

Cómo Configurar Postfix con SASL.

Este es un manual en versión BETA, puede haber algunos errores tipográficos y de ortografía.

Última modificación: Viernes 26 de Agosto de 2003, 8:55

AM. Joel Barrios Dueñas

jbarrios arroba linuxparatodos punto net

<http://www.linuxparatodos.net/>

Usted puede contribuir financiando la elaboración de más documentos como éste haciendo aportaciones voluntarias y anónimas en:

Bital, Banco Internacional, S.A. (México)

Cuenta: 4007112287, Sucursal 0643

A nombre de: Joel Barrios Dueñas.

Copyright.

© 1999, © 2000, © 2001, © 2002 y © 2003 Linux Para Todos. Se permite la libre distribución y modificación de este documento por cualquier medio y formato **mientras esta leyenda permanezca intacta junto con el documento** y la distribución y modificación se hagan de acuerdo con los términos de la **Licencia Pública General GNU** publicada por la Free Software Foundation; sea la versión 2 de la licencia o (a su elección) cualquier otra posterior. La información contenida en este documento y los derivados de éste se proporcionan tal cual son y los autores no asumirán responsabilidad alguna si el usuario o lector hace mal uso de éstos.

Software requerido:

Se recomienda utilizar **al menos** las siguientes versiones de software. No utilice versiones inferiores.

postfix-1.1.12

cyrus-sasl-2.1.10

cyrus-sasl-plain-2.1.10

cyrus-sasl-md5-2.1.10

imap-2001a

Antes de continuar verifique siempre la existencia posibles actualizaciones de seguridad. Para Red Hat Linux 8.0 y 9 hay paquetería de actualización en los siguientes enlaces:

<ftp://updates.redhat.com/8.0/en/os/i386/> , si posee alguna distribución basada sobre Red Hat™ Linux 8.0

<ftp://updates.redhat.com/9/en/os/i386/> , si posee alguna distribución basada sobre Red Hat™ Linux 9

Si utiliza `apt` para RPM , solo bastará ejecutar:

```
apt-get update
```

```
apt-get install postfix cyrus-sasl cyrus-sasl-plain
```

cyrus-sasl-md5 imap

Sustituir Sendmail con Postfix en el sistema.

Si utiliza distribuciones como Red Hat Linux 9 o Yellow Dog Linux 3.0, solo necesitará utilizar el comando *redhat-switch-mail* , el cual se encargará de realizar todos los cambios pertinentes en el sistema para cambiar de Sendmail hacia Postfix como *MTA* predeterminado.

SASL.

¿Que es SASL?

SASL es el acrónimo de "*Simple Authentication and Security Layer*" que significa "*Capa de Seguridad y Autenticación Simple*". SASL es un método para añadir soporte para autenticación a los protocolos como SMTP .

Procedimientos

Edite */usr/lib/sasl/smtpd.conf* y verifique que esté el siguiente contenido:

```
pwcheck_method: sasldb
```

Para activar el soporte SASL, se debe asignar al menos una contraseña a cualquier cuenta de usuario en el sistema.

```
/usr/sbin/saslpasswd -c usuario1
```

Debido a que Postfix gtrabaja por defecto con *chroot()* , y es conveniente que siga haciendolo así, es perferible mover */etc/sasldb* hacia la ruta acequible para postfix que corresponde a */var/spool/postfix/etc/sasldb* :

```
mv /etc/sasldb /var/spool/postfix/etc/sasldb
```

A fin de mantener un nivel de seguridad aceptable y permitir al mismo tiempo que ostfix pueda acceder al fichero de contraseñas, debemos asignar nuevos permisos a */var/spool/postfix/etc/sasldb*:

```
chmod 640 /var/spool/postfix/etc/saslauthd
chgrp postfix /var/spool/postfix/etc/saslauthd
```

Lo anterior designa permisos de lectura y escritura para *root* y de solo lectura para el usuario *postfix* .

Crear enlace simbólico para poder utilizar el comando *saslpasswd* normalmente y no tener que realizar el tedioso procedimiento de tener que copiar de nuevo dicho fichero cada vez que se de da alta o se modifique una cuenta.

```
ln -s /var/spool/postfix/etc/saslauthd /etc/
```

De de alta todas las cuentas de usuario restantes:

```
/usr/sbin/saslpasswd -c usuario2
/usr/sbin/saslpasswd -c usuario3
Etc.
```

Levantar el demonio *saslauthd* y añadirlo a los servicios activos:

```
/sbin/service saslauthd start
/sbin/chkconfig saslauthd on
```

IMAP y POP3

Configuraremos de una vez el protocolo mediante el cual el usuario recuperará su correo. Puede utilizarse IMAP (*Internet Message Access Protocol*) o bien POP3 (*Post Office Protocol, versión 3*) o bien ambos. Si se elige IMAP, el todo el correo permanecerá en el servidor hasta que sea eliminado explícitamente. Si se utiliza POP3, el correo será descargado por completo por medio del cliente del correo hacia la máquina del usuario.

