Notas para configurar Postfix

(c) Guillermo Ballester Valor
gbv@oxixares.com
Ogijares, Granada, España

Version 0.1 (11-Oct-2002)

Sumario:

1) Acerca de las listas de restricciones

Como en todos los servidores de correos, la configuración de los mismos es una tarea que puede ser suficientemente complicada como para desanimar a más de un administrador. Por fortuna, la configuración de postfix es sencilla y extremadamente flexible. Postfix es un servidor libre (y gratuito), rápido, seguro, flexible y sobre todo fácil de configurar. Recomiendo sinceramente un cambio desde Sendmail a Postfix.

Es conveniente leerse la documentación de Postfix. Sobre todo lo referente a las restricciones. En ningún caso debe omitirse la lectura del documento de configuración de Postfix . Aquí solo daremos, de forma muy resumida y no exahaustiva, unas pocas claves del proceso de filtrado y selección que sufre un mensaje cuando llega a un servidor Postfix :

1) Los filtros de Postfix estan organizados en forma de **listas de restriciones**. Para que un mensaje sea admitido debe pasar **TODAS LAS LISTAS** de restricciones. Una lista vacia sin restricciones tiene el valor por defecto **OK**. El orden en que se analizan las restricciones dentro de una lista es que esta escrito en **main.cf**

2) Una restriccion puede dar como resultado **OK**, **REJECT** o **DUNNO**.

Las que comienzan con **permit_** dan como resultado **OK** o **DUNNO**. Si **OK** entonces salta a la lista siguiente, si **DUNNO** entonces pasa a la siguiente restriccion de la lista.

Las que comienzan con **reject_** dan como resultado **REJECT** o **DUNNO**. Si**REJECT** entonces **el mensaje se rechaza** y el proceso de filtrado se detiene, si **DUNNO** entonces se analiza la siguiente restricción de la lista.

Algunas otras comienzan con **check_** y su misión es analizar determinados ficheros en los que, a modo de tablas, se analizan determinadas partes del mensaje. El resultado del análisis puede ser **OK**, **REJECT** o **DUNNO**.

Por último **check_relay_domains** da como resultado **OK** o **REJECT**, nunca DUNNO.

Existen, por supuesto, restriciones genéricas como **permit** y **reject** cuyo único resultado posible se deriva del nombre.

3) El Flujo de filtrado de un mensaje es el siguiente:

3.a) Requisitos previos (analiza el formato y protocolo de los mensajes):

smtpd_helo_required = yes/no
smtpd_rfc821_envelopes = yes/no

3.b) Filtrado por listas, el orden preestablecido de listas predefinidas es:

smtpd_client_restrictions
smtpd_helo_restrictions
smtpd_sender_restrictions

smtpd_recipient_restrictions

Además, el usuario puede definir sus propias clases de listas.

3.c) Por último, como última oportunidad para rechazar el mensaje , se puede analizar las cabeceras y cuerpo del mensaje

header_checks
body_checks

Los anteriores mecanismos (y otros que por sencillez aquí no se incluyen), dotan a Postix de una potencia y sencillez extraordinaria.

Si cambiamos algun parámetro de configuración de postfix no hay que recompilar nada, ni siquiera detener el servidor. Simplemente

#) rcpostfix reload

hará que postfix trabaje con la nueva configuración.

2) Limitando, filtrando o personalizando acceso a Postfix mediante tablas o ficheros.

En postfix, una de las formas más flexibles de personalizar o limitar el acceso a nuestro servidor es a través de tablas que se encuentran en ficheros definidos por el administrador. Cada fichero incluye una serie de líneas**patrón - acción** . Como puede verse más abajo, estos ficheros son invocados en alguna de las listas de restricciones y el resultado determina si el mensaje se rechaza (REJECT) si pasa a la siguiente lista de restricciones (OK) o si se continua en la siguiente restriccion de la lista (DUNNO).

Podemos tener uno o varios ficheros que controlan el acceso segun distintos elementos (cliente, remitente, destinatario etc).

