

Guia rápida de Postfix

Paco Brufal pbrufal@servitux.com

Versión: 0.9, Agosto 2004

Esta guía rápida explica cómo instalar y configurar el servidor de correo Postfix. Se basa en la distribución Debian Sid. Cualquier comentario será bienvenido. Esta guía se distribuye SIN NINGUNA GARANTIA. No me responsabilizo de los posibles problemas que conlleve el ejecutar todos los pasos que se describen. Esta guía se distribuye bajo licencia GPL (<http://www.gnu.org/>). La última versión de esta guía siempre estará disponible en <http://www.servitux.org/>

1. Introducción

Postfix es un servidor de correo (MTA) muy potente, programado por [Wietse Venema](#), y cuya página web es <http://www.postfix.org/>. En este documento voy a explicar cómo instalar el MTA Postfix en una Debian Sid (inestable), pero es totalmente válido para otras versiones de Debian, incluso para otras distribuciones de Linux.

Cada vez que quieras comprobar que tu servidor está funcionando de manera correcta, tanto para enviar como para recibir, puedes enviar un mensaje de correo a la siguiente dirección: echo@rediris.es. Cualquier mensaje que envíes a esta dirección te será devuelto.

2. Paquetes Debian

Los paquetes de Postfix para Debian que existen en este momento son (*apt-cache search postfix*)

```
postfix - A high-performance mail transport agent
postfix-dev - Postfix loadable modules development environment
postfix-doc - Postfix documentation
postfix-ldap - LDAP map support for Postfix
postfix-mysql - MYSQL map support for Postfix
postfix-pcre - PCRE map support for Postfix
postfix-snap - Postfix Mail Transport Agent - snapshot release
postfix-snap-dev - Postfix-snap loadable modules development environment
postfix-snap-doc - Postfix-snap documentation
postfix-snap-ldap - LDAP map support for Postfix-snap
postfix-snap-mysql - MYSQL map support for Postfix-snap
postfix-snap-pcre - PCRE map support for Postfix-snap
postfix-snap-tls - TLS and SASL support for Postfix snapshots
postfix-tls - TLS and SASL support for Postfix
```

Voy a dar una explicación rápida de qué es cada paquete. Los paquetes necesarios están marcados con un asterisco (*).

- postfix. Este es el paquete principal de Postfix. (*)
- postfix-dev. Entorno de desarrollo.
- postfix-doc. Documentación. (*)

- postfix-ldap. Soporte LDAP.
- postfix-mysql. Soporte MySQL.
- postfix-pcre. Soporte de expresiones regulares. (*)
- postfix-snap-*. Versiones *snapshot*. Pueden ser inestables.
- postfix-tls. Soporte TLS y SASL (SMTP autenticado).

3. Instalación

La instalación de los paquetes Debian se puede realizar de manera sencilla con el comando

```
apt-get install postfix postfix-doc postfix-pcre
```

Si existen dependencias con otros paquetes, *apt-get* también las instalará. Después de bajarse los paquetes de Internet, y antes de instalarlos, posiblemente se nos preguntarán una serie de cosas (relativas a la configuración). Responderemos a esas preguntas, ya que son muy sencillas y nos permitirán crear una configuración base. Luego podemos depurar más la configuración siguiendo esta guía.

El directorio donde se encuentran los ficheros de configuración de Postfix es `/etc/postfix/`, y el fichero principal de configuración se llama `main.cf`.

4. Comandos básicos de Postfix

Existen varios comandos que nos pueden ser útiles mientras usemos Postfix. Una breve lista sería

- *postfix stop*. Este comando para el servidor.
- *postfix start*. Este comando arranca el servidor.
- *postfix reload*. Este comando hace que el servidor relea la configuración sin parar el servicio.
- *mailq*. Para ver la cola de mensajes.
- *postfix flush*. Fuerza el envío de mensajes de la cola de espera.
- *postmap*. Este comando sirve para construir los ficheros auxiliares de Postfix.
- *postconf*. Muestra toda la configuración de Postfix.
- *newaliases*. Este comando reconstruye la base de datos de alias.

