

# Sistema de correo virtual con postfix

Ken Nowack *Autor*

Ezra Gorman *Autor*

Zack Gilburd *Editor*

Scygro *Editor*

José Alberto Suárez López *Traductor*

John Christian Stoddart *Traductor*

José Luis Rivero *Traductor*

Fernando J. Pereda Garcimartín *Traductor*

Fernando M. Bueno *Traductor*

*Actualizado 4 de septiembre, 2006*

***La versión original de este documento fue actualizada 24 de agosto, 2007***

## 1. Introducción

Para la mayoría de los usuarios de Gentoo, con un simple cliente de correo y fetchmail bastaría. Sin embargo, si se está hospedando un dominio en el sistema, se necesitará un MTA (Agente de Transporte de Correo) de verdad. Y si se están hospedando varios dominios, definitivamente se necesita algo más robusto para manejar todo el correo de los usuarios. Este sistema fue diseñado para ser una elegante solución a este problema.

Un sistema de correo virtual tiene que ser capaz de manejar correo de numerosos dominios con diferentes usuarios sobre una variedad de interfaces. Esto presenta algunos problemas con los que tendremos que tratar. Por ejemplo, ¿qué ocurre si dos usuarios en diferentes dominios quieren usar el mismo nombre de usuario? Si proveemos acceso imap y smtp-auth, ¿cómo combinamos todos los demonios de autenticación en un solo sistema? ¿Cómo proporcionar seguridad para los numerosos componentes que componen el sistema? ¿Cómo lo maneja todo?

Este documento muestra cómo configurar un sistema capaz de manejar correo para todos los dominios que el equipo pueda manejar, soporte de usuarios virtuales que no necesiten cuentas de shell, que los nombres sean específicos de cada dominio, que los clientes se puedan autenticar por web, imap, smtp, y pop3 contra una base de datos, utilizar ssl para la capa de transporte segura, que tenga una interfaz web, que pueda manejar listas de correo para cualquiera de los dominios, y sea controlado por una bonita, central y simple base de datos mysql.

Hay gran variedad de formas para mantener un sistema de correo virtual. Con tantas opciones, cualquier otra puede ser más adecuada dependiendo de las necesidades que se tengan. Hay que considerar investigar <http://www.es.gmail.org/> y <http://www.exim.org/> para explorar otras opciones.

Los siguientes paquetes serán usados en este sistema: apache, courier-imap, courier-authlib, postfix, mod\_php, phpmyadmin, squirrelmail, cyrus-sasl, mysql, php, y mailman.

Hay que asegurarse de activar las siguientes variables USE en /etc/make.conf antes de compilar los paquetes: USE="mysql imap libwww maildir sasl ssl". De otra forma se necesitarán recompilar cosas para obtener soporte para todos los protocolos. Además, es una buena idea desactivar cualquier otra variable que tenga que ver con correo y red, como ipv6.

**Importante:** Esta guía ha sido escrita para postfix-2.0.x. Si se está usando postfix < 2 algunas de las variables en este documento serán diferentes. Se recomienda actualizar. Cualquier otro paquete incluido en esta guía también es sensible a los números de versión. Se aconseja la lectura de la documentación incluida en los paquetes si se tienen problemas con el sistema.

**Importante:** Este documento usa apache-1.3.x. Apache-2 se considera estable en portage. Pero de cualquier forma hay un cierto número de problemas con la integración php. Hasta que el soporte php no se considere estable en apache-2.0.x, esta guía continuará usando la versión 1.3.x.

**Importante:** Se necesita un nombre de dominio para ejecutar un servidor de correo público, o al menos un registro MX para un dominio. Lo ideal sería tener control sobre al menos dos dominios para beneficiarse del nuevo sistema de correo virtual.

**Importante:** Hay que asegurarse de que `/etc/conf.d/hostname` contiene el nombre de máquina correcto del servidor de correo. Para que tomen efecto los cambios hechos en este archivo ejecute `/etc/init.d/hostname restart`. Verifique que esté correcto el nombre de su máquina con `hostname`. También verifique que no hay entradas conflictivas en `/etc/hosts`.

**Nota:** Se recomienda la lectura de este documento completo y la familiaridad con todos los pasos antes de ponerse a instalarlo. Si se tienen problemas con alguno de los pasos, puede comprobarse la guía de problemas frecuentes al final de este documento. También, no todos los paquetes aquí referenciados son necesarios, esto es para un sistema muy flexible. Por ejemplo, si no se desea una interfaz web, puede evitarse la sección de squirrelmail.

## 2. Postfix Básico

### Listado de Código 2.1: Instalar postfix

```
# emerge postfix
```

**Aviso:** Hay que verificar que no se ha instalado otro MTA, como `ssmtp`, `exim`, o `netqmail`, o seguro que se tendrán GRANDES problemas.

