Tester notre serveur de méls

23 mai 2001

Table des matières

1 Tester notre serveur de méls

Bon ça y'est, on a un serveur de méls configuré, on voudrait bien le tester maintenant, pour être sûr qu'il fonctionne correctement.

1.1 Lancer sendmail

Il faut commencer par lancer le démon sendmail (*-bd*) et lui dire l'intervalle de temps entre chaque scrutation de la file d'attente (*-q10m* pour **10** mn, par exemple) :

```
/usr/sbin/sendmail -bd -q10m
```

Si on veut que `sendmail` soit lancé par `inetd`, il faut mettre une entrée du style :

```
smtp stream tcp nowait nobody /usr/sbin/sendmail -bs
```

L'option *-bs* précise qu'il doit utiliser SMTP pour les entrées/sorties. Il faut alors lancer `runq` (qui lance `sendmail -q`, en fait) pour vider une fois la file d'attente, on mettre une entrée de ce style dans la `crontab` :

```
# Vide la file d'attente du mel toute les 15 minutes
0,15,30,45 * * * * /usr/bin/runq
```

1.2 Configurer syslogd

La première étape consiste à configurer `syslogd`, qui permet de gérer les messages produits par les démons. Pour cela, il faut juste vérifier que le fichier `/etc/syslog.conf` contient des entrées permettant de générer des messages :

```
mail.debug          /var/log/mail/info
mail.err            /var/log/mail/errors
```

Cela indique quels sont les fichiers recevant les messages (`/var/log/mail/info`) et les erreurs (`/var/log/mail/errors`).

Voici à quoi ressemble ces fichiers :

```
Jan  2 14:31:33 terre sendmail[1341]: f02DVXH01341: from=root, size=37,
class=0, nrcpts=1, msgid=<200101021331.f02DVXH01341@terre.mil.def>,
relay=root@localhost
Jan  2 14:31:33 terre sendmail[1343]: f02DVXH01341: to=mdecore,
ctladdr=root (0/0), delay=00:00:00, xdelay=00:00:00, mailer=local,
pri=30037, dsn=2.0.0, stat=Sent
```

On voit ici qu'un message a été envoyé par **root** sur la machine **terre** à l'utilisateur **mdecore**.

1.3 Compilation des bases de données

Makefile Différence : et pas :

1.4 Utiliser le mode test de sendmail

On peut maintenant tester la façon dont `sendmail` réécrit les adresses, et décide à quel *mailer* envoyer le message. Pour cela, on utilise l'option *-bt*.

On peut commencer par tester si les courriers locaux sont bien redistribués en utilisant le mailer local :

```
[mdecore@aline mdecore]$ sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 mdecore
canonify           input: mdecore
Canonify2          input: mdecore
Canonify2        returns: mdecore
canonify         returns: mdecore
parse              input: mdecore
Parse0             input: mdecore
Parse0           returns: mdecore
ParseLocal         input: mdecore
ParseLocal       returns: mdecore
Parse1             input: mdecore
Parse1           returns: $# local $: mdecore
parse            returns: $# local $: mdecore
> 3,0 mdecore@aline
canonify           input: mdecore @ aline
Canonify2          input: mdecore < @ aline >
Canonify2        returns: mdecore < @ aline . >
canonify         returns: mdecore < @ aline . >
parse              input: mdecore < @ aline . >
Parse0             input: mdecore < @ aline . >
Parse0           returns: mdecore < @ aline . >
ParseLocal         input: mdecore < @ aline . >
ParseLocal       returns: mdecore < @ aline . >
Parse1             input: mdecore < @ aline . >
Parse1           returns: $# local $: mdecore
parse            returns: $# local $: mdecore
>
```

ou, avec le kit-jussieu :

```
> 3,0 waves
rewrite: ruleset   3   input: waves
rewrite: ruleset  19   input: waves
rewrite: ruleset  19   input: waves < @ . LOCAL >
rewrite: ruleset  19 returns: waves < @ machine . division . domaine . fr . EXTERNE >
rewrite: ruleset  19 returns: waves < @ machine . division . domaine . fr . EXTERNE >
rewrite: ruleset   3 returns: waves < @ machine . division . domaine . fr . EXTERNE >
rewrite: ruleset   0   input: waves < @ machine . division . domaine . fr . EXTERNE >
rewrite: ruleset   0 returns: $# smtp $@ [ mailhost . division . domaine . fr ] $: waves < @ machine . division . domaine . fr >
> 3,0 mdecore@linux-france.org
rewrite: ruleset   3   input: mdecore @ linux-france . org
rewrite: ruleset  19   input: mdecore < @ linux-france . org >
rewrite: ruleset  19 returns: mdecore < @ linux-france . org . EXTERNE >
rewrite: ruleset   3 returns: mdecore < @ linux-france . org . EXTERNE >
rewrite: ruleset   0   input: mdecore < @ linux-france . org . EXTERNE >
rewrite: ruleset   0 returns: $# smtp $@ [ mailhost . division . domaine . fr ] $: mdecore < @ linux-france . org >
>
```

On constate que le mailer utilisé sera bien le mailer local (`returns : $# local`) dans les deux cas.

