Servidor de correo en GNU/Linux con Postfix + MySQL + Courier IMAP + Courier POP + Squirrelmail + Amavis + Clamav + Spamassassin

Alberto Molina Coballes <alberto.molina@hispalinux.es>
José Domingo Muñoz Rodríguez <josedom24@gmail.com>
IES Gonzalo Nazareno. Dos Hermanas (Sevilla)

24 de septiembre de 2006

Resumen

En este documento se describe la instalación y configuración de un servidor de correo postfix, con usuarios almacenados en una base de datos, posibilidad de acceder a los mensajes a través de los protocolos POP e IMAP y con un sistema de filtrado de mensajes para prevenir la difusión de mensajes de SPAM y virus. Esta documentación se elaboró para el curso *Máquinas virtuales para la puesta en marcha de un portal educativo* organizado por el CEP de Sevilla en Septiembre de 2006.

1. Introducción

El correo electrónico se trata sin duda de una de las aplicaciones más utilizadas en Internet. La mayoría de los usuarios de Internet están acostumbrados a términos como servidor smtp, servidor pop, nombre de usuario, etc.

Sin embargo, existe la idea de que la instalación y configuración de un servidor de correo electrónico es una tarea propia sólo de proveedores de servicios de Internet (ISP). Esto en parte se debe a la relativa complejidad de la configuración de un servidor de correo —sobretodo si la comparamos con configuraciones tan sencillas como las de un servidor web como apache o un servidor ftp—.

En este breve documento trataremos de explicar la manera de configurar un servidor de correo de forma sencilla con las siguientes características: validación de usuarios contra una base de datos, posibilidad de consultar el correo a través de servidores IMAP y POP y filtrado de correo en busca de SPAM y virus. La mayoría de los pasos a seguir son comunes a varias distribuciones GNU/Linux, aunque particularizaremos para Debian GNU/Linux (etch).

2. Partes principales de un servidor de correo

MTA son las siglas de *Mail Transfer Agent* o Agente de Transporte de Correo y constituye la base de un servidor de correo.

El uso inicial del correo electrónico está muy ligado a los primeros años de Internet. En aquella época Internet era utilizado principalmente por organismos como universidades y centros de investigación, que habitualmente poseían uno o varios servidores —normalmente bajo alguna variedad de UNIX— a los que se conectaban sus usuarios a través de terminales. Los servidores de estos organismos estaban funcionando de forma ininterrumpida —idealmente claro ;-)—, por lo que se diseñó un protocolo de intercambio de mensajes para esta situación, fue en el año 1982 cuando se establece el protocolo SMTP —Simple Mail Transfer Protocol— a través de la RFC 821 [1] [2].

Un servidor SMTP es capaz de enviar y recibir mensajes de otro servidor SMTP. Una vez que recibe un mensaje lo situa en un buzón accesible para el usuario (en máquinas UNIX es típicamente un directorio o un fichero colgando de */var/mail* o */var/spool/mail*).

En la figura 1 puede verse un esquema de esta situación[3].

Veremos a continuación la instación y configuración de postfix como MTA de un equipo.

[1] Según la wikipedia: Acrónimo inglés de Request For Comments. Conjunto de archivos de carácter técnico donde se describen los estándares o recomendaciones de cualquier cosa. Entre otros los de la propia Internet.

[2] Los RFC están disponibles en http://www.faqs.org/rfcs/

[3] Extraído del tutorial de sendmail de Mark D. Roth http://www.feep.net/sendmail/tutorial/