Tras que postfix sea instalado, es el momento de configurarlo. Vamos a cambiar las siguientes opciones en `/etc/postfix/main.cf`. Recuerde reemplazar las `$variables` con sus propios nombres.

### Listado de Código 2.2: `/etc/postfix/main.cf`

```
myhostname = $host.domain.name
mydomain = $domain.name
inet_interfaces = all
mydestination = $myhostname, localhost.$mydomain $mydomain
mynetworks = my.ip.net.work/24, 127.0.0.0/8
home_mailbox = .maildir/
local_destination_concurrency_limit = 2
default_destination_concurrency_limit = 10
```

Lo siguiente es cambiar `/etc/postfix/master.cf`. Esto activará que la salida sea detallada para poder depurar:

### Listado de Código 2.3: `/etc/postfix/master.cf`

```
# service type private unpriv chroot wakeup maxproc command + args
#          (yes)   (yes)   (yes)   (never) (50)
#
=====
(Sólo hay que añadir "-v" después de smtpd a la siguiente línea)
smtp      inet  n       -       n       -       -       smtpd -v
```

Lo siguiente es editar `/etc/mail/aliases` para añadir los alias locales. Debe haber al menos un alias para root como: `root: your@email.address`.

#### **Listado de Código 2.4: Iniciar postfix por primera vez**

```
# /usr/bin/newaliases
```

(Esto instalará el nuevo alias. Sólo se necesita hacer esto cuando se actuali

```
# /etc/init.d/postfix start
```

Ahora que postfix está corriendo, se ejecuta el cliente de correo de consola y nos enviamos un mensaje. Yo uso `mutt` para todo mi email. Comprobamos que postfix está repartiendo el correo a los usuarios locales, una vez esto esté hecho, pasaremos al siguiente paso.

**Nota:** Recomiendo que se verifique que esta configuración básica de postfix está funcionando antes de dar el siguiente paso en esta guía.

### **3. Courier-imap**

#### **Listado de Código 3.1: Instalar courier-imap**

```
# emerge courier-imap
```

#### **Listado de Código 3.2: Configuración de Courier-imap**

```
# cd /etc/courier-imap
```

(Si se quiere usar `ssl` para `imap` y `pop`, hay que crear certificados para este. Este paso está recomendado. Si no se quiere `ssl`, simplemente se puede evitar.)

```
# nano -w /etc/courier-imap/pop3d.cnf  
# nano -w /etc/courier-imap/imapd.cnf
```

(Se cambian los campos `C`, `ST`, `L`, `CN`, y `email` ajustándolos a nuestro caso.)

```
# mkpop3dcert  
# mkimapdcert
```

#### **Listado de Código 3.3: Inicie los servicios de courier que necesite.**

```
# /etc/init.d/courier-imapd start  
# /etc/init.d/courier-imapd-ssl start  
# /etc/init.d/courier-pop3d start  
# /etc/init.d/courier-pop3d-ssl start
```

Se ejecuta nuestro cliente de correo preferido y se comprueba que todas las conexiones para mandar y recibir correo funcionan. Ahora que lo básico funciona, vamos a seguir con el resto del manajo de paquetes para que el sistema funcione. Y otra vez, hay que comprobar que se ha instalado todo y que funciona antes de continuar.

### **4. Cyrus-sasl**

Ahora vamos a instalar `cyrus-sasl`. `Sasl` va a hacer la labor de pasar las variables de autenticación a `pam` que pasará esta información a `mysql` para autenticar usuarios de `smtp`. Para este documento no vamos a comprobar si `sasl` funciona

hasta que mysql no esté configurado correctamente con un usuario de prueba; lo que está bien, porque siempre autenticaremos contra una base de datos mysql.

#### **Listado de Código 4.1: Configurar e instalar el ebuild cyrus-sasl**

```
# emerge cyrus-sasl
```

Ahora, se edita /etc/sasl2/smtpd.conf.

#### **Listado de Código 4.2: Iniciar sasl**

```
# nano -w /etc/sasl2/smtpd.conf
mech_list: PLAIN LOGIN
pwcheck_method: saslauthd
# nano -w /etc/conf.d/saslauthd
SASLAUTHD_OPTS="${SASLAUTHD_MECH} -a rimap -r"
SASLAUTHD_OPTS="${SASLAUTHD_OPTS} -O localhost"
# /etc/init.d/saslauthd start
```

## **5. Certificados SSL para Postfix y Apache**

Ahora vamos a hacer los certificados SSL para postfix y apache.