Voyons maintenant pour les autres machines de notre réseau :

```
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 postmaster@anais
canonify           input: postmaster @ anais
Canonify2          input: postmaster < @ anais >
Canonify2        returns: postmaster < @ anais >
canonify         returns: postmaster < @ anais >
parse              input: postmaster < @ anais >
Parse0             input: postmaster < @ anais >
Parse0           returns: postmaster < @ anais >
ParseLocal         input: postmaster < @ anais >
ParseLocal       returns: postmaster < @ anais >
Parse1             input: postmaster < @ anais >
Mailertable        input: < anais > postmaster < @ anais >
Mailertable      returns: postmaster < @ anais >
MailerToTriple     input: < > postmaster < @ anais >
MailerToTriple   returns: postmaster < @ anais >
Parse1           returns: $# esmtp $@ anais $: postmaster < @ anais >
parse            returns: $# esmtp $@ anais $: postmaster < @ anais >
>
```

Ce test montre que le courrier sera directement distribué à l'utilisateur **postmaster** de l'hôte **anais** via esmtp (`returns : $# esmtp $@ anais $ : postmaster < @ anais >`).

On peut enfin tester les adresses extérieures à notre réseau, en prenant celle chez notre fournisseur d'accès par exemple :

```
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 mdecore@free.fr
canonify           input: mdecore @ free . fr
Canonify2          input: mdecore < @ free . fr >
Canonify2        returns: mdecore < @ free . fr . >
canonify         returns: mdecore < @ free . fr . >
parse              input: mdecore < @ free . fr . >
Parse0             input: mdecore < @ free . fr . >
Parse0           returns: mdecore < @ free . fr . >
ParseLocal         input: mdecore < @ free . fr . >
ParseLocal       returns: mdecore < @ free . fr . >
Parse1             input: mdecore < @ free . fr . >
Mailertable        input: < free . fr > mdecore < @ free . fr . >
Mailertable        input: free . < fr > mdecore < @ free . fr . >
Mailertable      returns: mdecore < @ free . fr . >
Mailertable      returns: mdecore < @ free . fr . >
MailerToTriple     input: < > mdecore < @ free . fr . >
MailerToTriple   returns: mdecore < @ free . fr . >
Parse1           returns: $# esmtp $@ free . fr . $: mdecore < @ free . fr . >
parse            returns: $# esmtp $@ free . fr . $: mdecore < @ free . fr . >
>
```

Le résultat ne diffère pas de l'exemple précédent, le serveur ESMTP du domaine **free.fr** sera contacté pour distribuer le courrier à l'utilisateur **mdecore**.

Si on a configuré notre réseau de telle sorte que tous les autres domaines que le notre ne soient pas reconus, on passe par le serveur du fournisseur d'accès. Dans ce cas, on aura comme sortie :

```
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 postmaster@unautredomaine.com
canonify           input: postmaster @ unautredomaine . com
Canonify2          input: postmaster < @ unautredomaine . com >
Canonify2        returns: postmaster < @ unautredomaine . com . >
canonify         returns: postmaster < @ unautredomaine . com . >
```

```
parse              input: postmaster < @ unautredomaine . com . >
Parse0             input: postmaster < @ unautredomaine . com . >
Parse0           returns: postmaster < @ unautredomaine . com . >
ParseLocal         input: postmaster < @ unautredomaine . com . >
ParseLocal       returns: postmaster < @ unautredomaine . com . >
Parse1             input: postmaster < @ unautredomaine . com . >
MailerToTriple     input: < smtp8 : [ mail . fai . fr ] > postmaster
< @ unautredomaine . com . >
MailerToTriple   returns: $# smtp8 $@ [ mail . fai . fr ] $: postmaster
< @ unautredomaine . com . >
Parse1           returns: $# smtp8 $@ [ mail . fai . fr ] $: postmaster
< @ unautredomaine . com . >
parse            returns: $# smtp8 $@ [ mail . fai . fr ] $: postmaster
< @ unautredomaine . com . >
>
```

On constate ici que le serveur SMTP **mail.fai.fr** sera contacté pour redistribué le courrier à l'utilisateur **postmaster** du domaine **unautredomaine.com** (`returns : $# smtp8 $@ [ mail . fai . fr ] $ : postmaster < @ unautredomaine . com . >`). En fait, tous les noms de domaines non reconnus sont supposés être exérieurs à notre réseau, si bien qu'on considère qu'il faut passer par le fournisseur d'accès. Le nom de domaine peut très bien ne pas exister, il passera quand même par fai.