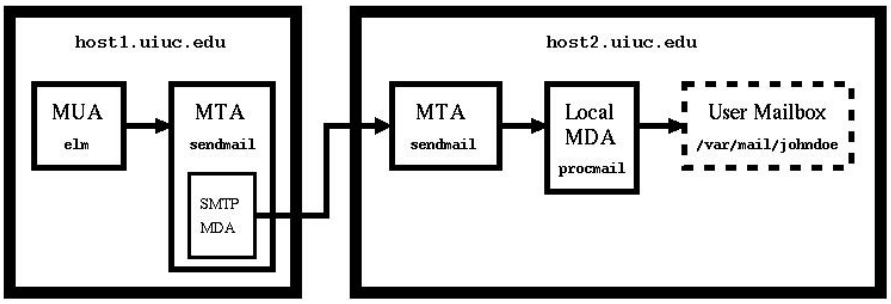

Figura 1: Partes elementales que intervienen en el envío y recepción de un mensaje de correo electrónico

2.1. MUA

MUA son las siglas de *Mail User Agent* o Agente de Usuario de Correo y es el programa que utiliza un usuario para comunicarse con el servidor de correo.

Existe un gran número de MUAs tanto en modo texto como gráfico. Los más conocidos en el entorno GNU/Linux son mail, pine y mutt en modo texto y evolution, kmail y thunderbird para el entorno gráfico.

Si el único elemento que interviniese en la recepción y envío de correo fuesen servidores SMTP, el correo electrónico sólo podría ser utilizado por usuarios reales de cada máquina que accediesen mediante una shell.

2.2. MDA

MDA son las siglas de *Mail Deliver Agent* o Agente de Envío de Correo y es el programa que normalmente se utiliza para filtrar los mensajes y depositarlos en el buzón de cada usuario.

El programa más habitual que se utiliza como MDA es procmail, que no veremos en este documento.

2.3. MAA?

MAA son las siglas de *Mail Access Agent* o agente de acceso de correo. No existe unanimidad en la denominación de este tipo de programas, de ahí el interrogante en el encabezado.

Se trataría de programas necesarios para obtener el correo de los buzones cuando éste no está en ficheros locales accesibles por el MUA.

Los protocolos más utilizados para realizar esta acción son pop3 e imap y los programas que veremos en este documento que lo implementan son courier-pop y courier-imap.

2.4. MRA

MRA son las siglas de *Mail Retrieval Agent* o agente de recuperación de correo, aunque tampoco se trata de un término demasiado extendido.

Se trata de un programa que se conecta a diversos MAA y deposita todo ese correo en diferentes buzones locales —típicamente tras filtrarlo con un MDA—, el ejemplo clásico es fetchmail.

3. Instalación y configuración inicial de postfix

Para instalar el MTA postfix en una máquina basta con teclear:

```
apt-get install postfix
```

Durante la instalación, el instalador de debian —*debconf*— hace una serie de preguntas con idea de dejar el MTA configurado al final. Los puntos más importantes son:

- Configuraremos la máquina como Internet site
- Mandaremos el correo de root al usuario alberto
- Como nombre de correo pondremos "cursocep.org"
- Como dominios finales pondremos:
  cursocep.org, localhost.localdomain, localhost
- Supondremos que sólo se va a enviar correo desde la propia máquina, por lo que a la pregunta de ¿redes locales? contestaremos que: 127.0.0.0/8
- No utilizaremos procmail para la entrega local
- No pondremos límite a los buzones

La mayoría de las opciones de configuración de postfix se ponen en el fichero *main.cf* del directorio */etc/postfix*. Tras la configuración inicial nos debe quedar algo como:

```
# See /usr/share/postfix/main.cf.dist for a commented,
# more complete version

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

myhostname = localhost
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = cursocep.org, localhost, localhost.localdomain
relayhost =
mynetworks = 127.0.0.0/8
```

```
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
```

La configuración de un MTA se puede complicar bastante dependiendo de si se permite que otros equipos envíen correo a través de él —lo que se denomina hacer relay de los clientes—, o debe enviar el correo a través de otro equipo, en lugar de directamente. En lo que sigue supondremos que nuestro equipo no hace relay de ningún otro y es capaz de enviar sus mensajes directamente al servidor destino.

Los ficheros de configuración de postfix se encuentran en */etc/postfix/*, pero la mayoría de cambios se hacen sólo en el fichero *main.cf*.