Para habilitar uno u otro protocolo, o ambos, utilizaremos el comando **chkconfig** :

```
chkconfig imap on
chkconfig ipop3 on
```

Postfix.

¿Que es Postfix?

Postfix es un MTA, que es acrónimo de "*Mail Transport Agent*" y que a su vez significa "*Agente de Transporte de Correo*". Postfix fue desarrollado como un reemplazo para Sendmail. Postfix es la versión libre de Secure Mailer de IBM cuyo desarrollo fue iniciado por Wietse Venema en el T.J. Research Lab propiedad de IBM.

Postfix es una magnífica alternativa hacia Sendmail, el cual desafortunadamente posee un pésimo historial de seguridad. ¿Por que se sigue usando Sendmail? Porque viene incluido como MTA por defecto en la mayoría de las distribuciones de Linux y otros sabores de UNIX. Postfix es menos complicado de configurar, es más rápido, fácil de administrar y seguro.

Postfix trabaja por defecto dentro de una *jaula* (chroot) localizada en */var/spool/postfix*, y por tal motivo es todavía más seguro reduciendo enormemente los riesgos para el sistema en el caso del surgimiento de una vulnerabilidad.

Procedimientos.

Editar al final del fichero */etc/postfix/aliases* la línea que por defecto especifica como alias de root a postfix, lo cual se debe cambiar y definir en su lugar una cuenta de usuario válida. **¡Eso es algo muy importante!**

root: usuario

Al terminar, ejecute el comando *newaliases* a fin de convertir */etc/postfix/aliases* en */etc/postfix/aliases.db*.

newaliases

Editar */etc/postfix/main.cf* y donde se definirán algunas variables.

Redundar el nombre del servidor, el cual también debe ser un nombre de dominio completamente resuelto por un DNS:

myhostname = mail.midominio.com

Definase además el dominio a utilizar.

mydomain = midominio.com

Redundar el dominio a utilizar para los mensajes salientes. Por defecto se añade \$myhostname, pero si así se desea, puede establecerse \$mydomain.

```
myorigin = $myhostname
```

En Red Hat Linux se establece localhost como única interfaz para escuchar peticiones. Puede comentarse la línea, puesto que Postfix escuchará peticiones por todas las interfaces por defecto:

```
# inet_interfaces = localhost
```

Defina los dominios a administrar:

```
mydestination = $myhostname, localhost.$mydomain,  
dominio.virtual
```

Defina los equipos a los que se permitirá enviar libremente el correo a través de postfix:

```
mynetworks = 192.168.1.0/24, 127.0.0.0/8
```

Si desea incrementar el nivel de seguridad o bien simplemente le interesa utilizar el método de autenticación que se explica más adelante, puede definir una lista de IP contenida en un fichero, el cual incluirá además de las IP propias del servidor y las de aquellos equipos a los que verdaderamente se permitirá enviar correo libremente a través de postfix:

```
mynetworks = $config_directory/mynetworks
```

Un ejemplo del contenido de mynetworks sería:

```
127.0.0.0/8  
192.168.1.254
```

Si se utiliza una dirección IP pública, como por ejemplo 148.240.39.174, nunca se defina el segmento de este en mynetworks (148.240.39.0/24), ya que de otro modo cualquier otro equipo de dicha red tendría permitido hacer uso del servidor de correo.

