

Servidor de correo con `sendmail` y SuSE

Luis Rial, luisrial@iies.es

v0.2, 15 Octubre 2002

En este artículo vamos a describir cómo poner en marcha un servidor de correo sobre SuSE Linux utilizando el programa `sendmail`. Los ejemplos e indicaciones recogidas se han probado sobre las versiones 6.3, 7.0 y 7.2 de SuSE.

1. Copyright

Copyright (c) 2002 LUIS RIAL. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la GNU Free Documentation License, Version 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation; Una copia de la GNU Free Documentation License se puede encontrar en <http://www.gnu.org/copyleft/>

2. Introducción

No cabe duda que, uno de los servicios más utilizados en Internet es el de correo electrónico. Para poder prestar este servicio son necesarios los servidores de correo.

Quizá, una de las primeras cosas de las que habría que hablar, es del distinto tipo de servidores de correo que nos podemos encontrar, o de los distintos cometidos que un servidor `SMTP` puede tener.

No es lo mismo un servidor con dominio propio que sin él. En el segundo caso, lo más probable es que sólo lo utilicemos para enviar nuestros mensajes, utilizando otra aplicación como `fetchmail` para recoger, desde el servidor de nuestro ISP (proveedor de acceso y/o servicios de Internet) los mensajes que nos envíen. También, en este caso, es importante hacer notar las limitaciones que nos podemos encontrar para el envío. Así, lo más probable es que nuestro ISP nos fuerce una sesión autenticada (se nos solicitaría `login` y `password`) o nos imponga el acceso através de sus recursos de acceso (es decir no podríamos acceder a Internet através de otro ISP y utilizarle a él para enviar nuestros mensajes).

Resumiendo, podríamos plantear los siguientes escenarios:

- Servidor sin dominio propio, utilizando el de nuestro ISP para recoger los mensajes que nos envíen, utilizándolo, a su vez, como *smarthost* para los que enviamos.
- Servidor con dominio propio. En este caso, lo habitual es tener conexión permanente a Internet, IP fija y pública, y nuestro servidor dado de alta como servidor de correo (registro MX correspondiente) para nuestro dominio en el correspondiente DNS.

En este documento nos centraremos en el segundo supuesto.

2.1 ¿Qué es un *mail hub*?

Un servidor de correo centralizado (denominado tradicionalmente como *mail hub* en inglés) se encarga de la gestión del correo entrante y saliente de su dominio.

Así, en un dominio (susehispano.org por ejemplo) podemos tener varios servidores de correo: uno por país, por ejemplo; o varias máquinas que, sin ser su cometido principal servir correo, corren `sendmail` para el envío de mensajes (un servidor web por ejemplo).

La utilización de un servidor de correo centralizado reporta varios beneficios:

- el resto de los servidores que corren `sendmail` pueden tener una configuración más sencilla;

- todo el correo con destino al dominio, es enviado al *mail hub*, el cual se encarga de la posterior distribución interna del mismo;
- todo el correo saliente del dominio es enviado por los distintos servidores al *mail hub*, que se encargará de su entrega al servidor de correo del dominio del destinatario;

También existen ciertos aspectos que pueden no resultar tan ventajosos:

- si nuestro dominio soporta mucho tráfico de correo entrante y/o saliente, todo es gestionado de forma centralizada por el *mail hub*, pudiendo producirse retrasos indeseados en la recepción y envío de los mensajes;
- por otro lado es necesario contar con un repositorio global de los usuarios de correo del dominio, algo así como un `/etc/passwd` compuesto por los `/etc/passwd` de cada servidor de correo;

Así, quizá el caso más completo que podríamos encontraros, es el de un *mail hub*, servidores *normales* por distintas ubicaciones para tratar el correo local, y un servidor con autenticación para el envío de correo desde fuera de nuestra ubicación habitual (desde casa, un hotel, etc.).