5. Modos de ejecución del servidor

Existen 2 modos de ejecución, por así decirlo. El modo *internet site* y el modo *internet site with smarthost*

5.1 *internet site*

El modo *internet site* se caracteriza porque el propio servidor se encarga de repartir los mensajes a sus destinatarios directamente, sin pasar por otro servidor predefinido. Para usar este modo, en el fichero de configuración `/etc/postfix/main.cf` **NO** debe estar definida la opción `relayhost`

```
relayhost =
```

Esta configuración es útil para ordenadores individuales que no están en una red local o tienen conexión permanente a Internet (como ADSL, cable, ...).

5.2 *internet site with smarthost*

El modo *internet site with smarthost* se caracteriza porque el servidor no envía los mensajes directamente a sus destinatarios, sino que los envía a otro servidor de correo, y aquel ya se encargará de enviarlo. Para usar este modo, hay que definir la opción `relayhost` y ponerle como argumento la dirección IP o el nombre de host del servidor SMTP que queramos

```
relayhost = smtp.mi-red-local.com
```

Esta configuración se suele dar en redes locales que ya tienen un servidor SMTP o en conexiones esporádicas a Internet con módem, por ejemplo (el servidor definido sería el de tu proveedor).

6. Seguridad

Por seguridad me refiero a configurar el servidor para que solo lo usen las personas que nosotros queremos, y no abusen de él para enviar **spam**.

6.1 Listas de bloqueo basadas en DNS

Las listas de bloqueo son unas listas de IP de servidores que supuestamente envían **spam**. Entre las listas más usadas se encuentran las RBL de mail-abuse.org o las SBL de spamhaus.org. Puede ver un listado completo de listas de bloqueo en <http://www.decluce.com/JunkMail/Support/ip4r.htm>. Al configurar Postfix para que use estas listas significa que cada vez que llegue un correo a nuestro servidor, Postfix comprobará que la IP del servidor que nos envía el mensaje no se encuentra en esas listas. Una configuración típica en el `main.cf` sería

```
maps_rbl_domains =  
    relays.ordb.org  
    list.dsbl.org  
    blackholes.mail-abuse.org  
    dialups.mail-abuse.org  
    relays.mail-abuse.org  
  
smtpd_client_restrictions =  
    permit_mynetworks  
    reject_maps_rbl  
    check_relay_domains
```

Otro ejemplo, esta vez para Postfix 2.0

```
smtpd_client_restrictions =  
    permit_mynetworks  
    reject_non_fqdn_recipient  
    hash:/etc/postfix/access  
    reject_rbl_client sbl.spamhaus.org  
    reject_rbl_client relays.ordb.org  
    reject_rbl_client opm.blitzed.org  
    reject_unauth_destination
```

NOTA. El uso de listas RBL puede producir el rechazo de mensajes legítimos. Antes de usar una lista RBL se recomienda encarecidamente comprobar cuáles son los criterios de dicha lista para incluir o no un determinado IP. Algunas listas (como SBL o DSBL) utilizan unos criterios muy claros y objetivos y producen pocos o ningún efecto indeseado, mientras que otras tienen unas normas mucho más

agresivas y producen el bloqueo a veces de proveedores enteros, incluyendo un montón de usuarios legítimos.

6.2 Control de envíos

El control de envíos significa que se pueden definir qué direcciones de correo pueden enviar correo a través de nuestro servidor, y qué direcciones de correo no pueden enviar correo a nuestro servidor.

Por host o redes

Mediante la directiva `mynetworks` definimos qué redes o hosts pueden enviar correo a través de nuestro Postfix. Un ejemplo sería

```
mynetworks = 127.0.0.0/8, 192.168.2.0/24, 172.16.3.4/32
```

Con esta configuración estamos definiendo:

- La red 127.0.0.0 puede enviar. Esta red siempre será nuestra propia máquina (*localhost*).
- Los 254 hosts de la red 192.168.2.0 pueden usar nuestro servidor.
- Solo el host 172.16.3.4 puede usar nuestro servidor, y ninguno más de la red 172.16.3.0. Por ejemplo, el 172.16.3.14 no podría.

