 1


#   Set headers which may provide inappropriate cues to the Bayesian
#   classifier
#
bayes_ignore_header X-Bogosity
bayes_ignore_header X-Spam-Flag
bayes_ignore_header X-Spam-Status
bayes_ignore_header X-Spam-Level
bayes_ignore_header X-Spam-Score

#   Some shortcircuiting, if the plugin is enabled
#
ifplugin Mail::SpamAssassin::Plugin::Shortcircuit
#
#   default: strongly-whitelisted mails are *really* whitelisted now, if
the
#   shortcircuiting plugin is active, causing early exit to save CPU load.
#   Uncomment to turn this on
#
#shortcircuit USER_IN_WHITELIST       on
#shortcircuit USER_IN_DEF_WHITELIST   on
#shortcircuit USER_IN_ALL_SPAM_TO     on
#shortcircuit SUBJECT_IN_WHITELIST    on

#   the opposite; blacklisted mails can also save CPU
#
#shortcircuit USER_IN_BLACKLIST       on
#shortcircuit USER_IN_BLACKLIST_TO    on
# shortcircuit SUBJECT_IN_BLACKLIST    on

#   if you have taken the time to correctly specify your
"trusted_networks",
#   this is another good way to save CPU
#
# shortcircuit ALL_TRUSTED             on

#   and a well-trained bayes DB can save running rules, too
```

```
#
# shortcircuit BAYES_99                spam
# shortcircuit BAYES_00                ham

endif # Mail::SpamAssassin::Plugin::Shortcircuit
```

Je ne commente pas toutes les options, seulement celles qui méritent qu'on s'y attarde :

- `bayes_path` : indique le chemin ou se situera la base de données du filtre bayesien, ici dans le dossier `/srv/vmail/bayes/` . On indique `bayes` (ou autre chose, hein) à la fin du chemin pour indiquer le nom a donner à cette base. Ici, cela donne :

```
ls /srv/vmail/bayes
bayes_journal  bayes.mutex  bayes_seen  bayes_toks
```

- `razor_config` : indique le chemin de la configuration de `razor`. Pour créer cette configuration vous pouvez utiliser les commandes suivantes :

```
razor-admin -home=/etc/spamassassin/.razor -register
razor-admin -home=/etc/spamassassin/.razor -create
razor-admin -home=/etc/spamassassin/.razor -discover
```

  Vous trouverez plus de détails ici

- `whitelist_from` permet de ne jamais classer comme spam les courriers originaires des adresses indiqués ici. Vous pouvez utiliser * comme joker.
- `blacklist_from` : *idem* mais cette fois les courriers seront toujours considérés comme du spam
- `pyzor_options` : Pensez également à configurer `pyzor` avec la commande `pyzor --homedir /etc/spamassassin discover`

Activer spamassassin

Nous devons désormais éditer le fichier `/etc/default/spamassassin` pour rendre notre configuration opérante :

/etc/default/spamassassin

```
# /etc/default/spamassassin
 ...

# Change to one to enable spamd
ENABLED=1

 ...
```

```
# Cronjob
# Set to anything but 0 to enable the cron job to automatically update
# spamassassin's rules on a nightly basis
CRON=1
```

En-têtes et tri automatique des spams

À ce stade, notre système antispam est opérationnel. L'essentiel des spam sont rejetés par le serveur avant même de "toucher" notre disque dur, et ceux qui passent cette barrière, pour l'immense majorité d'entre eux, aurons des en-tête ajoutés par nos soins, permettant de les classer automatiquement dans un dossier dédié.

Nous avons déjà fait le nécessaire pour trier les spams, alors penchons nous sur les en-têtes des courriers pour comprendre de quoi il retourne.

Voici les en-têtes typique d'un courrier légitime :

en-tetes

```
Return-Path: <xxxxxx@hotmail.com>
Delivered-To: xxxxxx@monserveur.tld
Received: from localhost (localhost [127.0.0.1])
    by monserveur.tld (Postfix) with ESMTP id A9D2422CC9
    for <xxxxxx@monserveur.tld>; Sun, 22 Sep 2013 20:50:06 +0200 (CEST)
Received: from monserveur.tld ([127.0.0.1])
    by localhost (machine.monserveur.tld [127.0.0.1]) (amavisd-new, port
10024)
    with ESMTP id xj720Ikg_d_h for <xxxxxxx@monserveur.tld>;
    Sun, 22 Sep 2013 20:50:03 +0200 (CEST)
Received: from dub0-omc3-s3.dub0.hotmail.com (dub0-omc3-
s3.dub0.hotmail.com [157.55.2.12])
    by monserveur.tld (Postfix) with ESMTP id 89A0B21B6C
    for <xxxxxxx@monserveur.tld>; Sun, 22 Sep 2013 20:50:03 +0200 (CEST)
Received: from DUB119-W7 ([157.55.2.8]) by dub0-omc3-s3.dub0.hotmail.com
with Microsoft SMTPSVC(6.0.3790.4675);
    Sun, 22 Sep 2013 11:50:07 -0700
X-TMN: [4EMvMTMvlTHZi2SjY5KcuHp7pmupl0Cp]
X-Originating-Email: [xxxxxxx@hotmail.com]
Message-ID: <DUB119-W7F8CD7FA02AEBEAD49C58AF2C0@phx.gbl>
Content-Type: multipart/alternative;
    boundary="_95b36b34-03ca-44d9-9768-786336cd3ea9_"
From: Mr Anonyme <xxxxxxx@hotmail.com>
To: Mr Anonyme_aussi <xxxxxx@monserveur.tld>
Subject: Le sujet d'un email va ici
```