El formato de las tablas de accesso sigue siendo

```
modelo          accion
```

cuando un modelo se reconoce en una direccion e-mail, dominio o nombre de host, se realiza la accion.

Una linea que comience con # es un comentario. Una linea que comience en blanco es una continuación.

2.1) MODELOS DE DIRECCION E-MAIL:

usuario@dominio
Se reconoce la direccion e-mail si coincide

mi.dominio
Se reconoce la parte de dominio de una dirección si esta coincide.

usuario@
reconoce todas las direcciones e-mail de usuario independientemente del dominio.

Para que se reconozcan subdominios en nombre de hosts

```
.dominio.ejemplo
```

(nótese el punto inicial). En cuanto a direcciones IP podemos incluir 1, 2, 3 o los 4 octetos. Si ponemos menos de cuatro octetos estos se entenderán como los mas significativos. Por ejemplo

```
217.127
```

reconocerá cualquier dirección ip *217.127.xxx.xxx*

2.2) ACCIONES:

Cuando se reconce un patrón, entonces se produce una acción y se termina el procesado del fichero. Como el fichero ha sido invocado desde una lista de restricciones, entonces postfix tendrá en cuenta ese resultado y procederá en consecuencia. Las posibles acciones son:

REJECT
Rechaza la dirección, dominio etc, que es reconocida en el modelo corespondiente. Se genera una respuesta genérica de error

[45]NN texto
Rechaza la direccion, dominio etc, que es reconocida en el modelo correspondiente generando el codigo de error [45]NN con el texto indicado. Por

ejemplo, para un reconocido emisor de spamm podríamos poner

```
@spammers.com            550 Lo siento, no admitimos correo de spammers.
```

OK
Acepta la direccion, dominio. etc.

restriccion
Aquí se puede poner una lista de restricciones. Es decir, si se reconoce el modelo, entonces se tiene que pasar por la lista de retricciones indicada. Es una forma de recursividad. Por ejemplo, se podría invocar analizar otra lista de restricciones

```
@sospechosos.com                 mis_restricciones
```

donde mis_restricciones ha sido definido en main.cf de la forma

```
smtpd_restriction_classes = mis_restricciones
mis_restricciones = reject_non_fqdn_sender,
        check_sender_access regexp:otros_remitentes_access
```

En este caso se invoca a un fichero con expresiones regulares.

2.3) CÓMO SE UTILIZAN

Para utilizar estos ficheros desde una lista de restricciones tenemos que indicarlos de la forma **tipo_de_restriccion tipo_de_fichero:fichero_map**

Dependiendo del tipo de restriccion, el análisis del fichero se centrará sobre distintos campos. Estos son los tipos de restricciones

check_client_access
chequea el cliente (IP-hostname) de donde se solicita el servicio de nuestro servidor

check_sender_access
chequea la direccion del remitente.

check_helo_access
chequea la direccion etc que un cliente envia en los comandos SMTP, HELO o EHLO.

check_recipient_access
chequea la direccion del destinatario

Por ejemplo, para filtrar qué remitentes admito, podría editar el fichero */etc/postfix/sender_access.* Una vez editado, con postmap

```
#) postmap hash:/etc/postfix/sender_access
```

Por último, en el fichero /etc/postfix/main.cf puedo poner la siguiente lista de restricciones

```
smtpd_sender_restrictions =
        reject_unknown_sender_domain
        check_sender_access hash:/etc/postfix/sender_access
```

Con ello, postfix realizará un chequeo del remitente de acuerdo con las tablas modelo-accion que hayamos puesto en '/etc/postfix/sender_access' . Si el resultado es REJECT, entonces se rechaza el correo, si es OK entonces se analizan las restantes listas de restricciones en 'main.cf' (si es que las hay). Nótese que el resultado puede ser derivado de aplicar, a su vez, otra lista de restricciones indicada en ese fichero.