relay-host

Mediante el sistema *relay-host* definimos que direcciones de correo pueden enviar a través de nuestro servidor. Esto es útil si las personas que queremos que envíen correo tienen una dirección e-mail estable, pero una IP que cambia muy a menudo. Una configuración típica sería esta

```
smtpd_recipient_restrictions =  
    permit_mynetworks,  
    check_sender_access hash:/etc/postfix/usuarios  
    reject_unauth_pipelining,  
    reject_non_fqdn_recipient,  
    reject_non_fqdn_sender,  
    reject_unknown_recipient_domain,  
    reject_unknown_sender_domain,  
    check_relay_domains
```

En la directiva `check_sender_access` vemos que hace referencia a un fichero llamado `/etc/postfix/usuarios`. Este fichero contiene algo parecido a esto:

```
usuario@dominio.com      OK  
usuario2@dominio.com     OK  
usuario3@dominio2.com    OK
```

Esta lista de e-mails significa que dichas direcciones pueden enviar a través de nuestro servidor, independientemente de la IP que tengan. Como puedes imaginar este método no es muy seguro, ya que si algún *spammer* averigua una dirección de correo válida de tu servidor, podrá usarla para enviar correo de manera indiscriminada.

Cada vez que se modifique este fichero se debe ejecutar el comando

```
cd /etc/postfix && postmap usuarios && postfix reload
```

ACL

Las ACL, o listas de control de acceso, son las direcciones de e-mail que **NÓ** pueden enviar correo a nuestro servidor. Si llega un mensaje con alguna de esas direcciones, el servidor lo rechazará. La configuración de las ACL sería

```
smtpd_sender_restrictions =  
    hash:/etc/postfix/access  
    reject_unknown_sender_domain  
    permit_mynetworks
```

Y el fichero `/etc/postfix/access` contendría

```
bob645@yahoo.com      REJECT  
METH0SYSTEM.IT        REJECT  
techemail.com          REJECT  
trafficmagnet.net      REJECT  
email.com              REJECT  
seekercenter.net       REJECT  
icai.ie                REJECT
```

Como vemos se pueden denegar direcciones e-mail concretas (bob645@yahoo.com), o dominios enteros (techemail.com). Cada vez que se modifique este fichero debemos ejecutar

```
cd /etc/postfix && postmap access && postfix reload
```

pop-before-smtp

Este método consiste en que los clientes, antes de poder enviar correo a través de nuestro servidor, deben recoger primero el correo mediante POP3 o IMAP. Al recoger el correo, un demonio controla los logs de los servidores POP3 o IMAP, e introduce en un fichero las IPs de los clientes. A partir de ese momento, desde esa IP se podrán enviar correos, con cualquier remitente, durante el tiempo especificado, que por defecto son 30 minutos.

En la distribución Debian, existe un paquete llamado *pop-before-smtp*. Lo instalaremos con el comando

```
apt-get install pop-before-smtp
```

Luego editamos el fichero `/etc/pop-before-smtp/pop-before-smtp.conf` para elegir el patrón (expresión regular) que se ajusta a las líneas de log que genera nuestro servidor POP3 o IMAP. Reiniciamos el demonio con el comando

```
/etc/init.d/pop-before-smtp restart
```

y comprobamos que al recoger el correo, nuestra IP se introduce en el fichero `/var/lib/pop-before-smtp/hosts.db` con el siguiente script:

```
#!/usr/bin/perl -w  
use strict;  
use DB_File;  
  
# Written by Jonas Smedegaard <dr@jones.dk>.  
# - but copied more or less verbatim from a mail regarding pop-before-  
smtp  
# by Bennett Todd <bet@rahul.net>.  
# If someone recovers the origin of this script please tell me, and I
```

```

will
# add it to this file.
#
# Freely redistributable, or by same rules as those of pop-before-smtp
# (until the original author eventually shows up and claims differently).

die "syntax: $0 filename.db [...] \n" unless @ARGV;

file: for my $file (@ARGV) {
    my %h;
    dbmopen(%h, $file, 0) || do {
        warn "$0: dbmopen($file): $!\n";
        next file;
    };
    print "$_ -> ${h{$_}} \n" for keys %h;
}