2.2 ¿Qué es un servidor de correo con autenticación?

Como regla general, un servidor de correo sólo acepta mensajes que:

- van dirigidos al dominio para el que está dado de alta en el correspondiente DNS, y además
- le es entregado por un servidor debidamente dado de alta en el DNS del dominio del remitente.

En los demás casos, el correo no es admitido.

Para conseguir excepciones a la regla anterior, existen los parámetros de `RELAY` que, en `sendmail` y **SuSE**, se configuran en `/etc/mail/access`. En líneas generales, en ese archivo incluimos reglas que permiten el `RELAY` (excepciones a la regla mencionada) a una máquina, un dominio, una subred, etc.

Más recientemente, existe la posibilidad de permitir el `RELAY` mediante *login* y *password*. Un servidor con esa característica, es un servidor de correo (`SMTP`) con autenticación (`AUTH`). En próximas revisiones de este documento intentaré tratar la configuración de un servidor de este tipo.

En la [figura 1](#) podemos ver la topología de la red de nuestro ejemplo. En ella, el servidor **smtp.susehispano.org** es el servidor central de correo (*mail hub*) mientras que **www.susehispano.org**, es un servidor web que precisa enviar correo.

A continuación vamos a ilustrar como configurar un servidor de correo centralizado (*mail hub*) utilizando `sendmail` sobre una distribución SuSE Linux (se ha probado sobre las versiones 6.3, 7.0 y 7.2). También veremos como configurar uno de los servidores (**www.susehispano.org**) del dominio que hará uso del *mail hub* anterior.

En SuSE Linux, al menos antes de la versión 8.0, para la configuración de `sendmail` son relevantes los siguientes archivos y variables del sistema:

- `/etc/mail/access`
- `/etc/mail/aliases`
- `/etc/mail/genericstable`
- `/etc/mail/virtusertable`
- `/etc/mail/mailertable`
- `variable sendmail_locahost`
- `variable from_header`
- `variable sendmail_smarthost`

*Nota: Para ilustrar los ejemplos de este documento hemos tomado **susehispano.org** como dominio de ejemplo. No obstante, las direcciones y configuraciones aquí mencionadas, nada tienen que ver con las realmente utilizadas en ese dominio.*

Por favor si detectas algún error a lo largo del documento, te agradecería me lo hicieras saber en la dirección de correo que aparece al principio. Gracias.

3. Configuración del *mail hub* con `sendmail`

El servidor de correo empleado en el ejemplo, que va a realizar las funciones de *mail hub*, es una máquina Linux con SuSE 6.3 y el paquete `sendmail` instalado.

Sus funciones son las de recibir todo el correo con destino al dominio **susehispano.org** y redirigirlo a las distintas máquinas que esperan recibir correo en este dominio.


```
localhost          1D IN A          127.0.0.1
www                1D IN A          10.168.68.252
                  1D IN MX         10 smtp.susehispano.org.
                  1D IN HINFO     ''Pentium III'' ''SuSE Linux 7.0''
                  1D IN TXT      ''Correo local susehispano.org''
smtp               1D IN A          10.168.68.253
                  1D IN MX         smtp.susehispano.org.
```

3.6 Variable `sendmail_locahost`

La variable del sistema mencionada arriba, debe tener el valor `localhost susehispano.org`. Esto permite a **smtp** quedarse y no rechazar los mensajes con destino al dominio **susehispano.org**. De no incluir lo anterior solo recibiría correos con destino a ella misma y que además se enviaran a `usuario@smtp.susehispano.org`. Esta variable se puede editar con la herramienta `yast` (sección *administración del sistema-cambiar archivo de configuración*).

3.7 Variable `from_header`

La variable del sistema mencionada arriba, debe tener el valor `susehispano.org`. De esta forma se enmascaran los remites de todos los correos, haciendo que aparezcan como `usuario@susehispano.org` en vez de como `usuario@smtp.susehispano.org` por ejemplo. Esta variable se puede editar con la herramienta `yast` (sección *administración del sistema-cambiar archivo de configuración*).