3) Postfix + dominios virtuales

Es muy habitual que un servidor SMTP aloje y dé servicios a mas de un dominio. Este documento trata de aclarar un poco la configuracion de Postfix para que pueda implementar esta funcionalidad. Por su puesto, la lectura de este documento debe complemetarse con la consulta de la documentación oficial del desarrollador de Postfix.

La mayor parte de lo que aquí se expone proviene de la documentación html que incluye el paquete postfix y del directorio de ejemplos de configuración que suele encontrarse en el directorio **/etc/postfix** tras la instalación.

Los usuarios de Sendmail que estén acostumbrados a manejar dominios virtuales a través de **virtusertable** no deben tener dificultad alguna para migrar esa funcionalidad a Postfix.

En general, hay dos formas de manejar los dominios virtuales en postfix: con el **Estilo Postfix** y con el **Estilo Sendmail**

3.1-Estilo Postfix

Los alias, usuarios locales y listas de correos no son visibles en un dominio virtual (mas abajo explico qué es eso). Para especificar ese estilo, tenemos que incluir en ’/etc/postfix/virtual’ líneas como las siguientes:

```
mi_dominio.virtual                  Y aqui ponemos cualquier cosa, no
importa.
usuario1@mi_dominio.virtual         direccion1
usuario2@mi_dominio.virtual         direccion2, direccion3, ...
usuario3                            direccion4, direccion5, ...
@mi_dominio.virtual                 direccion6, ....
```

En general, la forma de las líneas es

```
patrón     resultado
```

cuando uno de los patrones es reconocido en una dirección de correo, entonces ésta última es sustituida por el resultado.

En el ejemplo, la primera linea es obligatoria para especificar el estilo postfix, las otras líneas ilustran las distintas posibilidades. Nótese que podemos redirigir a una o más direcciones.

La segunda línea redirigira todo correo cuyo destinatario es *usuario1@mi_dominio.virtual* a direccion1.

La tercera línea muestra el hecho de que se puede redirigir a más de una dirección.

En la cuarta línea, *usuario3@algun.dominio* es redirigida a *direccion4*, *direccion5* ... cuando ’*algun.dominio*’ figura en alguno de los parametros de configuracion **$myorigin**, **$mydestination** o **$inet_interfaces**.

El la quinta linea, redirigirá todo el correo de cualquier usuario de *mi_dominio.virtual* a *direccion6* ...

En cuanto a las direcciones de reenvio, las que tienen forma ’@otro_dominio ’ son expandidas al mismo usuario de ’otro_dominio’, pero esa expansión solo funciona en la primera direccion de la lista (puede haber más, como se muestra en los ejemplos anteriores).

Como se ha dicho en un principio, el servidor postfix aceptará correo de cualquier usuario conocido en *mi_dominio.virtual* y rechazará como imposible cualquier usuario no conocido en ese dominio. Aquí radica la diferencia en el

estilo sendmail. Los usuarios locales NO son reconocidos a no ser que se adapten a algunas de las condiciones del fichero **virtual** dentro de ese dominio. Los alias definidos en **/etc/aliases** TAMPOCO son reconocidos en este estilo. Hay que utilizar el **estilo sendmail** para poder utilizar implicitamente alias y usuarios locales.

3.2-Estilo Sendmail

Para fijar ese estilo tenemos que añadir en **/etc/postfix/main.cf** el nombre del dominio virtual al parámetro de configuración **$mydestination**:

```
mydestination= $myhostname localhost.$mydomain $mydomain
            mi_dominio.virtual
```

En el fichero **/etc/postfix/virtual** ahora NO debemos incluir la primera línea del ejemplo del estilo postfix. Es decir, ahora quedaría:

```
usuario1@mi_dominio.virtual                direccion1
usuario2@mi_dominio.virtual                direccion2, direccion3, ...
usuario3                                   direccion4, direccion5, ...
@mi_dominio.virtual                        direccion6, ....
```

con el mismo significado y posibilidades. Ahora bien, la diferencia respecto al **estilo postfix** es que ahora son utilizables los alias, usuarios locales y listas de correos. Ejemplo, supongamos que tenemos definido el usuario local *usuariolocal* y que en el fichero aliases tenemos también el alias,

```
mi_alias:     usuariolocal
```

Ahora postfix SI que redirigirá el correo a destinatarios como *mi_alias@mi_dominio.virtual* o *usuariolocal@mi_dominio.virtual* a *usuariolocal@mi_dominio.local* aunque no aparezcan explícitamente en el fichero. Evidentemente, si tampoco el usuario es reconocido como alias o usuario local, es mensaje es devuelto.