1.5 Utiliser le mode debug de sendmail

Enfin, on peut utiliser les modes debug de `sendmail` pour avoir des messages plus élaborés. Le premier mode qui nous intéresse permet de suivre l'envoi d'un courrier avec l'option *-d21.2* :

```
$ /usr/lib/sendmail -d21.2 mdecore
rewrite: ruleset canonify          input: root
rewrite: ruleset Canonify2         input: root
rewrite: ruleset Canonify2       returns: root
rewrite: ruleset canonify        returns: root
rewrite: ruleset parse             input: root
rewrite: ruleset Parse0            input: root
rewrite: ruleset Parse0          returns: root
rewrite: ruleset ParseLocal        input: root
rewrite: ruleset ParseLocal      returns: root
rewrite: ruleset Parse1            input: root
rewrite: ruleset Parse1          returns: $# local $: root
```

```
rewrite: ruleset parse            returns: $# local $: root
rewrite: ruleset final              input: root
rewrite: ruleset final            returns: root
rewrite: ruleset canonify           input: root
rewrite: ruleset Canonify2          input: root
rewrite: ruleset Canonify2        returns: root
rewrite: ruleset canonify         returns: root
rewrite: ruleset 1                  input: root
rewrite: ruleset 1                returns: root
rewrite: ruleset final              input: root
rewrite: ruleset final            returns: root
rewrite: ruleset canonify           input: mdecore
rewrite: ruleset Canonify2          input: mdecore
rewrite: ruleset Canonify2        returns: mdecore
rewrite: ruleset canonify         returns: mdecore
rewrite: ruleset parse              input: mdecore
rewrite: ruleset Parse0             input: mdecore
rewrite: ruleset Parse0           returns: mdecore
rewrite: ruleset ParseLocal         input: mdecore
rewrite: ruleset ParseLocal       returns: mdecore
rewrite: ruleset Parse1             input: mdecore
rewrite: ruleset Parse1           returns: $# local $: mdecore
rewrite: ruleset parse            returns: $# local $: mdecore
rewrite: ruleset 2                  input: mdecore
rewrite: ruleset 2                returns: mdecore
rewrite: ruleset EnvToL             input: mdecore
rewrite: ruleset EnvToL           returns: mdecore
rewrite: ruleset final              input: mdecore
rewrite: ruleset final            returns: mdecore
rewrite: ruleset localaddr          input: mdecore
rewrite: ruleset Local_localaddr    input: mdecore
rewrite: ruleset Local_localaddr  returns: mdecore
rewrite: RHS $&h => ""
rewrite: ruleset localaddr        returns: mdecore
$
```

On voit la sélection de l'agent de transport, la réécriture de l'adresse de l'expéditeur (**root**) et du destinataire (*mdecore*).