4. Configuración de `sendmail` con *mail hub* externo

La máquina **www.susehispano.org** tiene usuarios y/o aplicaciones que necesitan recibir/enviar correo. La configuración de `sendmail` con SuSE 7.2 es la siguiente:

4.1 Archivo `/etc/mail/access`

Nuevamente tenemos una línea del tipo:

```
10.168.68 RELAY
```

4.2 Archivo `/etc/mail/aliases`

De la misma forma que en la configuración del anterior servidor tenemos una línea de este tipo:

```
root: juan, \root
```

4.3 Archivo `/etc/mail/genericstable`

De nuevo, para evitar que aparezca el *hostname* en el remite de los correos, incluimos por cada usuario de la máquina **www.susehispano.org** una línea del estilo:

```
juan: juan@susehispano.org
```

4.4 Archivo `/etc/mail/virtusertable`

Este archivo no hay que modificarlo respecto al original.

4.5 Archivo `/etc/mail/mailestable`

Este archivo no hay que modificarlo respecto al original.

4.6 Variable `sendmail_smarthost`

Esta variable, que se puede editar con `yast` como hemos comentado antes, tiene el valor `smtp.susehispano.org`. Esto hace que todo el correo saliente, se entregue a **smtp.susehispano.org** y que luego ella lo entregue al destinatario final.

4.7 Variable `from_header`

La variable del sistema mencionada arriba, debe tener el valor `susehispano.org`. De esta forma se enmascaran los remites de todos los correos.

5. Referencias

- Sendmail y fetchmail: Un servidor de correo local. <http://www.linuxfocus.org/Castellano/May2000/article130.html>
- Sendmail, By Bryan Costales & Eric Allman; ISBN 1-56592-222-0, 1050 pages. Second Edition, January 1997. O'Reilly & Associates

6. Listados

6.1 Servidor *mail hub*

Listado del archivo `/etc/mail/access`

```
# With this file you can control the access
# to your mailserver, example:
#
#   cyberspammer.com           550 We don't accept mai-l from spammers
#   okay.cyberspammer.com     OK
#   sendmail.org               OK
#   128.32                     RELAY
#
# Take a look at /usr/share/sendmail/README for a full description
127                           RELAY
10.168.68                     RELAY
```

Listado del archivo `/etc/mail/aliases`

```
# Copyright (c) 1997-1999 SuSE GmbH Nuernberg, Germany.
# Author: Florian La Roche <florian@suse.de>
#         Werner Fink      <werner@suse.de>
#
# The program 'newaliases' must be run after changing this file.

# It is probably best to not work as user root and redirect all
# email to 'root' to another account. Then you don't have to check
# for important email too often on the root account.
# The '\root' will make sure that email is also delivered to the
# root-account, but also forwarded to the user 'joe'.
root:                          admin, \root

# Basic system aliases that MUST be present.
postmaster:                    root
mailer-daemon:                 postmaster

# General redirections for pseudo accounts in /etc/passwd.
daemon:                         root
lp:                             root
news:                           root
uucp:                           root
games:                          root
man:                             root
at:                              root
postgres:                       root
mdom:                           root
amanda:                         root
ftp:                             root
wwwrun:                         root
squid:                          root
msql:                           root
gnats:                          root
nobody:                         root
```

```
# ''bin'' used to be in /etc/passwd
bin:      root

# Further well-known aliases for dns/news/ftp/mail/fax/web/gnats.
newsadm:  news
newsadmin: news
usenet:   news
ftpadm:   ftp
ftpadmin: ftp
ftp-adm:  ftp
ftp-admin: ftp
hostmaster: root
mail:     postmaster
postman:  postmaster
post_office: postmaster
# ''abuse'' is often used to fight against spam email
abuse:    postmaster
spam:     postmaster
faxadm:   root
faxmaster: root
webmaster: root
gnats-admin: root

# Majordomo can be used to have mailinglists on your site.
#majordomo: ''|usr/lib/majordomo/wrapper majordomo''
#owner-majordomo: root,
#majordomo-owner: root,

# sample entry for a majordomo mailing-list called ''test''
# read /usr/doc/packages/majordomo/README.linux for more information
# replace ''test'' with a new name and put the administrator into
# the ''owner-test'' alias instead of ''root''.
#
#test: ''|usr/lib/majordomo/wrapper resend -l test test-outgoing''
#test-outgoing: :include:/var/lib/majordomo/lists/test
#test-request: ''|usr/lib/majordomo/wrapper majordomo -l test''
#test-approval: owner-test,
#owner-test-outgoing: owner-test,
#owner-test-request: owner-test,
#owner-test: root,
#
# if you have bulk_mailer installed, you can replace the above
# ''test-outgoing'' line with the following:
#test-outgoing: ''|usr/bin/bulk_mailer owner-test@host.com
/var/lib/majordomo/li
# para que funcione el hub
pepe: pepe@www.susehispano.org
admin: admin@smtp.susehispano.org
```