Por si todas estas posibilidades no fueran pocas, en amboos estilos se pueden incluir también expresiones regulares.

Para generar el fichero **/etc/postfix/virtual.db** o **/etc/postfix/virtual.dbm** , una vez editado **/etc/postfix/virtual**hay que ejecutar **postmap**

```
#) postmap hash: /etc/postfix/virtual
```

También hay que incluir en ’/ect/postfix/main.cf’ la linea

```
virtual_maps = hash:/etc/postfix/virtual
```

Con lo que postfix lo utilizara al cabo de alrededor de un minuto, o inmediatamente tras

```
#) rcpostfix reload
```

En este ejemplo se ha utilizado el fichero **/etc/postfix/virtual**, pero podríamos haber editado otro, en cuyo caso deberíamos incluirlo en **$virtual_maps**

3.3 - Cómo se incluye más de un dominio virtual

Pues depende del estilo de modelo virtual que se adopte. Si se utiliza el estilo sendmail entonces se debe incluir los dominios en **main.cf** cuando se defina el parametro **$mydestination**

```
mydestination= $myhostname localhost.$mydomain $mydomain
            mi_dominio.virtual1 mi_dominio.virtual2  ...
```

En caso de utilizar estilo postfix entonces se deber incluir en el fichero definido en **$virtual_maps** una seccion por cada dominio virtual.

```
mi_dominio.virtual1   cualquier cosa
.....
mi_dominio.virtual2   cualquier cosa
.....
```

Lo habitual es que desemos, asimismo, distribuir el correo que venga de los dominios virtuales al resto de la red (hacer relay). Como le valor por defecto de **$relay_domains** es **$mydestination**, si ya hemos incluido todos los dominios en los que queremos hacer relay en **$mydestination**, no hace falta incluirlos en **$relay_domains**. En caso contrario, por ejemplo si se utiliza el estilo postfix, habria que incluirlos:

```
relay_domains = $mydestination
        mi_dominio.virtual1
        mi_dominio.virtual2
```

4) Autentificación SMTP mediante SASL

Para la distribución Linux SuSE, el autor ha escrito este documento. De forma general, en lo que sigue supone que se ha instalado la librería cyrus-SASL en cualquiera de las formas (habitualmente mediante RPMS) y que Postfix se ha compilado para la utilización de las mismas.

La autentificacion del usuario en un servidor SMTP es un tema muy importante, sobre todo cuando los usuarios están fuera de nuestra red local. No podemos (o no debemos) dejar nuestro servidor abierto a cualquier usuario. Si así lo hicieramos, en pocas horas el servidor se saturaría mandando miles de correos no solicitados (SPAM) proveniente de usuarios sin escrúpulos y, poco más tarde, tendríamos el 'privilegio' de figurar en las listas negras antispam resultando que nuestro propio correo se vería bloqueado en gran parte de la Red.

Hay muchas formas de evitar usos ilícitos de nuestro servidor. El más habitual es restringir la posibildad de envio de correo hacia el exterior solamente a los usuarios de una intranet. Los servidores externos sólamente podrían enviar correo cuando el destino final es nuestra intranet. No obstante ese es un mecanismo que solamente sirve cuando los potenciales usuarios estan dentro de una red conocida. Si el usuario se conecta a través de un modem, en que cada vez su dirección IP varía, es necesario habilitar algún mecanismo de identificación. Aún incluso dentro de una red, la identificación es una práctica aconsejable.