```
Date: Sun, 22 Sep 2013 20:50:07 +0200
Importance: Normal
In-Reply-To: <1379773197.13267.45.camel@kubrick>
References: <1379773197.13267.45.camel@kubrick>
MIME-Version: 1.0
X-OriginalArrivalTime: 22 Sep 2013 18:50:07.0132 (UTC)
FILETIME=[8E6D3DC0:01CEB7C4]
```

Comme vous le voyez, ces en-têtes regorges d'informations sur les différents acteur impliqués dans l'envoie et la réception du courrier, et sur les traitement qu'il font subir au dit courrier. Ces en-têtes varient, évidement, en fonction de nombreux paramètres.

Si un email est traité par spamassassin, à partir d'un certain score [9)] nous, verrons quelques lignes supplémentaires apparaître :

en-tete-spammy

```
X-Quarantine-ID: <FrfUwTROFeOr>
X-Spam-Flag: YES
X-Spam-Score: 18.9
X-Spam-Level: ******************
X-Spam-Status: Yes, score=18.9 tagged_above=-50 required=6.31
    tests=[ADVANCE_FEE_3_NEW=3.499, ADVANCE_FEE_3_NEW_MONEY=2.238,
    FREEMAIL_FORGED_REPLYTO=2.503, FREEMAIL_REPLYTO=1,
    LOTS_OF_MONEY=0.001, MONEY_FRAUD_3=2.917, MONEY_FROM_41=1.999,
    RCVD_IN_BL_SPAMCOP_NET=1.246, RCVD_IN_DNSWL_NONE=-0.0001,
    RP_MATCHES_RCVD=-0.652, SUBJ_ALL_CAPS=1.625, URIBL_BLOCKED=0.001,
    US_DOLLARS_3=2.523] autolearn=spam
```

- **X-Quarantine-ID** apparait si Amavis met le mail en quarantaine (au delà d'un score qui ne laisse vraiment aucun doute sur le statut du courrier (10, je crois). l'ID permet de retrouver le courrier dans le dossier de quarantaine.
- **X-Spam-Flag** est à YES quand le courrier atteinds un score (6.31 par défaut) défini dans la configuration d'Amavis. Vous pouvez abaisser ou augmenter le score si vous le souhaitez, mais je ne vous le conseille pas. C'est sur ce paramètre là que nous trions ou non le mail dans le dossier Junk/
- **X-Spam-Score** : le score obtenu par le courrier (là, nous avons un champion!)
- **X-Spam-Level** : le même score représenté par une gauge constituée de caractères (*). Certains clients mail utiliserons cet en-tête
- **X-Spam-Status** : Un résumé des en-tête précédents, et la liste des règles de spamassassin qui permettent d'attribuer ce score au courrier.

Entraîner Spamassassin collaborativement

C'est ici que nous reparlons du plugin `antispam` de Dovecot, que nous allons utiliser en conjonction avec la commande `sa-learn` de Spamassassin, pour permettre à nos utilisateurs de signaler les *faux-positifs*.

Ce plugin permet le comportement suivant:

- Lorsqu'un mail qui a été (automatiquement) classé dans le dossier `Junk/` d'une boîte mail est sorti manuellement, par l'utilisateur, il sera passé, via un `pipe`, en option à la commande `sa-learn –ham`,
- Lorsqu'un mail qui n'a pas été classé comme spam est déplacé manuellement dans le dossier `Junk/` de l'utilisateur, il est passé en option à la commande `sa-learn –spam`
- Ainsi, la base de données du filtre bayesien de Spamassassin est améliorée finement par l'action des utilisateurs, afin que les *faux-positifs* soient ensuite correctement détectés.

Pour que cette base de données soient prise en compte par Spamassassin, nous allons lancer la commande suivante avec `cron` :

crontab