```

Pasamos a configurar Postfix. En el fichero `/etc/postfix/main.cf` modificamos la siguiente línea para que incluya el fichero de IPs que genera el demonio *pop-before-smtp*:

```

mynetworks = 127.0.0.0/8, 192.168.1.0/24,
hash:/var/lib/pop-before-smtp/hosts

```

y reiniciamos Postfix con

```
/etc/init.d/postfix restart
```

6.3 Cifrado del envío de mensajes mediante TLS

Los mensajes que se envían desde un servidor a otro viajan en texto claro por defecto. Esto es, cualquier persona que pueda interponerse entre ambos servidores podrá leer el contenido del mensaje. Para evitar esta situación podemos recurrir al cifrado de la conexión mediante TLS (Transport Layer Security).

Para empezar debemos tener instalada la versión de Postfix con soporte TLS y las utilidades *openssl*. En Debian los paquetes se llaman *postfix-tls* y *openssl*, y se instalan con el comando

```
# apt-get install postfix-tls openssl
```

Si ya tenías instalado el paquete *postfix* no te preocupes, pueden convivir ambos sin problemas.

Acto seguido, crearemos un certificado que servirá para cifrar la conexión, ejecutando este comando

```
# openssl req -new -x509 -nodes -out postfix.pem -keyout postfix.pem -
days 3650
```

El certificado lo tendremos en el fichero **postfix.pem**. Ponle los permisos adecuados para que nadie excepto el servidor Postfix pueda leerlo, y pon el fichero en un directorio seguro.

Configuración en el cliente

Aquí veremos cómo configurar un servidor Postfix para que envíe mensajes mediante TLS. Esta configuración puede servir para los usuarios que usan su propia máquina como servidor de correo saliente, o envían todo el correo a otro servidor Postfix que permite conexiones cifradas.

En el fichero `main.cf` debemos poner las siguientes opciones

```
# usar TLS siempre que se pueda
smtp_use_tls = yes
# situación de la clave pública
smtp_tls_cert_file = /etc/ssl/postfix.pem
# situación de la clave privada (en los ficheros .pem, ambas están
juntas)
smtp_tls_key_file = $smtp_tls_cert_file
# nivel de log. Poner 2 hasta que todo funcione bien
smtp_tls_loglevel = 0
# tiempo de validez de las claves
smtp_tls_session_cache_timeout = 3600s
# aviso de conexión TLS
smtp_tls_note_starttls_offer = yes
# tiempo máximo del intercambio de claves
smtp_starttls_timeout = 300s
# fuente de entropía (hay sistemas que no tienen /dev/urandom)
tls_random_source = dev:/dev/urandom
```

Reiniciamos Postfix y veremos que cuando enviamos un mensaje a un servidor con soporte TLS saldrán estas líneas en el log

```
verify error:num=18:self signed certificate
TLS connection established to hostname.hostdomain: TLSv1 with cipher DHE-
RSA-AES256-SHA (256/256 bits)
Peer certificate could not be verified
```

No te asustes de los mensajes de error. Significan que la autenticidad de los certificados no ha podido ser comprobada. Esto se debe a que ninguna autoridad certificadora nos ha firmado los certificados.

Configuración en el servidor

Ahora veremos cómo configurar un servidor con soporte TLS para que los clientes puedan enviar sus mensajes cifrados.

El certificado que usaremos será el mismo que creamos para el Postfix cliente. Añadimos las siguientes líneas al fichero `main.cf`

```
# fichero de clave pública
smtpd_tls_cert_file = /etc/ssl/postfix.pem
# fichero de clave privada
smtpd_tls_key_file = /etc/ssl/postfix.pem
# nivel de log. Pon 2 hasta que todo funcione bien
smtpd_tls_loglevel = 0
# mostrar la posibilidad de recibir mediante TLS
smtpd_use_tls = yes
# fuente de entropía
tls_random_source = dev:/dev/urandom
# tiempo máximo de validez de las claves
smtpd_tls_session_cache_timeout = 3600s
```

Reiniciamos Postfix y a partir de ese momento cuando se conecte un cliente para enviar correo, nuestro servidor le mostrará la posibilidad de enviar correo cifrado. En el log veremos líneas parecidas a estas

```
TLS connection established from unknown[10.1.3.3]: TLSv1 with cipher DHE-
RSA-AES256-SHA (256 /256 bits)
```

Configuración conjunta y comentarios

Si queremos que nuestro servidor sea capaz de recibir y enviar correos cifrados siempre que se pueda, tan solo debemos poner las configuraciones anteriores en el mismo fichero `main.cf`.