#### **Listado de Código 5.1: Certificados ssl**

```
# cd /etc/ssl/  
# nano -w openssl.cnf
```

Hay que ajustar los siguientes valores por defecto para nuestro dominio:

```
countryName_default  
stateOrProvinceName_default  
localityName_default  
0.organizationName_default  
commonName_default  
emailAddress_default.
```

(Si las variables no están presentes, hay que añadirlas.)

```
# cd misc  
# nano -w CA.pl
```

Necesitamos añadir `-nodes` al `#` crear un certificado y `#` crear una petición de certificado para que nuestros nuevos certificados ssl sean cargados sin necesidad de una password. De otra forma no se podrán reiniciar los certificados ssl.

```
# crear un certificado  
system ("$REQ -new -nodes -x509 -keyout newreq.pem -out newreq.pem $DAYS");
```

```
# crear una petición de certificado  
system ("$REQ -new -nodes -keyout newreq.pem -out newreq.pem $DAYS");
```

```
# ./CA.pl -newca  
# ./CA.pl -newreq  
# ./CA.pl -sign  
# cp newcert.pem /etc/postfix  
# cp newreq.pem /etc/postfix  
# cp demoCA/cacert.pem /etc/postfix
```

Hacemos lo mismo para Apache

```
# openssl req -new > new.cert.csr  
# openssl rsa -in privkey.pem -out new.cert.key  
# openssl x509 -in new.cert.csr -out new.cert.cert -req -signkey new.cert.key
```

(Por ahora dejamos estos certificados ahí. Los agregaremos después de instala

## 6. Añadir a Postfix soporte para SSL y SASL

Ahora se edita el fichero de configuración de Postfix para que utilice ssl y sasl. Se añaden los siguientes parámetros al final del fichero para que serán más sencillos de encontrar.

**Listado de Código 6.1: /etc/postfix/main.cf**

```
# nano -w /etc/postfix/main.cf

smtpd_sasl_auth_enable = yes
smtpd_sasl2_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_sasl_local_domain =

(Las opciones broken_sasl_auth_clients y login y auth son para outlook y outl
smtpd_sasl_local_domain añade un dominio a los clientes que usen smtp-auth. H

smtpd_recipient_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_unauth_destination

(Las siguientes dos opciones activan el cifrado saliente)
smtp_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_use_tls = yes
#smtpd_tls_auth_only = yes
smtpd_tls_key_file = /etc/postfix/newreq.pem
smtpd_tls_cert_file = /etc/postfix/newcert.pem
smtpd_tls_CAfile = /etc/postfix/cacert.pem
smtpd_tls_loglevel = 3
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom

(smtpd_tls_auth_only está comentado para poder probar el sistema. Se puede ha

# postfix reload
```

Ahora vamos a comprobar que la configuración es aceptada por postfix.

### Listado de Código 6.2: Comprobar la configuración de ssl y tls

```
# telnet localhost 25

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mail.domain.com ESMTP Postfix
EHLO domain.com
250-mail.domain.com
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-STARTTLS
250-AUTH LOGIN PLAIN OTP
250-AUTH=LOGIN PLAIN OTP
250-XVERP
250 8BITMIME
^]
telnet> quit
```

Se comprueba que aparecen las líneas AUTH y STARTTLS. Como dije antes, tal y como está ahora AUTH no funcionará porque sasl intentará autenticar contra sasldb en lugar de hacerlo con el shadow, por alguna extraña razón, que nosotros no hemos configurado. Así que vamos a instalar mysql para que guarde la información sobre autenticación y la información de los dominios virtuales.

## 7. MySQL

Ahora vamos a instalar y configurar MySQL. Se necesitará el fichero genericmailsql.sql para este paso.

### Listado de Código 7.1: Instalar y configurar MySQL

```
# emerge mysql

# /usr/bin/mysql_install_db

Después de este comando hay que seguir los pasos en pantalla para
añadir un password para root en mysql, no mysqladmin, de otra forma la
base de datos estará abierta a todo el mundo.

# /etc/init.d/mysql start
# mysqladmin -u root -p create mailsql
# mysql -u root -p --opt mailsql < genericmailsql.sql

# mysql -u root -p mysql
mysql> GRANT SELECT,INSERT,UPDATE,DELETE
-> ON mailsql.*
-> TO mailsql@localhost
-> IDENTIFIED BY '$password';
Query OK, 0 rows affected (0.02 sec)

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

mysql> quit
```

(Hay que comprobar que el nuevo usuario (mailsql) puede conectar al servidor

```
# mysql -u mailsql -p mailsql
```

La nueva base de datos tiene algunos valores por defecto para dos dominios. Las siguientes tablas están incluidas:

- alias - información sobre los alias locales y los alias de mailman.
- relocated - mapas de direcciones de correo relocalizadas
- transport - transporte de correo por defecto para todos los dominios que se hospedan
- users - información sobre todos los usuarios
- virtual - información sobre los dominios virtuales

### Listado de Código 7.2: Ejemplo de tabla alias

```
id  alias  destination
1   root   foo@bar.com
2   postmaster  foo@bar.com
```

### Listado de Código 7.3: Ejemplo de tabla user

(Línea dividida para una mayor claridad.)

```
id email          clear    name      uid      gid      homedir   \  
  maildir                quota  postfix \  
10 foo@virt-bar.org $password realname virtid  virtid  /home/vmail \  
   /home/vmail/virt-bar.org/foo/.maildir/  y \  
13 foo@bar.com     $password realname localid localid /home/foo   \  
   /home/foo/.maildir/                       y
```

Los valores del uid y gid en virtid deben ser los del usuario y el grupo vmail.