```
30 5 * * * /usr/bin/sa-learn --sync
```

Il n'y a rien de plus à configurer, puisque nous avons déjà configuré le plugin plus haut.

Des mailing-lists avec GNU Mailman

Maintenant que notre serveur de mail est opérationnel, nous pouvons installer GNU Mailman pour faire fonctionner des mailing-lists.

Attention! Cette installation de mailman est adaptée à notre contexte, c'est à dire avec Postfix configuré pour utilisé des `virtual_domain` et des `virtual_mailbox` stockés dans une base Mysql. Elle ne fonctionnera pas dans un autre contexte.

Installation

Là encore, sous Debian, l'installation est très simple :

`apt-get install mailman` devrait suffire.

Un sous-domaine dédié

Pour faciliter l'accés à Mailman, nous allons lui dédier un sous-domaine, qui sera le même dans Apache et pour les adresses de listes : listes.monserveur.tld

Pour commencer, vous pouvez supprimer le lien symbolique `mailman.conf` dans `/etc/apache2/conf.d/` : celui ci rend l'interface de mailman accessible à une adresse qui n'est pas celle de notre sous-domaine.

Puis, nous allons créer un sous-domaine de ce type :

/etc/apache2/sites-available/listes.monserveur.tld

```
# Forcer un usage en https
<VirtualHost *:80>
   ServerName listes.monserveur.tld
   Redirect permanent / https://listes.monserveur.tld/
</VirtualHost>

<VirtualHost *:443>
    ServerAdmin root@monserver.tld
    ServerName listes.monserveur.tld
    # On active SSL
    SSLEngine on
    # Un certificat de chez cacert.org
    SSLCertificateFile    /etc/ssl/cacert/cert.pem
    SSLCertificateKeyFile /etc/ssl/cacert/privatekey.pem
    <FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>
     BrowserMatch "MSIE [2-6]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
     # MSIE 7 and newer should be able to use keepalive
     BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

    Alias /pipermail/ /var/lib/mailman/archives/public/
    <Directory "/var/lib/mailman/archives/public">
            Options +FollowSymLinks
            Order allow,deny
            Allow from all
    </Directory>
    Alias /images/mailman/ /usr/share/images/mailman/
    ScriptAlias /admin /usr/lib/cgi-bin/mailman/admin
```

```
    ScriptAlias /admindb /usr/lib/cgi-bin/mailman/admindb
    ScriptAlias /confirm /usr/lib/cgi-bin/mailman/confirm
    ScriptAlias /create /usr/lib/cgi-bin/mailman/create
    ScriptAlias /edithtml /usr/lib/cgi-bin/mailman/edithtml
    ScriptAlias /listinfo /usr/lib/cgi-bin/mailman/listinfo
    ScriptAlias /options /usr/lib/cgi-bin/mailman/options
    ScriptAlias /private /usr/lib/cgi-bin/mailman/private
    ScriptAlias /rmlist /usr/lib/cgi-bin/mailman/rmlist
    ScriptAlias /roster /usr/lib/cgi-bin/mailman/roster
    ScriptAlias /subscribe /usr/lib/cgi-bin/mailman/subscribe
    ScriptAlias /mailman/ /usr/lib/cgi-bin/mailman/
    ScriptAlias / /usr/lib/cgi-bin/mailman/listinfo
    <Directory /var/lib/mailman/archives/public/>
        Options FollowSymLinks
    </Directory>
    ErrorLog /var/log/apache2/listes.monserveur.tld.error.log
    LogLevel warn
    CustomLog "|/usr/bin/tee -a
/var/log/apache2/listes.monserveur.tld.access.log | /usr/bin/tee -a
/var/log/apache2/access.log" combined
    ServerSignature On
</VirtualHost>
```

Ce vhost active notamment SSL, redirige l'accès sur le port 80 vers le port 443, et agrège le access.log du vhost à l'access.log global du serveur. À vous de voir si cette configuration vous convient, elle n'est pas obligatoire. Ce qui est important ici, ce sont les directives `Alias` et `ScriptAlias`, qui permettent à la fois d'accéder aux ressources nécessaires pour Mailman, et d'avoir de jolies URL.

Vous pouvez désormais lancer la commande `newlist mailman`, qui va créer une première liste sur votre serveur, nommée mailman, indispensable au bon fonctionnement du logiciel.

Intégrer Mailman à Postfix

Il faut désormais informer Postfix de l'existence de Mailman, et de la façon de traiter les emails qui lui sont adressés :

main.cf

```
 ...
mydestination = machine.monserveur.tld, listes.monserveur.tld,
localhost.localdomain, localhost

 ...

mailman_destination_recipient_limit = 1
```

```
 ...

alias_maps = hash:/etc/aliases,
             hash:/var/lib/mailman/data/aliases
alias_database = $alias_maps

