



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Cómo montar un potente sistema de correo (II) (35977 lectures)

Per **Daniel Lombraña González**, [televinex](http://www.sleon.org) (<http://www.sleon.org>)

Creado el 10/12/2003 22:37 modificado el 10/12/2003 22:37

Este artículo pretende complementar: Como montar un potente sistema de correo con postfix de Javi Polo.

La idea fue completar las partes que él ha dejado como TODO. ¿Qué hay de nuevo? Pues se explica como usar contraseñas encriptadas y almacenadas en mysql, y una remodelación del php que nos proporcionaba Javi Polo.

Configuración de Postfix para el uso de contraseñas encriptadas

La manera de hacer que todo funcione es usando **PAM** con su módulo para mysql y el demonio **saslauthd** para la autenticación. Veamos la configuración. Primero tendremos que ir a este directorio `/etc/postfix/sasl` en él modificaremos el archivo `smtpd.conf` por lo siguiente:

```
pwcheck_method: saslauthd
saslauthd_path: /var/run/saslauthd/mux
```

La explicación es muy sencilla, le indicamos el programa que vamos a usar para autenticar y el camino a ese archivo. Para que todo esto funcione tenemos que tener instalado `sasl2-bin`, sino lo tenemos, lo de siempre:

```
apt-get install sasl2-bin
```

Con eso ya tendríamos hecha parte de nuestra nueva configuración. Ahora vamos a configurar el módulo PAM. Primero instalamos el siguiente paquete: **libpam-mysql**

```
apt-get install libpam-mysql
```

Una vez hecho accedemos a

```
/etc/pam.d
```

y creamos, sino lo tenemos ya, el siguiente archivo:

```
smtp
```

. El contenido del mismo es el siguiente:

```
auth sufficient pam_mysql.so \
user=USUARIO \
passwd=CONTRASEÑA \
host=localhost db=mail \
table=users \
usercolumn=address \
passwdcolumn=crypt \
crypt=1
```

```
auth sufficient pam_unix_auth.so
```

```
account required pam_mysql.so \
user=USUARIO \
passwd=CONTRASEÑA \
```



```

host=localhost \
db=mail \
table=users \
usercolumn=address \
passwdcolumn=encrypt \
crypt=1

account    sufficient          pam_unix_acct.so

```

Bueno con esto tendríamos lo que es la configuración, pero si recordáis la base de datos que creó Javi Polo no contaba con un campo para la clave encriptada. Esto se soluciona insertando un nuevo campo antes del campo **clear**. El campo se llamará **crypt** y en él estará la contraseña encriptada. Yo añadí este campo usando el **phpmyadmin**, un potente gestor de mysql hecho en php. Con ese software es muy sencillo.

Una vez añadido, hay que hacer un par de aclaraciones. Javi Polo ya nos comentaba que él NO quería sacar de la jaula a su postfix. Bien, yo tampoco, por lo tanto vamos a tener problemas con el archivo **mux** antes referenciado y el **mysql.sock**. ¿Por qué? Bien porque esos archivos están fueran de la jaula (chroot) y por lo tanto necesitamos hacerlos accesibles. La manera es muy sencilla siempre y cuando estén en la misma partición tanto

```
/var/run
```

como

```
/var/spool/postfix
```

. La manera de actuar es la siguiente: Accedemos a `/var/spool/postfix/var/run/` y creamos el siguiente directorio

```

mkdir saslauthd
cd saslauthd
ln /var/run/saslauthd/mux

```

Lo que hacemos al final es crear un enlace al demonio **saslauthd** para que Postfix pueda autenticar vía saslauthd. Para mysql ya se explicó en el otro documento.

Esto que acabo de hacer sólo funciona temporalmente y se debe a que si reiniciamos mysql o saslauthd se crean de nuevo tanto mux como mysql.sock con lo que el enlace duro ya no sirve. Hay que automatizarlo. Todavía no lo he hecho pero tiene que ser muy sencillo.

Cambios en el código PHP

Una vez hecho todo esto, solo me queda explicar los cambios hechos al php de Javi Polo. Lo primero darle las gracias por ese código, pues sino tendría que haber empezado desde cero.

Los cambios principales hacen referencia al uso de claves encriptadas. Hasta aquí su código las trataba siempre como claves en claro. Ahora cada vez que se añade o modifica un usuario, la contraseña se encripta, de manera que se guarda la clave encriptada en la base de datos y en el campo **crypt**, pero a su vez guardamos la contraseña en claro en el campo **clear**. Esto se hace así porque he pensado en esos ISPs a los que llaman los usuarios preguntando su clave y siempre se la cambias, así te lo ahorras. He de decir que este software TIENE que estar bajo https, sino nos pueden coger fácilmente la clave. La contraseña en claro SOLO se usa para leerla, nunca para la autenticación con pop3 o smtp.



Otro de los cambios hechos es que cada vez que se crea un usuario se envía un email de bienvenida al usuario, de manera que creamos el maildir y le comunicamos al usuario cómo y dónde puede modificar su contraseña.

Se ha intentado el borrado del maildir cuando se borra un usuario, pero esto todavía no me funciona, creo que es por temas de permisos. Todo se andará.

También se ha modificado el concepto de administrador. Antes se podían añadir administradores que lo podían hacer todo, dar de alta dominios, usuarios, administradores, redirecciones. Ahora se pueden crear dos tipos de administradores:

- administradores de un dominio
- administradores de todos los dominios

Esto se ha hecho así porque generalmente un dominio virtual lo querrá administrar su dueño, por lo que sólo dará altas o bajas de su dominio, así como de las redirecciones. El super administrador lo que puede hacer es crear dominios, borrar dominios y añadir más administradores de dominios.

Con respecto a los dominios se ha añadido una nueva característica: el número de buzones que puede tener un dominio. Esto está hecho porque un ISP puede vender packs de buzones, etc y evidentemente querrá controlar que un administrador no dé de alta 100 buzones si solo tiene contratados 30. En todo momento se refleja el número de buzones que tiene cada dominio para el administrador de ese dominio. También se ha añadido el que no se puede borrar un dominio si hay todavía buzones en ese dominio.

Otra característica que se ha añadido es la paginación de resultados. Esto se ha hecho porque he tenido el problema de tener dominios con más de 2000 usuarios y evidentemente mostrar todos estos usuarios no es elegante, con lo que he hecho una paginación similar a la del google. Más rápido y sencillo.

Por último se ha añadido la búsqueda de usuarios, para agilizar aún más la edición de un usuario. Se pueden hacer búsquedas por email y/o nombre del usuario.

En el tema de diseño, se han añadido colores e iconos para que sea todo más bonito. De todas formas se puede usar w3m o links para navegar y se entiende perfectamente.

Bueno, espero que haya sido de ayuda y que si hay algún fallo se me comunique. Taluek.



El código está [aquí](#)

Enlaces

<http://www.marlow.dk/tech/postfix.php>

<http://www.mandrakesecure.net/en/docs/postfix-sasl.php>

<http://marc.theaimsgroup.com/?l=postfix-users&w=2&r=1&s=mysql+%2B+passwords&q=b>

<http://www.delouw.ch/linux/Postfix-Cyrus-Web-cyradm-HOWTO/html/index.html>

<http://postfix.state-of-mind.de/>

<http://webmail.techie.cc/postfix-toaster.htm>

<http://www.probsd.net/vmail/>

<http://www.high5.net/howto/>

E-mail del autor: teleyinex [_ARROBA_ confluencia.net](mailto:_ARROBA_confluencia.net)

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1938>