Para que las modificaciones que vayamos haciendo tengan efecto debemos reiniciar el servicio mediante:

```
postfix reload
```

Ya podemos enviar correo. :-). Para verificarlo desde la shell de un usuario del sistema enviamos un mensaje utilizando el programa mail:

```
alberto@portatile:~$ mail albertomolina@tiscali.es
Subject: Asuntillo
Po zí
[CTRL + D]
Cc:
```

Al principio es muy conveniente tener abierto de forma continua los registros del sistema, por ejemplo con:

```
tail -f /var/log/mail.log
```

En los registros podrá verse que el mensaje ha sido enviado correctamente:

```
Jun 13 13:17:04 localhost postfix/pickup[6978]: 244CB144ED0F: \
uid=1000 from=<alberto>
Jun 13 13:17:04 localhost postfix/cleanup[7008]: 244CB144ED0F: \
message-id=<20050613111704.244CB144ED0F@localhost>
Jun 13 13:17:04 localhost postfix/qmgr[6979]: 244CB144ED0F: \
from=<alberto@cursocep.org>, size=328, nrcpt=1 (queue active)
Jun 13 13:17:13 localhost postfix/smtp[7010]: 244CB144ED0F: \
to=<albertomolina@tiscali.es>, relay=smtp.tiscali.es[212.166.64.67], \
delay=9, status=sent (250 <426E55CC0040A11C> Mail accepted)
Jun 13 13:17:13 localhost postfix/qmgr[6979]: 244CB144ED0F: removed
```

También podemos recibir correo :-). Aunque la verificación de esto dependa más de la conexión a Internet y la configuración del dominio, lo cual se sale del ámbito de este documento. Para una verficación rápida podemos mandar un mensaje entre usuarios de la propia máquina:

```
alberto@portatile:~$ mail viqui@cursocep.org
Subject: Prueba2
Po zí??
Cc:
```

Que produce los registros:

```
Jun 13 13:23:41 localhost postfix/pickup[6978]: 61179144ED0F: \
uid=1000 from=<alberto>
Jun 13 13:23:41 localhost postfix/cleanup[7185]: 61179144ED0F: \
message-id=<20050613112341.61179144ED0F@localhost>
Jun 13 13:23:41 localhost postfix/qmgr[6979]: 61179144ED0F: \
from=<alberto@cursocep.org>, size=329, nrcpt=1 (queue active)
Jun 13 13:23:42 localhost postfix/local[7187]: 61179144ED0F: \
to=<viqui@cursocep.org>, relay=local, delay=1, status=sent \
(delivered to command: procmail -a "$EXTENSION")
Jun 13 13:23:42 localhost postfix/qmgr[6979]: 61179144ED0F: removed
```

Con esta configuración los usuarios locales del sistema podrán enviar y recibir mensajes de correo electrónico. Sin embargo no es esa la situación habitual ni la deseable; debe configurarse el servidor de correo para que usuarios de equipos remotos puedan hacer uso del correo electrónico y además, por seguridad, esos usuarios deben ser sólo usuarios de este servicio —lo que a veces se denomina usuarios virtuales—.

Para conseguir lo anterior, validaremos los nombres de usuarios y contraseñas contra una base de datos —MySQL— e instalaremos en el equipo un servidor IMAP y POP3 —Courier—.