 ...

transport_maps = hash:/etc/postfix/transport,
                 hash:/var/lib/mailman/data/transport-mailman


 ...

relay_domains = mysql:/etc/postfix/mysql_relay_domains.cf,
listes.monserveur.tld

 ...

relay_recipient_maps = hash:/var/lib/mailman/data/virtual-mailman

 ...
```

Dans `/etc/postfix/transport` :

transport

```
listes.monserveur.tld      mailman:
```

N'oubliez pas la commande `postmap -v /etc/postfix/transport` à chaque modification de ce fichier.

L'essentiel des directives que nous modifions ici visent à informer Postfix qu'il ne doit pas chercher les destinataire de `*@listes.monserveur.tld` dasn la base Mysql, mais dans la configuration de Postfix.

Il nous faut également modifier `master.cf`

master.cf

```
 ...
mailman   unix  -       n       n       -       -       pipe
  flags=FR user=list
  argv=/usr/lib/mailman/bin/postfix-to-mailman.py ${nexthop} ${user}
```

```
...
```

Ces lignes sont normalement déjà présentes, vous n'avez qu'a les décommenter.

Configurer Mailman

L'essentiel de la configuration de Mailman se fait dans le fichier `/etc/mailman/mm_cfg.py`. Ce fichier est très bien documenté, je n'indique ici que ce qui me semble important.

mm_cfg.py

```
...

# Pour que nos URLs fonctionnent
DEFAULT_URL_PATTERN = 'https://%s/'
IMAGE_LOGOS         = '/images/mailman/'

...

DEFAULT_EMAIL_HOST = 'listes.monserveur.tld'

...

# Required when setting any of its arguments.
add_virtualhost(DEFAULT_URL_HOST, DEFAULT_EMAIL_HOST)

...

#---------------------------------------------------------------
# Uncomment this if you configured your MTA such that it
# automatically recognizes newly created lists.
# (see /usr/share/doc/mailman/README.Exim4.Debian or
# /usr/share/mailman/postfix-to-mailman.py)
#MTA=None   # Misnomer, suppresses alias output on newlist
POSTFIX_STYLE_VIRTUAL_DOMAINS = ['listes.monserveur.tld']
DEB_LISTMASTER = 'postmaster@monserveur.tld'
#---------------------------------------------------------------
# Uncomment if you use Postfix virtual domains (but not
# postfix-to-mailman.py), but be sure to see
# /usr/share/doc/mailman/README.Debian first.
MTA='Postfix'
# Pourtant nous utilisons postfix-to-mailman.py...
# ici ça ne fonctionne pas sans cette option

...
```

```
# Permet aux administrateurs de liste de la supprimer
OWNERS_CAN_DELETE_THEIR_OWN_LISTS = 1
# Les archives sont privees par defaut
DEFAULT_ARCHIVE_PRIVATE = 1
# Footer par defaut
DEFAULT_MSG_FOOTER = """_____________________________________________
Liste %(real_name)s
%(real_name)s@%(host_name)s
Gestion de l'abonnement et des options sur :
%(web_page_url)slistinfo%(cgiext)s/%(_internal_name)s
"""
POSTFIX_MAP_CMD = '/var/lib/mailman/data/virtual_to_transport'
PUBLIC_ARCHIVE_URL = 'https://%(hostname)s/pipermail/%(listname)s'
```

dans `/var/lib/mailman/data/virtual_to_transport` (qui doit être exécutable : `chmod +x`, nous mettons le contenu suivant :

virtual_to_transport

```
#!/bin/sh
sed -r -e 's/(^[^#]\S+\s+).+$/\1local/' $1 \
 > /var/lib/mailman/data/transport-mailman
/usr/sbin/postmap /var/lib/mailman/data/transport-mailman
/usr/sbin/postmap /var/lib/mailman/data/virtual-mailman
```

Vérifiez ensuite la présence de `postfix_to_mailman.py` dans le dossier `/etc/mailman/`. S'il n'y est pas, copiez-le depuis `/usr/share/mailman/postfix-to-mailman.py`.

Roundcube, un webmail moderne

Notre serveur de courrier est désormais pleinement fonctionnel, les utilisateurs peuvent s'y connecter en IMAP et SMTP, disposent d'un bon antispam, et ont accès à des mailing-listes. Nous reste à leur fournir un bon webmail, pour qu'ils accèdent à leurs emails quand ils n'ont pas accès à leur propre ordinateur.

Roundcube est un webmail moderne, à l'interface simple et pratique, et il disposent de plugins qui vont nous permettre de tirer le maximum de notre serveur de courrier.

Installation

Encore une fois, ce logiciel est packagé par Debian[10], il est donc facile de l'installer :