Para conseguir la identificación en los servidores SMTP, la practica más habitual es SMTP después de POP y SASL. Hay otros mecanismos muy débiles, como permitir sólamente usuarios con una determinada dirección de correo (un atacante podría suplantar nuestra identidad) .

El mecanismo de identificación SMTP despues de POP se basa en que para recojer el correo desde el servidor POP hay que identificarse. Pues bien, este método guarda la dirección IP desde la que nos identificamos como usuario POP y, a continuación, si solicitamos un servicio SMTP, el servidor comprueba que el usuario proviene de la dirección IP considerada 'amistosa'.

Otro mecanismo más sólido es la identificación directa en el servidor SMTP. Para ello SASL implementa, entre otros, una serie de mecanismos de encriptación que hace más seguro el envío de claves de usuario y Passwords a traves de la red. Es aconsejable el uso de SASL frente a otros mecanismos más inseguros.

Este documento trata de ilustrar cómo conseguir esa funcionalidad en un servidor Postfix.

Cuando la identificación SASL de un cliente tiene éxito, podemos utilizar los ficheros de configuración de postfix para darle los privilegios adecuados. Las librerías SASL invocadas por postfix tienen sus própias bases de datos de usuarios/contraseñas. Hay que crear un conjunto de usuarios/contraseñas para ser utilizadas por el servidor smtpd de postfix.

4.1 - Configuración SASL.

En este apartado veremos cómo configurar la libreria cyrus-SASL para lograr la identificación SASL. Es importante destacar que la identificación SASL en postfix sirve para dar acceso al servidor SMTP, y que este acceso puede ser todo lo privilegiado que queramos dependiendo de las restricciones que añadamos a la configuración de postfix. Los usuarios/contraseñas que se definan serán para SASL y, en principio, no tienen por qué nada que ver con usuarios/contraseñas de login locales utilizados, por ejemplo para POP3.

Crear o editar el fichero **/usr/lib/sasl/smtpd.conf.** En la única línea de este fichero debe figurar el método en el que se almacenarán las claves:

pwcheck_method: sasldb

Ahora tendremos que crear las claves para cada usuario. Para ello hay que definir un **REALM** que supondremos como el nombre del dominio donde está el servidor. Téngase en cuenta que para que postfix identifique correctamente, este **REALM debe ser único para todos los usuarios del servidor SMTP,** independientemente de que hayan uno o más dominios virtuales alojados en el servidor. El programa utilizado para generar las claves es**saslpasswd**.

#) saslpasswd -c -u REALM usuario

Nos preguntará la clave y la confirmación. Se puede consultar los usuarios/realm introducidos con **sasldblistusers**

#) sasldblistusers
user: usuario realm: mi_dominio mech: PLAIN
user: usuario realm: mi_dominio mech: CRAM_MD5
user: usuario realm: mi_dominio mech: DIGEST_MD5

Como puede verse, cada combinación realm/usuario soporta distintos tipos de encriptación. Hay que notar aquí que los usuarios y claves no tienen por qué ser los mismos que los utilizados para recoger el correo mediante POP. Pueden (¿y deben?) ser distintos. No obstante, algunos clientes de correo ofrecen la posibilidad de utilizar los mismos usuarios/password que POP.

El fichero que contiene estas claves es **/etc/sasldb** . Concretamente, para la distrubución SuSE el autor no ha logrado que la identificación funcione correctamente si no se cambian manualmente los permisos de ese fichero

#) chmod 644 /etc/sasldb

lo cual es una solución que no es satisfactoria y que debe ser subsanada.

4.2 - Configuración final POSTFIX.

Ahora hay que configurar Postfix para que utilice SASL. Este es un tema que puede cambiar según las necesidades y el grado de paranoia de cada Administrador. Como caso más general, supondremos que queremos dar acceso a usuarios externos identificados.