Redundemos los dominios que se permite para enviar correo designando la variable \$mydestination:

```
relay_domains = $mydestination
```

A fin de poder aprovechar las funciones de filtrado y administración del correo que posee procmail, defina en la ruta donde se localiza éste:

```
mailbox_command = /usr/bin/procmail
```

Si se va a utilizar autenticación, debe definirse lo siguiente:

```
smtpd_sasl_auth_enable = yes
smtp_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $myhostname
smtp_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtp_sasl_password_maps = hash:/etc/postfix/saslpass
smtpd_recipient_restrictions =
  permit_sasl_authenticated,
  permit_mynetworks,
  reject_unknown_client,
  check_relay_domains
```

Crear el fichero */etc/postfix/saslpass* y definir una cuenta específica por cada dominio administrado que además deberá autenticar escribiendo la contraseña correspondiente y que previamente se asignó con el comando *saslpasswd* .

```
mail.midominio.com
dominio1@mail.midominio.com:clave-del-usuario
dominio.virtual dominio2@dominio.virtual:clave-del-usuario
```

Se deben crear cuentas en el sistema para este fin particular, y no olvidar que a fin de poder ser utilizadas por Postfix, se asignan las contraseñas para dichas cuentas con el comando *saslpasswd* .

```
Se debe convertir /etc/postfix/saslpass
a/etc/postfix/saslpass.db
```

```
postmap /etc/postfix/saslpass
```

```
Asignar permisos apropiados a /etc/postfix/saslpass
y/etc/postfix/saslpass.db
```

```
chmod 600 /etc/postfix/saslpass*
```

La configuración ha concluído, y solo resta arrancar o reiniciar Postfix.

service postfix restart

Alta de las cuentas de correo.

Las cuentas de correo deben de darse de alta sin acceso a shell, recordando que, a diferencia de los protocolos *IMAP* o *POP3* , al acceso hacia el protocolo SMTP le será asignado una contraseña con **el comando saslpasswd en lugar del comando passwd** .

```
useradd -s /bin/false -c "Nombre del usuario" usuario  
saslpasswd -c usuario
```

Si necesita cambiar la contraseña a cualquier usuario, utilice el comando saslpasswd sin parámetros adicionales:

```
saslpasswd usuario
```

Si desea dar de baja la contraseña de cualquier usuario, utilice el siguiente comando:

```
saslpasswd -d usuario
```

Finalmente, para poder recuperar el correo ya sea a través de IMAP o POP3, si será necesario asignar una contraseña con el comando **passwd** :

```
passwd usuario
```

Probando el servicio SMTP.

```
telnet mail.midominio.com 25
```

Lo anterior devuelve la siguiente salida:

```
Trying 192.168.1.254...  
Connected to mail.midominio.com.  
Escape character is '^]'.  
220 mail.midominio.com ESMTP Postfix
```

Verifique comando "helo" para el dominio

```
helo mail.midominio.com
```

Lo anterior devuelve lo siguiente:

250 mail.midominio.com

Verifique funciones del servidor con comando "ehlo"

ehlo mail.midominio.com

Lo anterior debe devolver algo como lo siguiente.

250-mail.midominio.com

250-PIPELINING

250-SIZE 10240000

250-VERFY

250-ETRN

250-AUTH PLAIN LOGIN DIGEST-MD5 CRAM-MD5

250-AUTH=PLAIN LOGIN DIGEST-MD5 CRAM-MD5

250-XVERP

250 8BITMIME

Note las dos líneas que especifican los métodos de autenticación.

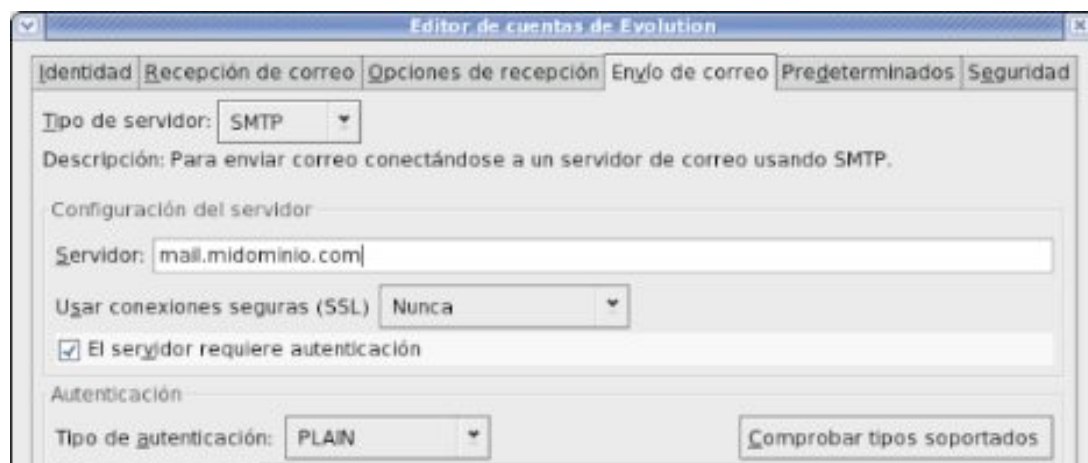
250-AUTH PLAIN LOGIN DIGEST-MD5 CRAM-MD5