```
apt-get install roundcube roundcube-core roundcube-mysql roundcube-plugins
```

`roundcube-plugins-extra`

Par défaut, Roundcube sera accessible à l'adresse `http://monserveur.tld/roundcube`. Vous pouvez conserver cette configuration ou lui dédier un vhost. Néanmoins, je vous conseille de forcer l'usage de SSL pour l'accés à Roundcube, ça éviteras que les mots de p&asse de vos utilisateurs se balladent dans la nature.

Configuration de base

La configuration de Roundcube se trouve dans `/etc/roundcube/` :

```
ls /etc/roundcube
apache.conf             debian-db.php  main.inc.php           plugins/
apache.conf.dpkg-dist   htaccess       main.inc.php.ucf-dist
db.inc.php              lighttpd.conf  mimetypes.php
```

Nous allons modifier le fichier `main.inc.php` pour qu'il corresponde à notre serveur de courrier (et quelques options pratique en plus) :

main.inc.php

```
...

// log driver:  'syslog' or 'file'.
$rcmail_config['log_driver'] = 'syslog';

...
// TCP port used for IMAP connections
$rcmail_config['default_port'] = 993;

...

// IMAP AUTH type (DIGEST-MD5, CRAM-MD5, LOGIN, PLAIN or empty to use
// best server supported one)
$rcmail_config['imap_auth_type'] = null;

...

// SMTP port (default is 25; 465 for SSL)
// quand à nous, nous préférons le port "submission", protégé par TLS
$rcmail_config['smtp_port'] = 587;

...

// SMTP AUTH type (DIGEST-MD5, CRAM-MD5, LOGIN, PLAIN or empty to use
// best server supported one)
```

```
$rcmail_config['smtp_auth_type'] = 'PLAIN';

...

// THIS OPTION WILL ALLOW THE INSTALLER TO RUN AND CAN EXPOSE SENSITIVE
CONFIG DATA.
// ONLY ENABLE IT IF YOU'RE REALLY SURE WHAT YOU'RE DOING!
$rcmail_config['enable_installer'] = false;

...

// enforce connections over https
// with this option enabled, all non-secure connections will be
redirected.
// set the port for the ssl connection as value of this option if it
differs from the default 443
$rcmail_config['force_https'] = true;



...

// automatically create a new Roundcube user when log-in the first time.
// a new user will be created once the IMAP login succeeds.
// set to false if only registered users can use this service
$rcmail_config['auto_create_user'] = true;



...

// don't allow these settings to be overriden by the user
$rcmail_config['dont_override'] =
array('logout_purge','drafts_mbox','junk_mbox','sent_mbox','trash_mbox');

...

// List of active plugins (in plugins/ directory)
// Attention, les plugins doivent être présent et configurés
// avant d'être listés ici
$rcmail_config['plugins'] =
array('fail2ban','archive','markasjunk2','password','quickrules','keyboard
_shortcuts','jqueryui','sieverules','compose_addressbook');

...

// store draft message is this mailbox
// leave blank if draft messages should not be stored
// NOTE: Use folder names with namespace prefix (INBOX. on Courier-IMAP)
$rcmail_config['drafts_mbox'] = 'Drafts';
```

```
// store spam messages in this mailbox
// NOTE: Use folder names with namespace prefix (INBOX. on Courier-IMAP)
$rcmail_config['junk_mbox'] = 'Junk';

// store sent message is this mailbox
// leave blank if sent messages should not be stored
// NOTE: Use folder names with namespace prefix (INBOX. on Courier-IMAP)
$rcmail_config['sent_mbox'] = 'Sent';

// move messages to this folder when deleting them
// leave blank if they should be deleted directly
// NOTE: Use folder names with namespace prefix (INBOX. on Courier-IMAP)
$rcmail_config['trash_mbox'] = 'Trash';

// display these folders separately in the mailbox list.
// these folders will also be displayed with localized names
// NOTE: Use folder names with namespace prefix (INBOX. on Courier-IMAP)
$rcmail_config['default_imap_folders'] = array('INBOX', 'Drafts', 'Sent',
'Junk', 'Trash');

// automatically create the above listed default folders on first login
$rcmail_config['create_default_folders'] = true;

// protect the default folders from renames, deletes, and subscription
changes
$rcmail_config['protect_default_folders'] = true;

// if in your system 0 quota means no limit set this option to true
$rcmail_config['quota_zero_as_unlimited'] = true;

...

// display remote inline images
// 0 - Never, always ask
// 1 - Ask if sender is not in address book
// 2 - Always show inline images
$rcmail_config['show_images'] = 0;

...