4. MySQL

Vamos a utilizar MySQL para almacanar una bases de datos con la información de nuestros usuarios. El primer paso es instalar el servidor y el cliente de dicha base de datos:

```
apt-get install mysql-server mysql-client
```

Se trata de una base de datos muy sencilla, con una sola tabla, que creamos con las siguientes instrucciones:

```
mysql -u root
mysql> create database maildb;
mysql> use maildb;
mysql> create table passwd(id char(128) NOT NULL UNIQUE PRIMARY KEY,
    -> clear char(128) NOT NULL, name char(128) NOT NULL,
    -> uid int(5) unsigned UNIQUE NOT NULL,
    -> gid int(5) unsigned NOT NULL,
    -> home char(255) NOT NULL, maildir char(255) NOT NULL);
```

Creamos un usuario para manejar dicha base de datos:

```
mysql> grant all privileges on maildb.* to mail@localhost identified
by 'asdasd';
```

Y creamos un par de registros en la tabla passwd para hacer pruebas:

```
mysql> insert into passwd values ('pepe@cursocep.org',
    -> 'qwerty', 'Nombre1', 1100, 8, '/var/mail/','pepe/Maildir/');
mysql> insert into passwd values ('pepa@cursocep.org',
    -> 'zxcvb', 'Nombre2', 1101, 8, '/var/mail/','pepa/Maildir/');
```

Si queremos que los usuarios del sistema sigan recibiendo correo deberemos incluirlos también en la base de datos con sus UIDs reales. En caso de utilizar mutt como MUA, hay que incluir la variable de entorno MAILDIR="ruta completa al buzón".

4.1. Modificación de postfix

En primer lugar es necesario instalar el paquete postfix-mysql. Esta es una ventaja de instalar este servidor en Debian, ya que otras distribuciones no incluyen soporte para MySQL en el paquete postfix.

En segundo lugar hay que realizar cambios en el fichero de configuración de postfix para que utilice mysql para validar los usuarios.

En el fichero */etc/postfix/main.cf* debemos incluir las líneas:

```
virtual_mailbox_base=/var/mail/
home_mailbox = Maildir/
mail_spool_directory = /

virtual_mailbox_maps = mysql:/etc/postfix/mysql_virt.cf
virtual_uid_maps = mysql:/etc/postfix/ids.cf
virtual_gid_maps = mysql:/etc/postfix/gids.cf


local_transport = virtual
```

Hay dos aspectos importantes a tener en cuenta, que se va a utilizar el formato Maildir para los buzones de los usuarios (necesario para utilizar un servidor IMAP) y que la información sobre la sitación de los buzones de los usuarios, su UID y GID se consultan a la base de datos mediante los ficheros *mysql_virt.cf*, *ids.cf* y *gids.cf* situados en el directorio */etc/postfix* y que tendremos que crear a mano. El contenido de estos ficheros será:

mysql_virt.cf

```
user = mail
password = asdasd
dbname = maildb
table=passwd
hosts=127.0.0.1
where_field=id
select_field=maildir
```

ids.cf

```
user=mail
password=asdasd
dbname=maildb
table=passwd
hosts=127.0.0.1
where_field=id
select_field=uid
```

gids.cf

```
user=mail
password=asdasd
dbname=maildb
table=passwd
hosts=127.0.0.1
where_field=id
select_field=gid
```

Puesto que en estos ficheros incluimos la contraseña de nuestro usuario de la base de datos, es conveniente restringir su lectura a usuarios del grupo postfix:

```
portatile:/etc/postfix# chgrp postfix mysql_virt.cf gids.cf ids.cf
portatile:/etc/postfix# chmod 640 mysql_virt.cf gids.cf ids.cf
```

5. Courier IMAP

Courier es también un MTA, sin embargo para esa labor es más recomendable utilizar postfix.

Para nuestras necesidades debemos instalar los paquetes:

```
apt-get install courier-imap courier-authmysql
```

Como es natural los ficheros de configuración se encuentran en */etc/courier*. En primer lugar editamos el fichero *authdaemonrc* y ponemos la línea:

```
authmodulelist="authmysql"
```