Respecto a la autoridad certificadora, nosotros mismos podemos crear una y firmar los certificados.

Como esto se sale fuera de la temática del documento, te recomiendo que visites

<http://www.openssl.org> para obtener más información.

7. Configuraciones avanzadas

Vamos a pasar ahora a las configuraciones avanzadas. Estas configuraciones no suelen ser usadas por usuarios individuales, pero pueden ser útiles para sysadmins.

7.1 Servidores Virtuales

Los servidores virtuales son realmente todos los dominios que gestiona nuestro servidor. Es decir, que un solo servidor de correo puede recibir e-mails para muchos dominios diferentes. La configuración de los servidores virtuales sería

```
mydestination = mihost.dominio.com, localhost.dominio.com, localhost,  
hash:/etc/postfix/virtual
```

Veamos lo que contiene el fichero `/etc/postfix/virtual`

```
dominiovirtual1.com      cualquiercosa  
usuario1@dominiovirtual1.com  usuariolocal1  
usuario2@dominiovirtual1.com  usuariolocal2  
usuario3@dominiovirtual1.com  usuariolocal3  
  
dominiovirtual2.com      cualquiercosa  
usuario1@dominiovirtual2.com  usuariolocal4  
usuario2@dominiovirtual2.com  usuariolocal5  
usuario3@dominiovirtual2.com  usuariolocal6
```

Lo de *cualquiercosa* es eso, cualquier palabra, da lo mismo la que sea, pero **ES OBLIGATORIO** que haya una (los ficheros de hash van por pares). Cada vez que modifiques este fichero debes ejecutar el comando

```
cd /etc/postfix && postmap virtual && postfix reload
```

7.2 Servidores de backup

Los servidores de backup son servidores de correo que solo actúan cuando el servidor principal tiene algún problema y no puede recibir correo (problemas de rutado, caída del servidor, etc.). La misión del servidor de backup es recoger todo el correo mientras el servidor principal está inaccesible, y guardarlo hasta que pueda ser entregado.

Aquí entra en juego el DNS, ya que debemos configurar la zona del dominio para que especifique dos servidores de correo con distintas prioridades. Un ejemplo sería este:

```
# Fichero /etc/bind/midominio.es
```

```

$TTL      86400

@         IN      SOA      root.midominio.es.  hostmaster.midominio.es. (
                                2003073101      ; Serial
                                86400            ; Refresh  (1 dia)
                                7200            ; Retry    (2 horas)
                                2592000        ; Expire   (30 dias)
                                172800 )        ; Default TTL (2 dias)

                IN      A      192.168.4.1
                IN      NS      dns1.midominio.es.
                IN      NS      dns2.midominio.es.
                IN      MX      1      correo
                IN      MX      2      correobackup

www        IN      A      192.168.4.2
correo     IN      A      192.168.4.3
correobackup IN      A      192.168.4.4
dns1       IN      A      192.168.4.1
dns2       IN      A      192.168.4.5

```

Fijate en las líneas "IN MX". Hemos puesto que el servidor principal es *correo* y tiene una prioridad **1**, y el servidor *correobackup* tiene una prioridad **2**. Esto quiere decir que cuando alguien envíe un correo a nuestro dominio, primero lo intentará enviar a la máquina *correo*, y en caso que no pueda enviarlo, lo enviará a la máquina *correobackup*.