// compose html formatted messages by default
// 0 - never, 1 - always, 2 - on reply to HTML message only
// autant favoriser les bonnes pratiques
$rcmail_config['htmleditor'] = 0;

...
```

```
// default setting if preview pane is enabled
$rcmail_config['preview_pane'] = true;

...

// Clear Trash on logout
$rcmail_config['logout_purge'] = true;

...

// Set to true to never delete messages immediately
// Use 'Purge' to remove messages marked as deleted
$rcmail_config['flag_for_deletion'] = 'Purge';

...

// If true all folders will be checked for recent messages
// utile puisque nous utilisons SIEVE
$rcmail_config['check_all_folders'] = true;

...

// 'Delete always'
// This setting reflects if mail should be always deleted
// when moving to Trash fails. This is necessary in some setups
// when user is over quota and Trash is included in the quota.
$rcmail_config['delete_always'] = true;

...
```

Des plugins pour Roundcube

Tout les plugins utilisés ici (ceux listés dans le fichier de configuration au-dessus), sont fournis par les paquets Debain que nous avons installé. Vous devez vérifier leur présence dans `/etc/roundcube/plugins/`. Si certains sont absents, vous pouvez les copier depuis `/usr/share/roundcube/plugins/`.

Ci dessous, j'explique comment configurer certains d'entre eux : si le fichier de configuration n'est pas présent, ou qu'il est vide, copier celui qui est présent dans `/usr/share/roundcube/plugins/nom-du-plugin/`

Ceux qui ne sont pas listés ne nécessitent pas de configuration particulière.

Fail2ban

Fail2ban permet de bloquer automatiquement, grâce à Iptables, les tentative de conections par force brute. Malheureusement, sans ce plugin, c'est l'IP de votre propre serveur qui est bloquée, et aucun de vos utilisateur ne peux plus se connecter.

La configuration générale d'IPtables et de Fail2ban n'est pas couverte ici, uniquement celle du plugin

/etc/fail2ban/jail.conf

```
...

[roundcube]
enabled  = true
port     = http,https
filter   = roundcube
action   = iptables-multiport[name=roundcube, port="http,https"]
logpath  = /var/log/syslog
bantime = 300
maxretry = 5

...
```

/etc/fail2ban/filter.d/roundcube.conf

```
[Definition]
failregex = FAILED login for .*. from <HOST>
ignoreregex =
```

MarkAsJunk2

Ce plugin permet de tirer profit du plugin antispam de Dovecot que nous avons configuré plus haut.

markasjunk2/config.inc.php

```
...
$rcmail_config['markasjunk2_learning_driver'] = null;
...
$rcmail_config['markasjunk2_read_spam'] = true;
```

```
...
$rcmail_config['markasjunk2_unread_ham'] = true;
...
$rcmail_config['markasjunk2_move_spam'] = true;
...
$rcmail_config['markasjunk2_move_ham'] = true;
...
$rcmail_config['markasjunk2_mb_toolbar'] = true;
...
$rcmail_config['markasjunk2_spam_cmd'] = null;
...
$rcmail_config['markasjunk2_ham_cmd'] = null;
...
```

Password

Ce plugin permet aux utilisateurs de changer de mot de passe directement dans Roundcube. Aprés l'avoir configuré, vous pouvez bloquer l'accès à la section *users* de Postfixadmin

password/config.inc.php

```
...
// A driver to use for password change. Default: "sql".
// See README file for list of supported driver names.
$rcmail_config['password_driver'] = 'sql';
...
// Determine whether current password is required to change password.
// Default: false.
$rcmail_config['password_confirm_current'] = true;
...
// Pour garantir des mots de passe (un peu) sécurisés
// Require the new password to be a certain length.
// set to blank to allow passwords of any length
$rcmail_config['password_minimum_length'] = 8;

// Require the new password to contain a letter and punctuation character
// Change to false to remove this check.
$rcmail_config['password_require_nonalpha'] = true;
...
// SQL Driver options
...
$rcmail_config['password_db_dsn'] = 'mysql://postfixadmin:postfixadmin-
password@localhost/postfix';
...
$rcmail_config['password_query'] = 'UPDATE mailbox SET password=%c WHERE
```

```
username=%u  LIMIT 1';
...
$rcmail_config['password_dovecotpw_method'] = 'CRAM-MD5';
...
```

SieveRules

Ce plugin permet aux utilisateurs de créer leurs propres scripts SIEVE, avec une interface graphique simple et complète.

sieverules/config.inc.php