En segundo lugar editamos el fichero *authmysqlrc* e incluimos los datos de nuestra base de datos en los siguientes parámetros:

```
MYSQL_SERVER localhost
MYSQL_USERNAME mail
MYSQL_PASSWORD asdasd
MYSQL_PORT 0
MYSQL_OPT 0
MYSQL_DATABASE maildb
MYSQL_USER_TABLE passwd
MYSQL_CLEAR_PWFIELD clear
DEFAULT_DOMAIN cursocep.org
MYSQL_UID_FIELD uid
MYSQL_GID_FIELD gid
MYSQL_LOGIN_FIELD id
MYSQL_MAILDIR_FIELD maildir
```

y reiniciamos los servicios mediante:

```
/etc/init.d/courier-authdaemon restart
/etc/init.d/courier-imap restart
```

6. Courier POP

Debemos instalar el paquete courier-pop:

```
apt-get install courier-pop
```

Y no hace falta configurar nada porque ya está configurado courier-imap ;-).

7. Squirrelmail

Para que nuestros usuarios puedan enviar y recibir correo vamos a instalar un "webmail", en particular squirrelmail. Para ello hacemos como es habitual:

```
apt-get install squirrelmail
```

que si no tenemos instalado apache2 o php4 nos los instalará, además de algún otro paquete adicional.

Squirrelmail no es más que una aplicación en PHP que valida a los usuarios de forma local y que utiliza el servidor IMAP del sistema para la recepción y clasificación de los mensajes y el servidor SMTP en modo local para el envío de los mismos.

En caso de utilizar apache2, no es necesario modificar el fichero de configuración del servidor web, para incluir soporte para php4, ya que viene incluido de forma automática.

La única modificación que tendremos que hacer en el fichero:

```
/etc/apache2/apache2.conf
```

es incluir la línea:

```
Alias /webmail/ /usr/share/squirrelmail/
```

donde *webmail/* es el directorio sobre el directorio raiz del servidor web donde estará la aplicación en php.

Para que los cambios se apliquen, reiniciamos el servidor web mediante:

```
/etc/init.d/apache2 restart
```

Para acceder al webmail, abrimos nuestro navegador y escribimos:

```
http://nuestra_ip/webmail/
```

con lo accederemos a la pantalla que se observa en la figura 2.

Si queremos modificar la configuración de squirrelmail, podemos hacerlo a través del programa:

```
/etc/squirrelmail/conf.pl
```