El fichero de configuración de Postfix es **/etc/postfix/main.cf.** Ya debemos tener una configuración por defecto que han generado los scripts de SuSE. Para ver los valores de los parametros de control generados:

#) postconf -n

Pero tendremos que editar el fichero. Incluir las siguientes definiciones:

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain = $mydomain
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
```

Con la primera línea habilitamos postfix para utilizar SASL. Con la segunda damos el valor adecuado de REALM a las librerías SASL. La tercera es una medida de seguridad para evitar que se identifiquen como anonymous (y dejaríamos un agujero de seguridad). La última es para no bloquear algunos clientes.

MUY IMPORTANTE: la definición de 'smtpd_sasl_local_domain' debe coincidir con el REALM con que hayamos definido los usuarios/contraseñas con saslpasswd. En el ejemplo anterior, se supone que el REALM utilizado es el parámetro 'mydomain' de postfix.

Ademas, hay que introducir restriciones. Como mínimo, debe haber la siguiente lista de restricciones en main.cf (puede ocupar varias líneas pero el primer caracter de una continuación debe dejarse en blanco. Cada restricción se separa por comas o espacio.

```
smtpd_recipient_restrictions =
      permit_mynetworks
      permit_sasl_authenticated
      check_relay_domains
```

5) Filtrado de correo no deseado. Listas negras.

La utilización básica de las listas negras es trivial en postfix.

1) Hay que definir en **main.cf** las listas que quieras utilizar. Por ejemplo

maps_rbl_domains = relays.ordb.org, dnsbl.njabl.org

2) después, en las listas de restricciones, se invocan dichas listas con la restricción

reject_maps_rbl.

6) Un ejemplo real.

Por último, para ver la configuracion que tenemos en main.cf (quitando comentarios y otras historias) podemos hacer:

```
#) postconf -n
```

Y obtendremos un listado con lo importante de main.cf. Aquí tenen un listado real de mi configuración actual (yo no tengo dominios virtuales)

mozart #) postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
broken_sasl_auth_clients = yes
canonical_maps = hash:/etc/postfix/canonical
command_directory = /usr/sbin
config_directory = /etc/postfix
content_filter = vscan:
daemon_directory = /usr/lib/postfix
debug_peer_level = 2
default_process_limit = 10
defer_transports =
disable_dns_lookups = no
header_checks = regexp:/etc/postfix/regexp_intruders
inet_interfaces = all
local_recipient_maps = $alias_maps unix:passwd.byname
mail_name = Postfix on SuSE Linux 8.0 (i386)

mail_owner = postfix
mail_spool_directory = /var/mail
mailq_path = /usr/bin/mailq
manpage_directory = /usr/share/man
maps_rbl_domains = relays.ordb.org, dnsbl.njabl.org
masquerade_domains = oxixares.com
masquerade_exceptions = root
message_size_limit = 4096000
mydestination = $myhostname, localhost.$mydomain, $mydomain
myhostname = mozart.oxixares.com
mynetworks_style = subnet
myorigin = $mydomain
newaliases_path = /usr/sbin/sendmail
queue_directory = /var/spool/postfix
readme_directory = /usr/share/doc/packages/postfix/README_FILES
relay_domains = $mydestination
relayhost =
relocated_maps = hash:/etc/postfix/relocated
sample_directory = /etc/postfix
sender_canonical_maps = hash:/etc/postfix/sender_canonical
sendmail_path = /usr/sbin/sendmail
setgid_group = maildrop
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
smtpd_helo_required = yes
smtpd_recipient_restrictions = reject_non_fqdn_sender,
reject_non_fqdn_recipient, permit_mynetworks,
permit_sasl_authenticated, reject_unauth_destination,
reject_unauth_pipelining, reject_invalid_hostname,
reject_non_fqdn_hostname, reject_maps_rbl,
warn_if_reject reject_unknown_client, permit
smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain = $mydomain
smtpd_sasl_security_options = noanonymous
smtpd_sender_login_maps = hash:/etc/postfix/login_auth
strict_rfc821_envelopes = yes
transport_maps = hash:/etc/postfix/transport
virtual_maps = hash:/etc/postfix/virtual