Listado del archivo /etc/mail/genericstable

```
#
# map outgoing sender addresse from foo to bar@domain.com:
# foo      bar@domain.com
#
# admin    admin@susehispano.org
```

Listado del archivo /etc/mail/mailertable

```
# Copyright (c) 1997-1999 SuSE GmbH Nuernberg, Germany.
# Author: Florian La Roche <florian@suse.de>
#
# sendmail will look for all non-local email into this file to determine
# the transport way to the next host. the destination hostname is used
# to find an entry in this file.
#
# all uucp examples will use normal domain addressing for email.
# this should be used by nearly everyone today.
#
# this will send all email via uucp to an attached uucp host.
# a uucp server should have an entry for each attached uucp host.
#uuhost.domain.com      uucp-dom:uuhost
#
# to configure one uucp host that needs to send all non-local mail
# to a uucp-server called ''userver'', we just configure a smarthost entry:
#.      uucp-dom:userver
#
```

```

# hosts sending email should be running all the day. if other hosts
# are down, they can try in regular intervals to deliver email.
# if you want to work on a machine that is not turned on all the time,
# you can fetch email from the main email hub and send all outgoing
# email directly to your local email hub for further delivery.
# that is called a smarthost-entry:
#.                               smtp:mailhub.domain.com
#
# send all email for a special host to another host or to a specific IP:
#host.sub.org                    smtp:host.domain.com
#host.sub.org                    smtp:[192.168.0.1]
#
# send email for all hosts below .sub.org to another host:
#.sub.org                        smtp:host.domain.com
#
# send all email for a specific host to one local user called "foo":
#host.sub.org                    local:foo
#
www.susehispano.org              smtp:[www.susehispano.org]

```

6.2 Servidor sendmail con *mail hub* externo

Listado del archivo `/etc/mail/access`

```

# /etc/mail/access
#
# Author: Werner Fink <werner@suse.de>
#
# Description:
#
#   With this file you can control the access
#   to your mail server.
#
# Format:
#
#<email addr>                <keyword or ### text>
#<domain name>              <keyword or ### text>
#<network addr>             <keyword or ### text>
#
#           ^^^^^^^^^^^
#           (these are <TAB> stops)
#
# Network IP-addresses have to end on octet boundary, e.g. 127.0.0
# The right hand side '<keyword or ### text>' could be one of
# the keywords
#
#   OK           (accept mails even if other rules would reject them)
#   REJECT       (reject mails even if other rules would accept them)
#   RELAY        (relay this domain, implicit OK within other rules)
#   DISCARD      (mail are discard)
#
# or an '###' RFC 821 compliant error code and some text, e.g.
#
#   550 We don't accept mail from spammers
#
# Examples:
#
#cyberspammer.com          550 We don't accept mail from spammers
#sendmail.org              OK
#192.168                   RELAY
#
# Extensions:
#
#   See /usr/share/sendmail/README for the FEATURE 'blacklist_recipients'.
#
# Default for loop back is RELAY
127                        RELAY
10.168.68                 RELAY