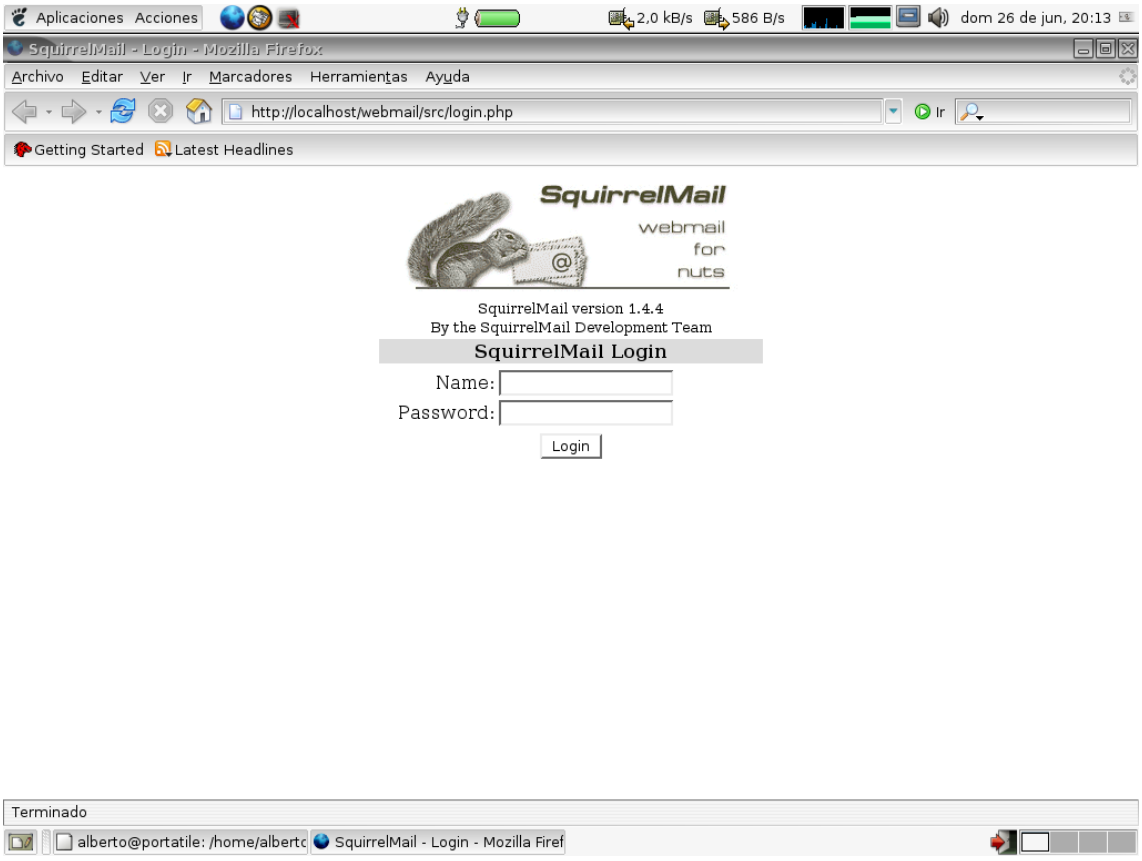

Figura 2: Página de ingreso de squirrelmail

8. Filtrado de correo

Instalamos los siguientes paquetes:

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

El programa encargado de comunicarse con postfix y filtrar los mensajes de correo electrónico es AMaViS (A Mail Virus Scanner). De forma sucinta podemos explicar que pretendemos que los mensajes, después de recibirlos postfix, en lugar de mandarlos a los buzones de los usuarios los envíe a amavis, que los analizará en busca de virus —con clamav— o correo no deseado —con spamassassin—; lógicamente los mensajes serán de nuevo devueltos a postfix para que los envíe a los buzones.

Durante la instalación de clamav nos pedirán bastantes parámetros para su configuración adecuada. En la mayor parte de los casos es suficiente con aceptar los valores por defecto, salvo los siguientes:

- Tipo de socket, elegiremos UNIX
- socket: /var/run/clamav/clamd.ctl
- No utilizar los registros del sistema
- Utilizar en su lugar /var/log/clamav/clamav.log
- Usuario para ejecutar el demonio: amavis
- Grupo para clamav-daemon: clamav

Para no tener problemas con la escritura del socket, damos permiso de escritura al grupo clamav en el directorio /var/run/clamav.

8.1. Configuración de postfix

Para que postfix re-envíe los mensajes que llegan a amavis y los reciba de vuelta debemos incluir las siguientes líneas en los ficheros *main.cf* y *master.cf* de la siguiente manera:

main.cf

```
content_filter=smtp-amavis:[localhost]:10024
```

master.cf

```
smtp-amavis unix --y -2 smtp
-o smtp_data_done_timeout=1200
-o disable_dns_lookups=yes
127.0.0.1:10025 inet n -y --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
```

8.2. Configuración de amavis

Editamos */etc/amavis/amavisd.conf* y modificamos las siguientes directivas:

- $mydomain = 'cursocep.org';
- $forward_method = 'smtp:127.0.0.1:10025';
- $notify_method = $forward_method;

Para re-enviar los mensajes a postfix

- $final_spam_destiny = D_PASS;

Para decidir qué hacer con los mensajes marcados como spam, ya que por defecto los rechaza (D_REJECT).

- $sa_tag_level_deflt = 4.0;
- $sa_tag2_level_deflt = 6.3;
- $sa_kill_level_deflt = $sa_tag2_level_deflt;

Define los niveles para considerar un mensaje como SPAM y marcarlo.

8.3. Pruebas de funcionamiento

Vamos a ver a continuación que para comprobar el buen funcionamiento de todo el sistema debemos fijarnos en las cabeceras de los mensajes de correo.