#### Listado de Código 7.4: Ejemplo de tabla transport

```
id  domain          destination \  
1   bar.com         local: \  
2   virt-bar.org   virtual:
```

#### Listado de Código 7.5: Ejemplo de tabla virtual

```
id  email          destination \  
3   root@virt-bar.org other@email.address
```

## 8. Apache y phpMyAdmin

Ahora vamos a configurar apache y una interfaz para trabajar con la base de datos.

#### Listado de Código 8.1: Configurar apache y phpmyadmin

```
# emerge apache mod_php mod_ssl phpmyadmin
```

Hay muchas guías sobre cómo configurar apache con php. Como esta: <http://www.linuxguruz.com/z.php?id=31>. También hay muchos mensajes en <http://forums.gentoo.org> con información detallada sobre problemas comunes en la instalación (puede buscarse 'apache php'). Así que, no lo cubriremos aquí. Después de instalar y configurar apache y php, se puede continuar con esta guía. Ahora crearemos el .htaccess en el directorio que esté phpMyAdmin. Si no se hace esto, los motores de búsqueda entrarán e indexarán estas páginas y cualquiera podrá obtener información sobre la base de datos buscando en google, lo que es *¡MALO!*. Hay muchas guías acerca de esto, como por ejemplo: <http://www.csoft.net/docs/micro/htaccess.html.en>.

Ahora vamos a instalar los certificados que hicimos para Apache. Las directivas de Apache-SSL que se necesitan son las siguientes:

- SSLCertificateFile /path/to/certs/new.cert.cert
- SSLCertificateKeyFile /path/to/certs/new.cert.key

#### Listado de Código 8.2: Instalar los certificados SSL en Apache

```
# cp /etc/ssl/misc/new.cert.cert /etc/apache/conf/ssl/  
# cp /etc/ssl/misc/new.cert.key /etc/apache/conf/ssl/  
# nano -w /etc/apache/conf/vhosts/ssl.default-vhost.conf
```

Cambiamos los siguientes parámetros:

```
ServerName host.domain.name  
ServerAdmin your@email.address  
SSLCertificateFile /etc/apache/conf/ssl/new.cert.cert  
SSLCertificateKeyFile /etc/apache/conf/ssl/new.cert.key  
  
# /etc/init.d/apache restart
```

**Nota:** Si se tiene una instalación previa de Apache, se necesitará reiniciar completamente el servidor para instalar los nuevos certificados. Hay que comprobar los logs para verificar que Apache se reinició satisfactoriamente.

Ahora configuremos phpMyAdmin.

### Listado de Código 8.3: Configurar phpMyAdmin

```
# nano -w /home/httpd/htdocs/phpmyadmin/config.inc.php
```

Hay que cambiar los siguientes parámetros

```
$cfg['Servers'][$i]['host'] = 'localhost'; // MySQL hostname  
$cfg['Servers'][$i]['controluser'] = 'mailsql'; // MySQL información del usua  
// (este usuario debe tener acceso de solo  
$cfg['Servers'][$i]['controlpass'] = '$password'; // lectura a "mysql/user" y  
$cfg['Servers'][$i]['user'] = 'mailsql'; // Usuario MySQL  
$cfg['Servers'][$i]['password'] = '$password'; // Password MySQL
```

Ahora se accede a la página de phpmyadmin y navegamos por las tablas. Podremos añadir los alias locales, editar la tabla de usuarios para añadir un usuario de prueba, y añadir la información sobre los dominios. Por defecto, las tablas tienen unos valores que servirán de guía. Hay que asegurarse de poner la información correcta en la base de datos. Por ejemplo, hay que asegurarse de que los directorios de los usuarios locales existen y que tienen los valores uid/gid correctos. Los maildirs deberían ser creados automáticamente por postfix cuando el usuario recibe su primer mensaje. Así que, en general, es una buena idea mandar un mensaje de bienvenida tras crear los nuevos usuarios de forma que .maildir sea creado.