El fichero `main.cf` de la máquina *correobackup* debe llevar estas opciones:

```

mydestination = dominio.es
transport_maps = hash:/etc/postfix/transport

```

y el fichero `transport` contiene:

```

midominio.es      smtp:correo.midominio.es

```

Con esto especificamos que todo el correo que llegue para el dominio `midominio.es` debe ser reenviado a la máquina `correo.midominio.es` mediante el protocolo SMTP. El servidor de backup intentará enviar los mensajes a la máquina principal cada poco tiempo, si no lo consigue, lo intenta más tarde. Por defecto, `postfix` devuelve los mensajes que no ha podido enviar en 5 días, así que si tu servidor principal de correo va a estar más de 5 días off-line, puedes aumentar el tiempo de vida de los mensajes en *correobackup* mediante el comando `maximal_queue_lifetime` en el fichero *main.cf*.

7.3 Medios de transporte

Los medios de transporte sirven para desviar el correo entrante a otros servidor de correo en función del dominio. Esto es útil para servidores de ISP que manejan cantidades grandes de correo. Una configuración típica sería

```

transport_maps = hash:/etc/postfix/transport

```

y el fichero `/etc/postfix/transport` contiene

```

dominio1.com      smtp:servidor2.dominio2.com
dominio2.com      smtp:servidor3.dominio3.com

```

dominio3.com

smtp:servidor4.dominio4.com:10025

En la última línea hemos especificado un número, el 10025, que sería el puerto de destino del servidor remoto. En ese puerto debería haber un demonio escuchando las peticiones externas para redigir el correo al servidor. El hecho de especificar un puerto distinto al 25 (SMTP) puede servir para evitar *firewalls*, *proxies* o incluso para asegurarnos que ningún *sniffer* interceptará nuestro correo. Al modificar este fichero, se debe ejecutar el comando

```
cd /etc/postfix && postmap transport && postfix reload
```

7.4 Antivirus y AntiSpam

Cada día son más los virus que se propagan a través del correo electrónico. Con Postfix y un poco de tiempo, se pueden evitar la mayoría de ellos.

Mediante expresiones regulares

Postfix soporta búsqueda de expresiones regulares en las cabeceras de los mensajes. En estas cabeceras es donde siempre vienen definido el o los ficheros que van adjuntos al mensaje. A diferencia de otros ficheros, estos no necesitan ser procesados con **postmap**, simplemente con ejecutar **postfix reload** después de editarlos es suficiente.

Para configurar las búsquedas mediante expresiones regulares la configuración sería esta

```
body_checks = regexp:/etc/postfix/anti_virus,  
pcre:/etc/postfix/pcre_anti_virus  
header_checks = pcre:/etc/postfix/cabeceras
```

El fichero **anti-virus**, filtra los ficheros adjuntos, y el cuerpo del mensaje. Este fichero contiene lo siguiente

```
# Virus  
/(filename|name)="(Happy99|Navidad|prettypark)\.exe"/ REJECT  
/(filename|name)="(pretty park|zipped_files|flcss)\.exe"/ REJECT  
/(filename|name)="(Msininit|wininit|msi2l6)\.exe"/ REJECT  
/(filename|name)="(Avp_updates|Qi_test|Anti_cih)\.exe"/ REJECT  
/(filename|name)="(Emanuel|kmbfejkm|NakedWife)\.exe"/ REJECT  
/(filename|name)="(Seicho_no_ie|JAMGCJJA|Sulfnbk|decrypt-password)\.exe"/  
REJECT  
/(kak|day)\.(reg|hta)/ REJECT  
/Rem I am sorry.*/ REJECT  
/Te mando este archivo para que me des tu punto de vista/ REJECT  
/I send you this file in order to have your advice/ REJECT  
/Espero me puedas ayudar con el archivo que te mando/ REJECT  
/Espero te guste este archivo que te mando/ REJECT  
/Este es el archivo con la información que me pediste/ REJECT  
  
# ficheros extraños  
/(filename|name)=".*\.(asd|chm|dll|hlp|hta|js|ocx|pif)"/ REJECT  
/(filename|name)=".*\.(scr|shb|shs|vb|vbe|vbs|wsf|wsh)"/ REJECT  
  
# CLSID  
/(filename|name)=".*\.{.*}"/ REJECT  
  
# Iframe
```

```
/(\<IFrame\ src=\\"|\<IFRAME\ src=\\"|\<IFRAME\ SRC=\\"))/ REJECT
```