On peut aussi utiliser l'option *-d21.15* pour voir en détails la façon dont l'algorithme réécrites les adresses :

```
$ /usr/lib/sendmail -d21.15 mdecore
```

```
rewrite: ruleset canonify           input: root
-----trying rule (line 507): $@
----- rule fails
-----trying rule (line 510): $*
-----rule matches: $: $1 < @ >
$1: bfffd8ec="root"
rewritten as: root < @ >
-----trying rule (line 511): $* < $* > $* < @ >
----- rule fails
-----trying rule (line 512): @ $* < @ >
----- rule fails
-----trying rule (line 513): $* : : $* < @ >
----- rule fails
-----trying rule (line 514): : include : $* < @ >
----- rule fails
-----trying rule (line 515): $* [ IPv6 $- ] < @ >
----- rule fails
-----trying rule (line 516): $* : $* [ $* ]
----- rule fails
-----trying rule (line 517): $* : $* < @ >
----- rule fails
-----trying rule (line 518): $* < @ >
-----rule matches: $: $1
$1: bfffd8ec="root"
rewritten as: root
-----trying rule (line 519): $* ;
----- rule fails
-----trying rule (line 520): $* < $* ; >
----- rule fails
-----trying rule (line 523): $@
----- rule fails
-----trying rule (line 526): $*
-----rule matches: $: < $1 >
$1: bfffd8ec="root"
rewritten as: < root >
-----trying rule (line 527): $+ < $* >
----- rule fails
-----trying rule (line 528): < $* > $+
----- rule fails
-----trying rule (line 529): < >
----- rule fails
```

```
-----trying rule (line 530): < $+ >
-----rule matches: $: $1
$1: bfffd8ec="root"

...

rewritten as: root
rewrite: ruleset canonify         returns: root
rewrite: ruleset 1                  input: root
rewrite: ruleset 1                returns: root
rewrite: ruleset final              input: root
-----trying rule (line 613): $* < @ >
----- rule fails
-----trying rule (line 616): $* < @ $+ . > $*
----- rule fails
-----trying rule (line 619): $* < @ *LOCAL* > $*
----- rule fails
-----trying rule (line 622): $* < $+ > $*
----- rule fails
-----trying rule (line 623): @ $+ : @ $+ : $+
----- rule fails
-----trying rule (line 624): @ $*
----- rule fails
-----trying rule (line 627): $+ @ $- . UUCP
----- rule fails
-----trying rule (line 630): $+ % $=w @ $=w
----- rule fails
rewrite: ruleset final            returns: root
rewrite: ruleset canonify           input: mdecore
-----trying rule (line 507): $@
----- rule fails
-----trying rule (line 510): $*
-----rule matches: $: $1 < @ >
$1: bfffdfec="mdecore"
rewritten as: mdecore < @ >
-----trying rule (line 511): $* < $* > $* < @ >
----- rule fails
-----trying rule (line 512): @ $* < @ >
----- rule fails
-----trying rule (line 513): $* : : $* < @ >
----- rule fails
```

```
-----trying rule (line 514): : include : $* < @ >
----- rule fails
-----trying rule (line 515): $* [ IPv6 $- ] < @ >
----- rule fails
-----trying rule (line 516): $* : $* [ $* ]
----- rule fails
-----trying rule (line 517): $* : $* < @ >
----- rule fails
-----trying rule (line 518): $* < @ >
-----rule matches: $: $1
$1: bfffdfec="mdecore"

...

rewritten as: < > mdecore
-----trying rule (line 744): < > $+
-----rule matches: $: < > < $1 < > $&h >
$1: bfffdd9c="mdecore"
rewrite: RHS $&h => ""
rewritten as: < > < mdecore < > >
-----trying rule (line 745): < > < $+ < > + $* >
----- rule fails
-----trying rule (line 746): < > < $+ < > $* >
-----rule matches: $: < > < $1 >
$1: bfffdd9c="mdecore"
rewritten as: < > < mdecore >
-----trying rule (line 747): < > < $+ + $* > $*
----- rule fails
-----trying rule (line 748): < > < $+ > + $*
----- rule fails
-----trying rule (line 749): < > < $+ >
-----rule matches: $@ $1
$1: bfffdd9c="mdecore"
rewritten as: mdecore
rewrite: ruleset localaddr        returns: mdecore
$
```

1.6 Tester le mailer SMTP

Pour voir la façon dont `sendmail` dialogue avec le serveur SMTP local, on peut l'appeller disectement pour envoyer le message, et mettre l'option

-v :

```
$ /usr/lib/sendmail -v mdecore@multimania.com
From: mdecor@division.domaine.fr
To: mdecore@multimania.com

Petit essai
mdecore@multimania.com... Connecting to machine.division.domaine.fr. via smtp...
220 machine.division.domaine.fr ESMTP Sendmail 8.11.1/jtpda-5.2 ready at
Tue, 2 Jan 2001 14:40:00 +0100 (CET)
>>> EHLO machine.division.domaine.fr
250-machine.division.domaine.fr Hello mdecor@machine.division.domaine.fr [193.51.26.5],
pleased to meet you
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-SIZE
250-DSN
250-ONEX
250-ETRN
250-XUSR
250 HELP
>>> MAIL From:<mdecor> SIZE=65
250 2.1.0 <mdecor>... Sender ok
>>> RCPT To:<mdecore@multimania.com>
250 2.1.5 <mdecore@multimania.com>... Recipient ok
>>> DATA
354 Enter mail, end with "." on a line by itself
>>> .
250 2.0.0 f02De0t09353 Message accepted for delivery
mdecore@multimania.com... Sent (2.0.0 f02De0t09353 Message accepted
for delivery)
>>> QUIT
221 2.0.0 machine.division.domaine.fr closing connection
$
```

Il suffit juste de mettre les deux en-têtes ***From :*** et ***To :*** suivies du corps du message.