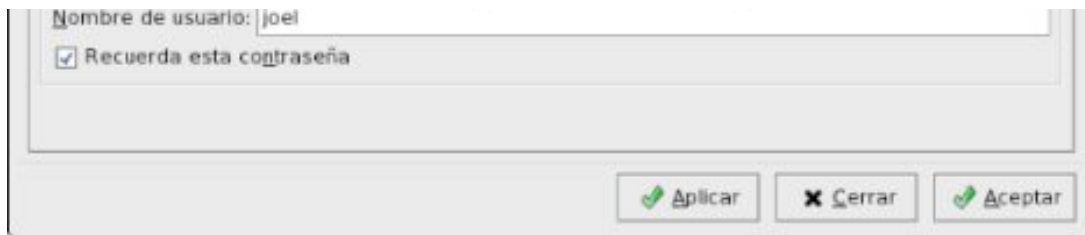
250-AUTH=PLAIN LOGIN DIGEST-MD5 CRAM-MD5

Ambas especifican los métodos soportados. La segunda particularmente se utiliza para los clientes de correo electrónico que autentican con errores o bien con el viejo y obsoleto protocolo "AUTH=PLAIN", como sería el caso de Outlook Express 4.

Para salir del *shell* , ejecute el comando "**quit**" .

Solo resta probar la configuración del servidor de correo con cualquier cliente de correo electrónico, el cual deberá de configurarse con soporte para autenticación para el protocolo SMTP . Puede seleccionarse PLAIN, LOGIN, DIGEST-MD5 o CRAM-MD5 como método de autenticación.





Parámetros adicionales para Postfix.

Hay algunos parámetros adicionales para */etc/postfix/main.cf* que permitirán controlar mejor el comportamiento del servidor de correo de manera sencilla.

mailbox_size_limit

Si se desea limitar el tamaño máximo de los buzones de correo, puede hacerse a través de *mailbox_size_limit* asignando cualquier valor deseado en bytes. Por ejemplo, si se quiere limitar el tamaño de los buzones de correo a 20 MB, por supuesto factorizando por 1024, se utilizaría la siguiente línea:

```
mailbox_size_limit = 20480000
```

message_size_limit

Si se desea limitar el tamaño máximo de un mensaje de correo electrónico, puede hacerse a través de *message_size_limit*. Por ejemplo, si se quiere limitar el tamaño máximo de un mensaje a 5 MB, se utilizaría la siguiente línea:

```
message_size_limit = 5120000
```

recipient_canonical_maps

Si se desea transformar las direcciones de correo externas en direcciones de correo internas, se debe añadir la siguiente línea:

```
recipient_canonical_maps =  
hash:/etc/postfix/recipient_canonical
```

Si por ejemplo, entregar el correo para *webmaster@midominio.com* en la cuenta local *pepe* y *webmaster@dominio.virtual* en la cuenta local *pedro*, dentro

de */etc/postfix/sender_canonical* debería ponerse lo siguiente:

```
webmaster@midominio.com pepe  
webmaster@dominio.virtual pedro
```

Al terminar de editar el fichero, se debe ejecutar el siguiente comando para convertir */etc/postfix/recipient_canonical* en */etc/postfix/recipient_canonical.db* :

```
postmap /etc/postfix/recipient_canonical
```

sender_canonical_maps

Si se desea transformar direcciones de correo internas en direcciones de correo externas, se debe añadir la siguiente línea:

```
sender_canonical_maps = hash:/etc/postfix/sender_canonical
```

Por ejemplo, si se tiene un dominio inválido o no registrado como sería *miredlocal.org* , y se desea que el correo emitido desde la cuenta julio salga como soporte@midominio.com, y ss desea que el correo emitido por pablo salga como soporte@dominio.virtual, */etc/postfix/recipient_canonical* debería ponerse lo siguiente:

```
julio soporte@midominio.com  
pablo soporte@dominio.virtual
```

Al terminar de editar el fichero, se debe ejecutar el siguiente comando para convertir */etc/postfix/sender_canonical* en */etc/postfix/sender_canonical.db* :

```
postmap /etc/postfix/sender_canonical
```

Manejo del Spam y correo proveniente de fuentes indeseables.

check_client_access

Hay varias formas de hacer frente al Spam y correo proveniente de fuentes indeseables. Una es a través del parámetro *check_client_access* , con el cual se define la

localización del fichero que contendrá las tablas de de control de acceso.

```
check_client_access = hash:/etc/postfix/access
```