```

Listado del archivo `/etc/mail/aliases`

```

# Copyright (c) 1997-1999,2000 SuSE GmbH Nuernberg, Germany.
# Author: Florian La Roche
#         Werner Fink      <werner@suse.de>
#

```

```
# The program 'newaliases' must be run after changing this file.

# It is probably best to not work as user root and redirect all
# email to 'root' to another account. Then you don't have to check
# for important email too often on the root account.
# The '\root' will make sure that email is also delivered to the
# root-account, but also forwarded to the user 'joe'.
root:      juan, \root

# Basic system aliases that MUST be present.
postmaster:  root
mailer-daemon: postmaster

# General redirections for pseudo accounts in /etc/passwd.
administrator:  root
daemon:         root
lp:            root
news:          root
uucp:          root
games:         root
man:           root
at:           root
postgres:     root
mdom:         root
amanda:       root
ftp:          root
wwwrun:       root
squid:        root
mysql:        root
gnats:        root
nobody:       root
# 'bin' used to be in /etc/passwd
bin:          root

# Further well-known aliases for dns/news/ftp/mail/fax/web/gnats.
newsadm:      news
newsadmin:    news
usenet:       news
ftpadm:       ftp
ftpadmin:     ftp
ftp-adm:      ftp
ftp-admin:    ftp
hostmaster:   root
mail:         postmaster
postman:      postmaster
post_office:  postmaster
# 'abuse' is often used to fight against spam email
abuse:        postmaster
spam:         postmaster
faxadm:       root
faxmaster:    root
webmaster:    root
gnats-admin:  root

# Majordomo can be used to have mailinglists on your site.
#majordomo:   ''|/usr/lib/majordomo/wrapper majordomo''
#owner-majordomo:  root,
#majordomo-owner:  root,

# sample entry for a majordomo mailing-list called 'test'
# read /usr/doc/packages/majordomo/README.linux for more information
# replace 'test' with a new name and put the administrator into
# the 'owner-test' alias instead of 'root'.
#
#test:        ''|/usr/lib/majordomo/wrapper resend -l test
test-outgoing''
#test-outgoing:  :include:/var/lib/majordomo/lists/test
#test-request:  ''|/usr/lib/majordomo/wrapper majordomo -l test''
#test-approval:  owner-test,
#owner-test-outgoing:  owner-test,
#owner-test-request:  owner-test,
#owner-test:      root,
#
# if you have bulk_mailer installed, you can replace the above
# 'test-outgoing' line with the following:
#test-outgoing:  ''|/usr/bin/bulk_mailer owner-test@host.com
/var/lib/majordomo/lists/test''
#
```

Listado del archivo /etc/mail/genericstable

```
# /etc/mail/genericstable
#
```

```
# Author: Werner Fink <werner@suse.de>
#
# Description:
#
# map outgoing sender addresses from the (unqualified) left hand side
# to the qualified addresses on the right hand side. The same types
# of addresses as for masquerading are looked up, i.e., only header
# sender addresses unless the allmasquerade and/or masquerade_envelope
# features are given (rc.config -> FROM_HEADER). Qualified addresses
# must have the domain part in the list of names given by the by the
# macro GENERICS_DOMAIN (rc.config -> SENDMAIL_GENERICS_DOMAIN).
#
# Format:
#
#user@uqghost          realuser@fqghost
#user                ^^^^^^^^^^^^^^^ realuser@fqghost
#
# (these are <TAB> stops)
juan                   juan@susehispano.org
```