```
...
$rcmail_config['sieverules_host'] = 'localhost';
...
$rcmail_config['sieverules_port'] = 4190;
...
```

Annexes

Surveiller les logs

D'une façon générale, vous avez intérêt à surveiller les logs de votre serveur, il y a de nombreux programmes pour cela. C'est encore plus vrai pendant que vous configurez un serveur de courrier. À chaque étape, vous devez avoir littéralement **le nez sur les logs**, pour détecter au plus vite une erreur de configuration (il y en aura forcément). Vous n'avez pas envie de vous apercevoir que votre serveur de mail est planté au bout de deux jours, une fois que vous aurez perdu des emails, non ?

Un petit programme très pratique pour ça se nomme ccze (`apt-get install ccze`). Il colorise les logs pour les rendre plus lisible : les `error` et les `warning` vous sauterons aux yeux.

Il s'utilise de la façon suivante : `tail -f /var/log/mail.log | ccze` (il peut coloriser également d'autres logs que ceux des emails). Chaque fois que vous relancerez un service, tel que postfix, dovecot, amavis ou spamassassin, il est **impératif** que vous ayez en même temps le `mail.log` qui défile sous vos yeux, sinon vous courez à la catastrophe.

Trouver de la documentation

Tout les logiciels utilisés ici sont dotés d'une documentation officielle précise et précieuse. Lisez la!

En particulier :

Postfix

- documentation officielle
- Sa traduction en français (Je ne sais pas si elle est à jour, j'utilise la doc du site officiel]]
- Virtual_domains hosting
- SASL
- TLS
- Postscreen
- Tout les paramètres de main.cf

Dovecot

- Documentation officielle
- Utilisateurs virtuels
- Tout les plugins
- Sieve et ManageSieve

Spamassassin

- Documentation officielle
- FAQ de Spamassassin
- Using Spamassassin

Trouver de l'aide

Malgré la documentation, vous aurez sans doute besoin d'aide. Le plus simple est d'aller la chercher à la source : sur les mailing-listes des logiciels concernés. Aussi bien Postfix que Spamassasin et Dovecot disposent de mailing-listes actives et nombreux sont ceux qui sont prêts à vous aider. Veillez simplement à poser correctement vos question, en fournissant les éléments de votre configuration, et des extraits des logs montrant votre problème.

Des règles Spamassassin pour le spam français

Les règles standards de spamassassin sont très efficace, mais voici quelques règles suplémentaires que j'utilise pour être plus restrictif avec le spam français :

spamassassin_fr.rules

```
##########################################################################
###########
```

```
##### FRENCH SPECIFIC SPAMASSASSIN RULES.
##### USE AND REDISTRIBUTE WITH THIS NOTE AT YOUR OWN RISK AND PLEASURE.
##### AUTHOR: John GALLET
##### Version: 2008-JUNE-21
##### Latest: http://www.saphirtech.fr/
##### Status: It Works For Me (tm)
###############################################################################
###########
# Spam is legal in France !
body FR_SPAMISLEGAL                     /\b(Conform.+ment|En
vertu).{0,5}(article.{0,4}34.{0,4})?la loi\b/i
describe FR_SPAMISLEGAL                 French: pretends spam is (l)awful.
lang fr describe FR_SPAMISLEGAL     Invoque la loi informatique et
libertes.
score FR_SPAMISLEGAL                    2.5

body FR_SPAMISLEGAL_2                   /\bdroit d.acc.+s.{1,3}(de
modification)?.{0,5}de rectification\b/i
describe FR_SPAMISLEGAL_2               French: pretends spam is (l)awful.
lang fr describe FR_SPAMISLEGAL_2   Invoque le droit de rectification
cnil.
score FR_SPAMISLEGAL_2                  2.5

#####
# yeah, sure.
body FR_NOTSPAM                         /\b(ceci|ce).{1,9} n.est
pas.{1,5}spam\b/i
describe FR_NOTSPAM                     French: claims not to be spam.
lang fr describe FR_NOTSPAM     Affirme ne pas etre du spam.
score FR_NOTSPAM                        4.0

#####
## I can pay my taxes
body FR_PAYLESSTAXES
/\b(paye|calcul|simul|r.+dui|investi).{1,7}(moins|vo|ses).{0,5}imp.+t(s)?\
b/i
describe FR_PAYLESSTAXES                French: Pay less taxes
lang fr describe FR_PAYLESSTAXES    Simulateurs et reductions d'impots.
score FR_PAYLESSTAXES                   3.0

body FR_REALESTATE_INVEST               /\b(loi)?
(de.robien|girardin).{1,15}(neuf|recentr.+|ancien|IR|IS|imp.+t(s)?|industr
iel(le)?)\b/i
describe FR_REALESTATE_INVEST           French: Invest in real-estate with
tax-reductions
lang fr describe FR_REALESTATE_INVEST   Reduction impots immobilier.
score FR_REALESTATE_INVEST              2.5
```

