

# Cómo utilizar SASL con Postfix.

## 1.- Introducción.

La autentificación 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 posibilidad de envío de correo hacia el exterior solamente a los usuarios de una intranet. Los servidores externos sólo podrían enviar correo cuando el destino final es nuestra intranet. No obstante ese es un mecanismo que solamente sirve cuando los potenciales usuarios están 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 sólo usuarios con una determinada dirección de correo (un atacante podría suplantar nuestra identidad) .

El mecanismo de identificación SMTP después de POP se basa en que para recoger 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 través 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](#). Postfix es un servidor libre (y gratuito), rápido, seguro, flexible y sobre todo fácil de configurar. Recomiendo sinceramente un cambio desde [Sendmail](#) a 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 propias bases de datos de usuarios/contraseñas. Hay que crear un conjunto de usuarios/contraseñas para ser utilizadas por el servidor smtpd de postfix.

Esta primera versión del documento es para usuarios de SuSE que pueden conseguir esa funcionalidad de una forma muy sencilla. Los usuarios de otras distribuciones tendrán que realizar pasos algo distintos. No obstante, los puntos 2.3 y siguientes son aplicables a todas las distribuciones, variando si acaso el directorio donde se encuentran los ficheros de configuración.

## 2.- Instalación a través de RPM. (SuSE).

La instalación de la funcionalidad a través de SuSE es relativamente sencilla, aunque actualmente existen algunos problemas con los scripts de configuración . El autor se basa en una distribución SuSE 8.0.

### 2.1 - Paquetes necesarios. Instalación básica.

Lo primero es instalar los paquetes RPM que necesitamos. Estos son :

```
bash
glibc
heimdal-lib
openssl
db
```

gdbm  
openldap2-client  
pam  
pcre  
cyrus-sasl  
postfix

Normalmente, la mejor forma es utilizar YAST2 para instalar Postfix. En ese caso, YAST2 nos informará de que la instalación de Postfix entra en conflicto con Sendmail y que debemos elegir uno u otro. Si elegimos Postfix, entonces desinstala sendmail y realiza la configuración básica de postfix.

## 2.2 - Configuración preliminar.

Lo primero que hay que hacer es editar algunos ficheros en **/etc/sysconfig** o bien utilizar YAST2 para hacerlo.

De forma directa. En el fichero **/etc/sysconfig/mail** comprobar que

```
MAIL_CREATE_CONFIG = "yes"  
SMTPD_LISTEN_REMOTE = "yes"
```

En el fichero **/etc/sysconfig/postfix** deberemos fijar

```
POSTFIX_RELAY_HOST = " "
```

si todo el correo debemos mandarlo a algún otro servidor, entonces poner ahí el nombre de ese servidor, por ejemplo "mailserver.mi\_dominio.org". Lo habitual, puesto que estamos intentando configurar un servidor seguro, es dejarlo en blanco.

```
POSTFIX_MASQUERADE_DOMAIN = " "
```

aquí se introduce una lista de dominios en los que postfix camuflará toda la estructura de subdominios. Por ejemplo si introduzco "mi\_dominio1.org mi\_dominio2.org" entonces las direcciones como <usuario@maquina3.mi\_dominio1.org> se camuflaran como <usuario@mi\_dominio1.org>.

```
POSTFIX_LOCAL_DOMAINS = " "
```

Introducir aquí la lista de dominios para los que postfix aceptará correo (normalmente nuestro dominio o dominios).

```
POSTFIX_DIALUP = "no"
```

Aquí suponemos que nuestro servidor está conectado permanentemente a la red. En caso de conexión ocasional vía modem habría que fijar "yes".

```
POSTFIX_NODNS = "no"
```

Postfix realiza consultas DNS para traducir direcciones de dominio a IP. Si utilizamos Postfix para difundir solamente correo local y la conexión al servidor DNS no es permanente (como en el caso de DIALUP), entonces la difusión del correo se podría bloquear. En el supuesto caso que nos ocupa dejar el valor "no".

```
POSTFIX_CHROOT = "no"
```

```
POSTFIX_UPDATE_CHROOT_JAIL = "no"
```

Lo mejor es dejar esas variables en "no". El uso de chroot para los servicios de Postfix debe dejarse a usuarios avanzados.

Se pueden cambiar algunas de esas variables (no todas) mediante **yast2 -> sistema -> editor\_de\_sysconfig -> mail -> postfix**

Una vez cambiados en su caso algunos de los valores indicados anteriormente, como root ejecutar **SuSEconfig**

```
#) /sbin/SuSEconfig
```

Tras esto se creará una configuración básica de Postfix. Si queremos que postfix funcione como servicio del sistema al arrancar, deberemos configurar los niveles de ejecución.

**yast2 -> sistema -> editor\_de\_niveles\_de\_ejecución -> editar detalles**

aquí comprobaremos si el postfix está activado, y bajo qué niveles. Lo habitual es habilitar postfix para los niveles **3 y 5**.

Si queremos que nuestro servidor comience a trabajar ya, si no lo está haciendo, tras salir de yast2

```
#) rcpstfix start
```

NOTA: Todavía no funcionará la identificación SASL. Para conseguirlo hay que realizar algún trabajo extra más (leer siguiente apartado).

## 2.3 - Configuración SASL.

En este apartado veremos cómo configurar la librería 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 **saslpaswd**.

```
#) saslpaswd -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**. **TRUCO:** 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.

## 2.4 - 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 restricciones. 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
```

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 . De forma muy resumida y no exhaustiva:

1) Los filtros de Postfix estan organizados en forma de listas de restricciones. 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 restricción 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 restricción 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.

Otras restricciones, que consultan determinados ficheros, pueden dar como resultado cualquiera de las tres.

Por último **check\_relay\_domains** da como resultado OK o REJECT, nunca DUNNO.

Existen, por supuesto, restricciones 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 es:

```
smtpd_client_restrictions  
smtpd_helo_restrictions  
smtpd_sender_restrictions  
smtpd_recipient_restrictions
```

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 Postfix 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

```
#) rpostfix reload
```

hará que postfix trabaje con la nueva configuración.

## **2.5 - Configuración final.**

Una vez se ha modificado el fichero **main.cf** como se indica en el apartado anterior, es conveniente ejecutar, en este orden, los siguientes scripts finales.

```
#) SuSEconfig  
#) rpostfix restart
```

Si todo ha ido bien, la identificación mediante SASL debe funcionar.

Cualquier sugerencia o comentario para mejorar este manual es bienvenida. [Contacta conmigo](#)

---

Guillermo Ballester Valor  
email [gbv@oxixares.com](mailto:gbv@oxixares.com)  
Ogijares, Granada, España