Por ejemplo si mandamos un mensaje de pruebas a uno de nuestros usuarios tendremos una cabecera como la siguiente:

```
Return-Path: <alberto@cursocep.org>
X-Original-To: pepe@cursocep.org
Delivered-To: pepe@cursocep.org
Received: from localhost (localhost [127.0.0.1])
     by localhost (Postfix) with ESMTP id B256A14DC47E
     for <pepe@cursocep.org>; Mon, 27 Jun 2005 16:36:35 \
     +0200 (CEST)
Received: from localhost ([127.0.0.1])
     by localhost (portatile [127.0.0.1]) (amavisd-new, port 10024)
     with ESMTP id 05625-01 for <pepe@cursocep.org>;
     Mon, 27 Jun 2005 16:36:33 +0200 (CEST)
Received: by localhost (Postfix, from userid 1000)
     id 7E444144EC22; Sun, 26 Jun 2005 22:33:36 +0200 (CEST)
To: pepe@cursocep.org
Subject: Beti campeón
Message-Id: <20050626203336.7E444144EC22@localhost>
Date: Sun, 26 Jun 2005 22:33:36 +0200 (CEST)
From: alberto@cursocep.org (Alberto Molina Coballes)
X-Virus-Scanned: by amavisd-new-20030616-p10 (Debian) at \
cursocep.org
```

donde puede observarse que el mensaje ha sido analizado por amavis.

Si enviamos un típico mensaje de SPAM (con el asunto URGENT AND CONFIDENTIAL y hablando de magníficos negocios ;-)) observamos que nuestro servidor lo marca con '***SPAM***' en el asunto, pero no lo borra para evitar falsos positivos:

```
Return-Path: <alberto@cursocep.org>
X-Original-To: pepe@cursocep.org
Delivered-To: pepe@cursocep.org
Received: from localhost (localhost [127.0.0.1])
     by localhost (Postfix) with ESMTP id B4A3514DE041
     for <pepe@cursocep.org>; Mon, 27 Jun 2005 18:46:07 \
     +0200 (CEST)
Received: from localhost ([127.0.0.1])
     by localhost (portatile [127.0.0.1]) (amavisd-new, port 10024)
     with ESMTP id 07786-01 for <pepe@cursocep.org>;
     Mon, 27 Jun 2005 18:46:00 +0200 (CEST)
Received: by localhost (Postfix, from userid 1000)
     id BA846144EA7D; Mon, 27 Jun 2005 18:46:00 +0200 (CEST)
To: pepe@cursocep.org
Subject: ***SPAM*** URGENT AND CONFIDENTIAL
Message-Id: <20050627164600.BA846144EA7D@localhost>
```

```
Date: Mon, 27 Jun 2005 18:46:00 +0200 (CEST)
From: alberto@cursocep.org (Alberto Molina Coballes)
X-Virus-Scanned: by amavisd-new-20030616-p10 (Debian) at \
cursocep.org
X-Spam-Status: Yes, hits=0.9 tagged_above=-5.0 required=-3.3
     tests=ALL_TRUSTED, MILLION_USD, NIGERIAN_SUBJECT2, RISK_FREE,
     SUBJ_ALL_CAPS, URG_BIZ
X-Spam-Level:
X-Spam-Flag: YES
```

Referencias

[1] Cómo montar un potente sistema de correo con postfix, Javi Polo.
http://bulma.net/body.phtml?nIdNoticia=1621

[2] Sistema de correo con Postfix, OpenLDAP, Courier ((POP3&IMAP) + SSL), SASL, Spamassassin, Amavis-new y SquirrelMail", Sergio González González.
http://es.tldp.org/Manuales-LuCAS/doc-tutorial-postfix-ldap-courier-spamassassin-amavis-squirrelmail/

[3] Tutorial: ISP-style Email Service with Debian-Sarge and Postfix 2.1, Christoph Haas.
http://workaround.org/articles/ispmail-sarge/