1.7 Tester sendmail sans perturber l'envoi et la réception des messages

Noter qu'on peut spécifier quel fichier de configuration utiliser avec l'option *-C*, ce qui permet de tester un fichier sans perturber l'envoi de messages des utilisateurs. Pour être encore plus tranquille, le mieux est encore de lancer un serveur SMTP sur un port différent du port habituel, afin de ne pas perturber l'envoi la réception de messages. Pour lancer `sendmail` sur le port **26**, faire :

```
$ /usr/lib/sendmail -C $HOME/sendmail.cf.new -bd -oOport=26
$ telnet localhost 26
Trying 127.0.0.1...
...
```

1.8 Vérifier les en-têtes

En dernier test, il faut vérifier les en-têtes tel qu'ils seront lu par le destinataire du message. On pourrait simplement s'envoyer un message à soi même, mais le message sera réécrit deux fois. Pour résoudre ce problème, il existe des adresses spéciales, qui s'appellent des réflecteurs.

Voici ce que donne le message reçu :

```
------ Le serveur echo du domaine cnam.fr
------ a recu votre message le ven 29 déc 16:37:09 MET 2000

------ Ci-dessous les en-tetes et le corps de votre message


> From mdecor@division.domaine.fr  Fri Dec 29 16:37:09 2000
> Received: from mailhost.domaine.fr (root@mailhost.domaine.fr [193.51.24.1])
>         by fermi.cnam.fr (8.8.8/jpm-301097)  with ESMTP id QAA18424
>         for <echo@cnam.fr>; Fri, 29 Dec 2000 16:37:08 +0100 (MET)
> Return-Path: <mdecor@division.domaine.fr>
> Received: from machine.division.domaine.fr (machine.division.domaine.fr [193.51.26.1])
>           by mailhost.domaine.fr (8.11.1/jtpda-5.3.3) with ESMTP id eBTFb8E99947
>           for <echo@cnam.fr>; Fri, 29 Dec 2000 16:37:08 +0100 (CET)
> Received: from alsace.division.domaine.fr (alsace.division.domaine.fr [193.51.26.6])
>           by machine.division.domaine.fr (8.11.1/jtpda-5.2) with ESMTP id eBTFb7t84750
>           for <echo@cnam.fr>; Fri, 29 Dec 2000 16:37:07 +0100 (CET)
> From: mdecor@division.domaine.fr (Mathieu DECORE)
> Received: from (mdecor@localhost)
```

```
>           by alsace.division.domaine.fr (8.11.1/jtpda-5.2) id eBTFb7L02344
>           for echo@cnam.fr; Fri, 29 Dec 2000 16:37:07 +0100 (CET)
> Date: Fri, 29 Dec 2000 16:37:07 +0100 (CET)
> Message-Id: <200012291537.eBTFb7L02344@alsace.division.domaine.fr>
> To: echo@cnam.fr
> Subject: Test

------

>
> Le test...
>

------ Fin de votre message
```

1.9 En cas de problème...

Il y a quelque petite manipulations qu'on peut effectuer si tout ne marche pas comme on le voudrait :

- si le domaine courant refuse de reçevoir des méls, changer la ligne :
  ```
  # my official domain name
  # ... define this only if sendmail cannot automatically determine
  # your domain
  #Dj$w.Foo.COM
  ```
  du fichier `/etc/sendmail.cf` en précisant le nom du domaine :
  ```
  Djwapmove.com
  ```
- si le domaine courant refuse de reçevoir des méls alors que tous les hôtes virtuels l'acceptent, insérer la ligne :
  ```
  Cwwapmove.com
  ```
  dans le fichier `/etc/sendmail.cf` en précisant le nom du domaine ;
- si on veut que `sendmail` ne masque que adresses d'expéditeur des méls sortants ; supprimer les régles locales **S10** et **S30** dans le fichier `/etc/sendmail.cf` ;
- si le *mailer* a un comportement un peu bizard, on peut le remplacer par le script shell suivant :
  ```
  #!/bin/sh

  exec >> /tmp/mail.log
  echo ----------------------------
  date
  echo arguments :
  ```

```
for i ; do echo $i ; done
echo message :
cat
exit 0
```

et remplacer le *mailer* douteux en remplacant la ligne :

```
Mlocal,         P=/usr/bin/procmail, F=lsDFMAw5:/|@qSPfhn09, S=EnvFromL/HdrFromL
                T=DNS/RFC822/X-Unix,
                A=procmail -a $h -d $u
```

du fichier `/etc/sendmail.cf` par :

```
Mlocal,         P=/tmp/monmailer.sh, F=lsDFMAw5:/|@qSPfhn09, S=EnvFromL/HdrFromL
                T=DNS/RFC822/X-Unix,
                A=procmail -a $h -d $u
```