/etc/postfix/access contendrá una lista de direcciones IP, nombres de dominios, nombres de equipos y direcciones de correo electrónico. Puede definirse si se acepta o de rechaza explícitamente el correo proveniente de los elementos de dicha lista. A continuación se muestran distintos ejemplos:

El valor "**OK**" especifica aceptar explícitamente correo.

```
amigo@algundominio.com OK
otroamigo@otrodominio.com OK
```

El valor "**REJECT**" rechaza todo el correo proveniente y dirigido hacia las direcciones especificadas:

```
#Rechaza correo de las siguientes direcciones y dominios
spammer@productoinutil.com REJECT
dominiohostil.net REJECT
enviosmasivos.com REJECT
masivo.com REJECT
209.61.129.8 REJECT
```

```
#rechaza el correo proveniente de todo el segmento
172.16.18
172.16.18 REJECT
```

Cualquier código "**5xx**" significa error fatal e indica al cliente de correo electrónico no intentar de nuevo. A diferencia de **REJECT**, que rechaza la entrega del mensaje, un código **550**, por ejemplo, rebota el mensaje incluyendo una parte del contenido original.

```
proveedor@necio.com 550 No me interesan sus productos
holgazan@flojo.com 554 Ponte a hacer algo util
```

Cualquier código "**4xx**" significa que el cliente de correo electrónico debe reintentar más tarde.

```
lista-de-correo@undominio.org 450 Estamos realizando
mantenimiento
```

smtpd_client_restrictions y listas negra de servicio de nombres de dominio

(DNSBL)

Anteriormente se había establecido la variable `smtpd_client_restrictions` con los valores `permit_sasl_authenticated` `permit_mynetworks` `ycheck_relay_domains` , los cuales son necesarios para un nivel de seguridad aceptable si se utiliza autenticación para el servicio de SMTP . Pueden añadirse más restricciones, como por ejemplo `reject_maps_rbl`, que especifica la consulta de **listas negras** ofrecidas por distintos servicios como `mail-abuse.org`, `spamhaus.org` o `spamcop.com`. Se requiere entonces añadir `reject_maps_rbl` a la variable `smtpd_client_restrictions` :

```
smtpd_client_restrictions =  
permit_sasl_authenticated,  
permit_mynetworks,  
check_relay_domains,  
reject_unknown_client, reject_maps_rbl
```

A continuación se añaden las definiciones para las listas negras que se desee utilizar o las que se hayan contratado:

```
maps_rbl_domains =  
relays.ordb.org,  
opm.blitzed.org,  
list.dsbl.org,  
sbl.spamhaus.org,  
spamhaus.relays.osirusoft.com,  
relays.mail-abuse.org,  
cbl.abuseat.org
```

Agregar muchas listas ciertamente garantiza un mínimo de Spam, pero significa también que el correo tardará mucho más tiempo en validarse y entregarse en los buzones de correo. Pude llegar a demorar hasta varios minutos después de el envío desde el cliente de correo electrónico. Recomendamos elegir dos o tres listas negras a lo sumo.

smtpd_helo_required y **smtpd_helo_restrictions**

Por lo general los programas utilizados para enviar spam nunca envían comando **helo** al conectarse al servidor, o bien lo hacen desde servidores sin un registro MX o A en un DNS. Por tal motivo es buena idea habilitar las restricciones

que negarán el acceso a cualquier cliente de correo que no envíe comando **helo**

```
smtpd_helo_required = yes  
smtpd_helo_restrictions = reject_unknown_hostname
```

Nota: Es importante no caer en el error común de confundir `smtpd_helo_restrictions = reject_unknown_hostname` con `smtpd_client_restrictions = reject_unknown_client` .

Bibliografía.

Hatch, Brian; **Filtering E-Mail with Postfix and Procmail**
;Security Focus ;<http://www.securityfocus.com/infocus/1593>
Ben Koette, Patrick; **Postfix SMTP AUTH (and TLS)**
HOWTO ;Postfix Howtos, Guides and Tipps by Ralf
Hildebrandt and Patrick Koetter
;<http://postfix.state-of-mind.de/patrick.koetter/smtpauth/>

Referencias.

RFC2821 : Simple Mail Transfer Protocol (SMTP).
RFC2060 : Internet Message Access Protocol (IMAP)
Version 4rev1.
RFC1939 : Post Office Protocol Version 3 (POP3).