## 9. El usuario vmail

Llegados a este punto uno se pregunta cuál será el usuario bajo el que funcionen los usuarios virtuales. Así que configuremos eso ahora:

### Listado de Código 9.1: Añadir el usuario vmail

```
# adduser -d /home/vmail -s /bin/false vmail  
# uid=`cat /etc/passwd | grep vmail | cut -f 3 -d :`  
# groupadd -g $uid vmail  
# mkdir /home/vmail  
# chown vmail: /home/maile
```

Así que, cuando se creen cuentas de correo virtuales, hay que utilizar el directorio personal y el uid y gid de vmail. Cuando se creen cuentas locales hay que utilizar los gid, uid, y directorio personal correspondientes. Estuvimos pensando en crear una página php para administrar esto, pero aún no lo hemos hecho, dado que

phpMyAdmin funciona muy bien para esto.

## 10. Configurar la autenticación MySQL y con dominios virtuales

Ahora vamos a reconfigurar nuestra autenticación para que use la base de datos mailsq1 en courier-imap y en postfix. En todos los ejemplos que siguen, hay que reemplazar \$password por la contraseña que se escogió a la hora de crear el usuario mailsq1 en MySQL.

### Listado de Código 10.1: Configurar la autenticación

```
# nano -w /etc/courier/authlib/authdaemonrc
authmodulelist="authmysql authpam"
```

```
# nano -w /etc/courier/authlib/authmysqlrc
MYSQL_SERVER          localhost
MYSQL_USERNAME        mailsq1
MYSQL_PASSWORD        $password
MYSQL_DATABASE        mailsq1
MYSQL_USER_TABLE      users
```

(Hay que asegurarse de que la siguiente línea está comentada, dado que usamos

```
#MYSQL_CRYPT_PWFIELD crypt
MYSQL_CLEAR_PWFIELD  clear
MYSQL_UID_FIELD      uid
MYSQL_GID_FIELD       gid
MYSQL_LOGIN_FIELD    email
MYSQL_HOME_FIELD     homedir
MYSQL_NAME_FIELD     name
MYSQL_MAILDIR_FIELD  maildir
```

```
# /etc/init.d/courier-authlib restart
# /etc/init.d/saslauthd restart
```

¡Ya casi está! Ahora haremos las configuraciones necesarias para que postfix interactúe con la base de datos para sus demás necesidades de transporte. Recuerde reemplazar cada valor con el nombre del propio usuario, user id, contraseña, alias, dirección de correo electrónico, etc.

### Listado de Código 10.2: /etc/postfix/mysql-aliases.cf

```
# nano -w /etc/postfix/mysql-aliases.cf
# mysql-aliases.cf

user          = mailsq1
password      = $password
dbname        = mailsq1
table         = alias
select_field  = destination
where_field   = alias
hosts         = unix:/var/run/mysqld/mysqld.sock
```

### Listado de Código 10.3: /etc/postfix/mysql-relocated.cf

```
# nano -w /etc/postfix/mysql-relocated.cf
# mysql-relocated.cf

user          = mailsql
password      = $password
dbname        = mailsql
table         = relocated
select_field  = destination
where_field   = email
hosts         = unix:/var/run/mysqld/mysqld.sock
```

#### **Listado de Código 10.4: /etc/postfix/mysql-transport.cf (opcional)**

```
# nano -w /etc/postfix/mysql-transport.cf
# mysql-transport.cf

user          = mailsql
password      = $password
dbname        = mailsql
table         = transport
select_field  = destination
where_field   = domain
hosts         = unix:/var/run/mysqld/mysqld.sock
```

#### **Listado de Código 10.5: /etc/postfix/mysql-virtual-gid.cf (opcional)**

```
# nano -w /etc/postfix/mysql-virtual-gid.cf
# mysql-virtual-gid.cf

user          = mailsql
password      = $password
dbname        = mailsql
table         = users
select_field  = gid
where_field   = email
additional_conditions = and postfix = 'y'
hosts         = unix:/var/run/mysqld/mysqld.sock
```

#### **Listado de Código 10.6: /etc/postfix/mysql-virtual-maps.cf**

```
# nano -w /etc/postfix/mysql-virtual-maps.cf
# mysql-virtual-maps.cf

user          = mailsql
password      = $password
dbname        = mailsql
table         = users
select_field  = maildir
where_field   = email
additional_conditions = and postfix = 'y'
hosts         = unix:/var/run/mysqld/mysqld.sock
```

#### **Listado de Código 10.7: /etc/postfix/mysql-virtual-uid.cf (opcional)**

```
# nano -w /etc/postfix/mysql-virtual-uid.cf
# mysql-virtual-uid.cf

user          = mailsql
password      = $password
dbname        = mailsql
table         = users
select_field  = uid
where_field   = email
additional_conditions = and postfix = 'y'
hosts         = unix:/var/run/mysqld/mysqld.sock
```

#### **Listado de Código 10.8: /etc/postfix/mysql-virtual.cf**

```
# nano -w /etc/postfix/mysql-virtual.cf
# mysql-virtual.cf
user          = mailsql
password      = $password
dbname        = mailsql
table         = virtual
select_field  = destination
where_field   = email
hosts         = unix:/var/run/mysqld/mysqld.sock
```

Por último, hay que editar /etc/postfix/main.cf una vez más.