```
#####
# I won at the casino
body FR_ONLINEGAMBLING                  /\b(casino(s)?|jeu(x)?|joueur(s)?)
(en ligne|de grattage)\b/i
describe FR_ONLINEGAMBLING              French: Online gambling
lang fr describe FR_ONLINEGAMBLING  Jeux en ligne.
score FR_ONLINEGAMBLING                 3.0

#####
# Baby, did you forget to take your meds ?
body FR_ONLINEMEDS                      /\bpharmacie(s)? (en
ligne|internet)\b/i
describe FR_ONLINEMEDS                  French: Online meds ordering
lang fr describe FR_ONLINEMEDS      Achat de medicaments en ligne.
score FR_ONLINEMEDS                     3.0

######
# Tell me why
body FR_REASON_SUBSCRIBE                /\bVous recevez ce(t|tte)?
(message|mail|m.+l|lettre|news.+) (car|parce que)\b/i
describe FR_REASON_SUBSCRIBE            French: you subscribed to my spam.
lang fr describe FR_REASON_SUBSCRIBE    Indique pourquoi vous recevez le
courrier.
score FR_REASON_SUBSCRIBE               1.5

#####
# How to unsubscribe
body FR_HOWTOUNSUBSCRIBE
/\b(souhaitez|d.+sirez|pour).{1,10}(plus.{1,}recevoir|d.+sincrire|d.+sinsc
ription|d.+sabonner).{0,10}(information|email|mail|mailing|newsletter|lett
re|liste|message|offre|promotion|programme)(s)?\b/i
describe FR_HOWTOUNSUBSCRIBE            French: how to unsubscribe
lang fr describe FR_HOWTOUNSUBSCRIBE    Indique comment se desabonner.
score FR_HOWTOUNSUBSCRIBE               2.0

####
# Various "CRM" (Could Remove Me)
#####
header FR_MAILER_1                      X-Mailer =~
/(delosmail|cabestan|ems|mp6|wamailer|phpmailer|eMailink|Accucast|Benchmai
l)/i
describe FR_MAILER_1                    French spammy X-Mailer
lang fr describe FR_MAILER_1        X-Mailer couramment employe pour des
spams en francais.
score FR_MAILER_1                       2.0

header FR_MAILER_2                      X-EMV-CampagneId =~ /.+/
describe FR_MAILER_2                    French spammy mailer header
```

```
lang fr describe FR_MAILER_2        X-Mailer couramment employe pour des
spams en francais.
score FR_MAILER_2                       2.0

###########################################################################
###########
##### END FRENCH SPECIFIC SPAMASSASSIN RULES.
###########################################################################
###########
```

Pour les utiliser, copier les simplement à la fin de votre fichier `/etc/spamassassin/local.cf`. Je les ai trouvé ici, et j'ai baissé la valeur des scores, histoire de ne pas avoir trop de *faux-positifs*. Libre à vous d'utiliser où non ces règles qui n'ont rien d'officielles.

Remerciements

Administrer un serveur email est à la portée de tout le monde (j'ai appris à le faire sans aucune formation), mais réclame du temps, et bien souvent de l'aide de la part d'utilisateurs expérimentés. J'ai monté mon premier serveur de courrier en 2006, et la configuration que je vous présente ici est une descendante, largement améliorée, mais sans doute pas parfaite, de celle que j'utilisais à l'époque.

Je tiens à remercier ici Yannick, qui m'a véritablement pris par la main pour monter ce premier serveur de courrier, Wietse Venema, l'auteur de Postfix, et qui parmi les développeurs de logiciel de cette importance, est certainement l'un des plus accessible, gentil, et actif sur la mailing-liste de support. Un petit mot pour Pierre aussi, qui m'a donné quelques tips utiles.

Linux, Serveur, Mail, Sécurité

1) Dans cette documentation, nous reprenons les terminologie directement issue de la configuration des logiciels, en anglais, pour vous faciliter la vie lorsque vous aurez à chercher sur internet de la documentation ou de l'aide. Oui, vous en aurez très certainement besoin

2) Bien entendu, les mots de passes utilisés ici ne sont là qu'a titre d'exemples, remplacez les par des mots de passes forts

3) serveur de mail par défaut de Debian

4) Vous devez au préalable avoir configuré Apache pour cela

5) le principal fichier de configuration de Postfix

6) D'ailleurs, après vérification, en octobre 2013, sur ce serveur, postscreen rejette plus de 96% des connexions…

7) On à le droit de parler de viagra, même quand on est pas un spammeur

8) le fichier `master.cf` complet présenté plus haut contient déjà ces lignes

9)

par défaut, ce score est à 0, je crois. Chez nous, ce score est à -50, pour avoir les en-têtes y compris sur les courriers légitimes

10)

en version 0.7.2 dans Debian stable. Personellement, j'ai backporté la version 0.9.4 depuis Sid