El siguiente fichero es **pcre_anti_virus**, que filtra los ficheros adjuntos por extensiones y tambien fichero codificados en MIME

```
/^begin\s+\d{3}\s+.\+?\.(bat|chm|cmd|com|hta|jse?|pif|scr|shb|vb[esx]|  
ws[fh])\b/ REJECT  
/^s+(file)?name="?.+?\.(bat|chm|cmd|com|hta|jse?|pif|scr|shb|vb[esx]|  
ws[fh])\b/ REJECT
```

Y por último, el fichero **cabeceras**, que filtra las cabeceras de los mensajes (el **from**, **subject**, etc.).

```
/^From: Hahaha <hahaha@sexyfun.net>$/ REJECT  
/^Subject: Enanito si, pero con que pedazo!$/ REJECT  
/^Subject: Re: Your password!$/ REJECT Estas infectado con el Frethem.  
Desinfectate.
```

Como ves, se pueden poner mensajes después del **REJECT**. Estos mensajes serán recibidos por la persona que mandó el mensaje.

Tienes una lista muy extensa de expresiones regulares en la URL

<http://www.hispalinux.es/~data/postfix/>

Integración con Amavisd-New

[Amavisd-New](#) es un software que filtra todos los mensajes de correo, haciendolos pasar por un programa antivirus y/o un programa antispam. En este documento no trataré la instalación o configuración de **amavis**, sino la integración con Postfix 2.0

Una vez instalado y configurado Amavisd-New, añadimos al final éstas líneas en el fichero **master.cf**:

```
smtp-amavis unix - n - 2 smtp  
-o smtp_data_done_timeout=1200  
-o smtp_send_xforward_command=yes  
  
127.0.0.1:10025 inet n n - - smtpd  
-o content_filter=  
-o local_recipient_maps=  
-o relay_recipient_maps=  
-o smtpd_restriction_classes=  
-o smtpd_client_restrictions=  
-o smtpd_helo_restrictions=  
-o smtpd_sender_restrictions=  
-o smtpd_recipient_restrictions=permit_mynetworks,reject  
-o mynetworks=127.0.0.0/8  
-o strict_rfc821_envelopes=yes  
-o smtpd_error_sleep_time=0  
-o smtpd_soft_error_limit=1001  
-o smtpd_hard_error_limit=1000
```

y en el fichero **main.cf** añadimos:

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

Reiniciamos Postfix con `postfix reload` y ya lo tenemos listo.

Greylisting

Greylisting es el método por el cual se deniega el primer envío de un remitente desconocido, mediante un código de error **450** (deferred). Muchos de los virus y spammers no siguen el protocolo SMTP correctamente, con lo que nunca volverán a enviar ese mensaje. Mediante el *greylisting* podemos evitar que nos lleguen mensajes de virus y proxies abiertos, pero no podemos evitar que nos lleguen de servidores de correo mal configurados que permiten relay, aunque con un poco de suerte en el siguiente reenvío ese servidor ya esté en alguna lista RBL y podremos evitarlo.

El funcionamiento es como sigue:

- Llega un correo con remitente desconocido
- Se deniega con un error 450 (intentar más tarde)
- Se guarda la IP, el From y el To en un fichero
- Si el correo era un spam o un virus, es muy raro que nos lo vuelvan a enviar
- Si el correo viene de un servidor SMTP, será enviado de nuevo pasados unos minutos
- Cuando llega de nuevo el correo, lo dejará pasar

Veamos cómo implementar el **greylisting** en un servidor Debian Sid (unstable) Postfix 2.1 o superior (las versiones anteriores no soportan esta característica):

Instalamos el paquete **postgrey**

```
# apt-get install postgrey
```

Editamos el fichero `/etc/postfix/main.cf` y lo dejamos tal que así:

```
[...]
smtpd_recipient_restrictions =
[...]
    reject_unauth_destination,
    check_policy_service inet:127.0.0.1:60000
[...]
```

Con esto configuramos Postfix para que compruebe cada correo que llega mediante el demonio **greylist**, que está escuchando en el puerto 60000 de la IP 127.0.0.1

Reiniciamos Postfix y ya lo tendremos funcionando.