#### **Listado de Código 10.9: /etc/postfix/main.cf**

```
# nano -w /etc/postfix/main.cf

alias_maps = mysql:/etc/postfix/mysql-alias.cf
relocated_maps = mysql:/etc/postfix/mysql-relocated.cf

local_transport = local
local_recipient_maps = $alias_maps $virtual_mailbox_maps unix:passwd.byname

virtual_transport = virtual
virtual_mailbox_domains =
    virt-bar.com,
    $other-virtual-domain.com

virtual_minimum_uid = 1000
virtual_gid_maps = static:$vmail-gid
virtual_mailbox_maps = mysql:/etc/postfix/mysql-virtual-maps.cf
virtual_alias_maps = mysql:/etc/postfix/mysql-virtual.cf
virtual_uid_maps = static:$vmail-uid
virtual_mailbox_base = /
#virtual_mailbox_limit =
```

Por razones de seguridad es necesario cambiar los permisos de todos los /etc/mail/mysql-\*.cf:

#### **Listado de Código 10.10: Cambiar los permisos de los archivos**

```
# chmod 640 /etc/postfix/mysql-*.cf
# chgrp postfix /etc/postfix/mysql-*.cf
```

En Postfix 2.0.x, hay cambios significativos desde 1.1.x. Destacando que las tablas transport, virtual-gid y virtual-uid no son necesarias. Las tablas aún se incluyen, por si se quieren usar.

**Nota:** Se recomienda la lectura de VIRTUAL\_README incluido en la documentación de postfix para más información.

#### **Listado de Código 10.11: Hacer a postfix recargar sus tablas**

```
# postfix reload
```

Ahora, si todo ha ido bien, se debería tener el sistema funcionando. Los usuarios serán capaces de autenticarse contra la base de datos sql, usando su dirección de email completa, para pop3, imap, y smtp. Se recomienda comprobar que todo funciona correctamente en este momento. Si se tienen problemas, siempre puede comprobarse la sección de problemas frecuentes en esta guía.

## **11. Squirrelmail**

### **Listado de Código 11.1: Instalar squirrelmail**

```
# emerge squirrelmail
```

(Si se instala squirrelmail hacia localhost, puede accederse al mismo con http (Hay que sustituir 1.4.3a-r2 con la versión que se use)

```
# webapp-config -I -h localhost -d /mail squirrelmail 1.4.3a-r2
# cd /var/www/localhost/htdocs/mail/config
# perl ./conf.pl
```

(Hay que cambiar las opciones de la Organización, Servidor y Directorio en squirrelmail. Ahora se debería poder conectar a squirrelmail, con la dirección de correo completa, y utilizar el nuevo sistema de webmail).

## **12. Mailman**

Último paso: mailman. La nueva versión de mailman tiene una buena interfaz para soportar dominios virtuales, que es por lo que lo uso, y sin mencionar que es un muy buen paquete. Para instalar este paquete y hacer que funcione bien con dominios virtuales vamos a necesitar un poco de hacking. Yo recomiendo leer toda la documentación de mailman, incluyendo README.POSTFIX.gz, para entender qué se está haciendo aquí.

Otra nota más, las versiones actuales de mailman se instalan en /usr/local/mailman. Si, como yo, se quiere cambiar la localización por defecto de la instalación, puede modificarse el archivo ebuild, cambiando la variable INSTALLDIR.

### **Listado de Código 12.1: Instalar mailman**

```
# emerge mailman
```

### **Listado de Código 12.2: Configurar los valores predeterminados: Mailman/Defaults.py**

```
# nano -w /var/mailman/Mailman/Defaults.py
```

(Hay que cambiar los valores a continuación para reflejar el dominio primario

```
DEFAULT_EMAIL_HOST = 'domain.com'
DEFAULT_URL_HOST = 'www.domain.com'
```

### **Listado de Código 12.3: Configuración de mailman: mm\_cfg.py**

```
# nano -w /usr/local/mailman/Mailman/mm_cfg.py
MTA = "Postfix"
POSTFIX_STYLE_VIRTUAL_DOMAINS = ['virt-domain.com', 'virt.domain2.com']
add_virtualhost('www.virt.domain.com', 'virt.domain.com')
add_virtualhost('www.virt.domain2.com', 'virt.domain2.com')
```

(Esto es necesario para que funcionen los dominios virtuales.)

### **Listado de Código 12.4: Y por último, aunque no lo último**

(Cuando se haya terminado, añadiremos la primera lista.)

```
# su mailman
# cd ~
# bin/newlist test
Enter the email of the person running the list: your@email.address
Initial test password:
Hit enter to continue with test owner notification...
```

(Las listas de dominios virtuales tendrán que ser especificadas con el format

```
# bin/genaliases
```

(Ahora que los alias han sido regenerados, hay que verificar que han sido añ

```
# nano -w data/aliases
# STANZA START: test
# CREATED:
test:          "|/var/mailman/mail/mailman post test"
test-admin:   "|/var/mailman/mail/mailman admin test"
test-bounces: "|/var/mailman/mail/mailman bounces test"
test-confirm: "|/var/mailman/mail/mailman confirm test"
test-join:    "|/var/mailman/mail/mailman join test"
test-leave:   "|/var/mailman/mail/mailman leave test"
test-owner:   "|/var/mailman/mail/mailman owner test"
test-request: "|/var/mailman/mail/mailman request test"
test-subscribe: "|/var/mailman/mail/mailman subscribe test"
test-unsubscribe: "|/var/mailman/mail/mailman unsubscribe test"
# STANZA END: test

# /etc/init.d/mailman start
# rc-update add mailman default
```

(Hay que utilizar este comando para iniciar mailman en cada reinicio.)

### Listado de Código 12.5: Añadir soporte de alias mailman en postfix

```
# nano -w /etc/postfix/main.cf
owner_request_special = no
recipient_delimiter = +
```

(Puede leerse README.POSTFIX.gz para más detalles.)

```
alias_maps =
  hash:/var/mailman/data/aliases,
  mysql:/etc/postfix/mysql-aliases.cf

virtual_alias_maps =
  hash:/var/mailman/data/virtual-mailman,
  mysql:/etc/postfix/mysql-virtual.cf
```

(Esto añade el fichero de alias de mailman a postfix. Se pueden utilizar tablas de MySQL para esto, pero odio hacer esto a mano. Además, si no se están usando dominios virtuales, añadir los mapas de dominios virtuales a postfix causará problemas, avisado).

Ahora se deberían poder configurar listas de correo para cualquier dominio en la máquina. Como última nota sobre el asunto, hay que asegurarse de que se ejecutan todos los comandos de mailman con el usuario mailman (su mailman) o los permisos serán erróneos y habrá que corregirlos. Hay que leer la documentación de mailman para más información sobre cómo configurar y

manejar listas de correo con mailman.

## 13. Filtrando el contenido y Anti-Virus

Para filtrado de contenido y anti-virus, se puede consultar la [Guía Gentoo para el filtrado de correo](#).

## 14. Todo en uno

Ok, ya está todo, editamos `/etc/postfix/master.cf` y quitamos el modo detallado (verbose) para el uso en producción. Probablemente se quieran añadir los servicios al arranque del sistema para que se inicien cada vez que se reinicie el servidor. Hay que asegurarse de añadir todos los paquetes que se están usando - apache, mysql, saslauthd, postfix, courier-imapd, courier-imapd-ssl, courier-pop3d, y courier-pop3d-ssl, dependiendo de los tipos de acceso que se quieran ofrecer. Normalmente yo los inicio todos.

### Listado de Código 14.1: Todo en uno

```
# postfix reload
# rc-update add $service default
```

*¡A disfrutarlo!*

## 15. Problemas frecuentes

### Introducción

Lidiando con problemas: Esta es una pequeña guía para lidiar con los posibles problemas que puedan surgir para la configuración que hemos hecho. No es exhaustiva, pero puede ser un comienzo para empezar a pelearse con los problemas. Con una configuración complicada como esta, es necesario que se asigne correctamente el problema con el componente que está fallando. En general para hacer esto sirve con seguir los siguientes pasos. Empezar por la base del sistema y empezar a ascender hasta conseguir dar con el componente que tiene el problema.

### Paso 1: Comprobar los ficheros de configuración

Los errores al escribir un fichero de configuración son mortales, especialmente cuando se trata de sistemas de autenticación. Han de comprobarse los ficheros de configuración y sql en busca de estos errores. Se puede depurar todo lo que uno quiera, pero si no se le está pasando la información correcta, el sistema de correo no va a funcionar. Si se hace un cambio en un fichero de configuración para un servicio, hay que asegurarse de reiniciar dicho servicio para que los cambios en el fichero tengan efecto.

### Listado de Código 15.1: Cómo reiniciar un servicio

```
# /etc/init.d/servicio restart
```

### Paso 2: ¿Los servicios necesarios están corriendo?

Si no están corriendo, hay que iniciarlos. Es tremendamente difícil depurar un servicio que no está corriendo. A veces un servicio puede actuar como si estuviera iniciado pero puede seguir sin funcionar. A veces, cuando se tiene un fichero incorrecto, o una mala transmisión entra en un componente de correo, el servicio abre un puerto para evitar que sea usado por otro proceso. A veces se puede detectar esto con netstat. Reiniciar la máquina limpiará la lista de puertos abiertos

por procesos que no están funcionando bien.

#### **Listado de Código 15.2: Comprobar el estado de un servicio**

```
# /etc/init.d/$service status
# netstat -a | grep $service (o bien: $port)
```

#### **Paso 3: ¿Están todos los servicios usando los últimos ficheros de configuración?**

Si se ha modificado alguno de los ficheros, hay que reiniciar ese servicio para asegurarse de que está utilizando la última versión de dicho fichero. Algunos programas como postfix volcarán su configuración actual.

#### **Listado de Código 15.3: Algunos servicios pueden mostrar su configuración actual**

```
# apachectl fullstatus (necesita lynx instalado)
# apachectl configtest (comprueba el fichero de configuración)
# postconf -n (mostrará los parámetros que está usando postfix)
# /etc/init.d/$service restart
```

#### **Paso 4: Comprobación de bitácoras (log).**

Las bitácoras son mis amigas. El siguiente paso son siempre las bitácoras. A veces es muy útil provocar un fallo en el sistema de correo para, tras esto, comprobar el error en los registros. Ver si hay algo de información que nos ayude a diagnosticar el problema, o quizá, mostrarnos qué componente está fallando.

#### **Listado de Código 15.4: Comprobar las bitácoras**

```
# kill -USR1 `ps -C metalog -o pid=` (para deshabilitar el búfer de metalog)
# nano -w /var/log/mail/current
# cat /var/log/mysql/mysql.log
# tail /var/log/apache/error_log
```

Se pueden encontrar útiles los parámetros de debug\_peer en main.cf. Añadiendo esto, se incrementará el nivel de depuración.

#### **Listado de Código 15.5: Añadir soporte debug\_peer**

```
# nano -w /etc/postfix/main.cf
debug_peer_level = 5
debug_peer_list = $host.domain.name
```

(Se descomenta también uno de los comandos de depuración sugeridos.)

#### **Paso 5: Hablar con el propio servicio.**

SMTP, IMAP y POP3 responden a sesiones telnet. Tal y como vimos cuando comprobamos la configuración de postfix. A veces es útil abrir una sesión telnet a uno de los servicios para ver qué está pasando.

#### **Listado de Código 15.6: Conectar a un servicio con telnet**

```
# telnet localhost $port
```

(SMTP es 25, IMAP es 143, POP3 es 110. Se debe recibir una cadena OK, que nos muestre que el servicio se está ejecutando y listo para aceptar peticiones).

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK Courier-IMAP ready. Copyright 1998-2002 Double Precision, Inc.
```

## **Paso 6: A veces las grandes armas nos dan la información que se necesita: strace.**

Debe tenerse este programa instalado. Es una herramienta muy buena para depurar programas. Se pueden iniciar comandos desde la consola con strace y ver las llamadas al sistema a la vez que ocurren. Muchas veces vuelca una enorme cantidad de información, así que se tendrá que ver en tiempo real mientras que intenta una transacción fallida con el sistema de correo, o bien, mandar la salida a un fichero para luego examinarlo.

### **Listado de Código 15.7: Usar strace**

```
# emerge strace
# strace $command
# strace -p `ps -C $service -o pid=`
```

## **Paso 7: Investigación**

Una vez que se tiene la información, si se puede diagnosticar y arreglar el problema, ¡bien! Si no, probablemente se necesite navegar un poco por la red en busca de información que nos ayude a diagnosticar y arreglar el problema. Aquí hay una lista de sitios donde se puede comprobar si el problema ha sido resuelto. También hay una buena guía para configurar smtp-auth que contiene buenas ideas sobre depuración.

- <http://forums.gentoo.org/> - Excelentes foros para usuarios de Gentoo
- <http://bugs.gentoo.org/> - Base de datos de bugs de Gentoo - Excelente lugar para buscar errores específicos
- <http://postfix.state-of-mind.de/> - guía smtp-auth
- <http://marc.theaimsgroup.com/?l=postfix-users> - Lista de correo de Postfix
- [http://sourceforge.net/mailarchive/forum.php?forum\\_id=6705](http://sourceforge.net/mailarchive/forum.php?forum_id=6705) - Lista de correo de Courier-imap
- <http://www.google.es/linux> - Si todo falla, siempre existe google, nunca me ha fallado.
- El canal #gentoo (#gentoo-es para encontrar usuarios que hablen español) en irc.freenode.net es un buen sitio para buscar ayuda.