



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

Sistema de cuentas de correo virtuales con PostFix, OpenLDAP y Courier

(59692 lectures)

Per **Jesús Roncero Franco**, [golan](http://www.roncero.org) (<http://www.roncero.org>)

Creado el 17/04/2004 02:40 modificado el 17/04/2004 02:40

Normalmente, montar un sistema de correo con linux es una tarea sencilla. Basta instalar postfix o qmail para gestionar los correos del sistema y del dominio que tengamos. Cuando tenemos más de un dominio y queremos tener un control más fino de los usuarios de correo, el sistema se puede llegar a complicar bastante si no hacemos uso de las capacidades de las cuentas virtuales.

En este artículo veremos como utilizar el servidor de correo [Postfix](#)⁽¹⁾, el servidor de directorio [OpenLDAP](#)⁽²⁾ y los servicios de correo de [Courier](#)⁽³⁾ para gestionar el correo, los usuarios y el pop e imap utilizando un potente sistema de cuentas virtuales, en la que la información sobre los usuarios estará almacenada en el directorio LDAP.

Advertencia

Todo este artículo está basado en muchas horas de leer documentación, hacer experimentos y modificar ficheros de configuración hasta ver turbio. No soy ningún experto en LDAP, ni Postfix ni Courier, por lo que no te garantizo que te vaya a funcionar a la primera. A mi me ha funcionado así, lo que no quiere decir que sea una configuración perfecta.

Este artículo presupone ciertos conocimientos sobre PostFix y LDAP. Si estas verde en estos dos aspectos, te recomiendo que leas antes un poco de documentación sobre el tema. En Bulma tienes buenos artículos introductorios. O mira al final del artículo, que encontrarás algunos enlaces interesantes.

La modificación de postfix y de su autenticación te puede hacer perder correo e incluso te puede dejar expuesto a ser un relay abierto en internet. Todas las pruebas que hagas, mejor que las hagas con un sistema que no sea en producción o que esté en una red privada.

En fin, lo de siempre, RTFM.

Agradezco críticas y correcciones a los errores ortográficos, de configuración y de conceptos que tenga este artículo (seguro que los tiene).

Por último, el artículo es extenso y trata muchos temas. Es posible que algunas cosas no estén bien explicadas o se me hayan olvidado, simplemente porque llevo demasiadas cosas ;-)) en la cabeza con este artículo. Intentaré mejorarlo sobre la marcha o solventarlo con los comentarios que salgan en este artículo.

Jesús Roncero Franco (c) 2004

¿Qué es lo que queremos hacer exactamente?

- Vamos a tener un sistema de correo con Postfix, el cual queremos que nos controle el correo local del sistema (correos de las cuentas locales) y que además nos controle el correo de cuentas virtuales (**es decir, cuentas que no existen en el sistema y que, por tanto, no pueden ser usadas para hacer login en la máquina**). Además queremos que nuestro sistema haga de servidor MX secundario de otros dominios, es decir, que almacene el correo de otros dominios en caso de que sus servidores primarios de correo estén inaccesibles.
- Toda la información de los usuarios (cuentas y alias) estará almacenada en un servidor de directorio LDAP, que puede estar en la misma máquina o en otra de la misma red. Además, utilizaremos una herramienta web para la



visualización y administración del servidor LDAP (phpldapadmin), aparte de las típicas herramientas en línea de comandos.

- Un sistema POP3, POP3-SSL, IMAP e IMAP-SSL que utilizará las cuentas virtuales y autenticará contra el servidor LDAP.

Todo ello lo montaremos sobre un servidor GNU/Linux Debian corriendo Woody, con la mayoría de paquetes de woody excepto algunos cambios por comodidad.

Haremos las siguientes suposiciones y declaración de intenciones:

- Dominio principal de la máquina: **correo.prueba.com** (nombre canónico de la máquina y registro A del DNS en internet).
- Dominios virtuales: **bulmeros.com** y **bulmita.com**
- El servidor ldap estará almacenado en la misma máquina y habremos creado un alias (dentro de la intranet) que apuntará a la misma máquina, llamada ldap.dominio.com que no se verá desde internet. Así mismo, este alias se utilizará como dominio virtual de web para acceder al programa de administración de LDAP en php.
- Todo el correo de todos los usuarios de correo virtuales estará almacenado en un directorio del sistema y pertenecerán a un mismo usuario. El correo se almacenará en formato Maildir. Utilizaremos Maildir porque Courier es el formato que soporta.
- El servidor LDAP será ejecutado por un usuario **slapd** para mayor seguridad.

Si se quiere que el servidor LDAP esté accesible desde fuera, deberá estar protegido su acceso y tener cuidado con qué información se muestra.

Instalación y configuración de LDAP

Esta es la parte del proceso más larga y más complicada, en parte por la complejidad que tiene LDAP. En realidad LDAP es muy potente y permite almacenar todo tipo de información en él, pero es una aplicación a la que cuesta *hacerse a ella*. Una vez que lo haces, todo es más fácil e incluso disfrutas con ella (*jamás pensé que diría esto ;-)*).

Compilando e instalando OpenLDAP desde las fuentes

El servidor openldap que se encuentra en debian woody viene compilado sin soporte para utilizar ssl. Para ello debemos recompilar nuestro propio paquete de openldap con soporte para ldaps. Aunque esto no es necesario, está bien hacerlo (podemos instalarlo desde otra fuente o no utilizar ssl).

Lo haremos de la siguiente manera:

```
$ cd
$ mkdir debian-ldap
$ cd debian-ldap
$ apt-get source slapd
$ apt-get build-dep slapd
$ apt-get install libssl-dev
```

El paquete slapd es el que contiene el servidor LDAP en debian cuyo demonio se llama de la misma manera, slapd. El paquete libssl-dev contiene las bibliotecas de desarrollo de SSL para compilar OpenLDAP.

Activamos el SSL:

```
$ cd openldap*
$ vi debian/rules
```

y reemplazamos **--without-tls** con **--with-tls**.

Compilamos los paquetes con

```
dpkg-buildpackage -uc -uc -rfakeroot
```



Una vez hecho este paso, tendremos en el directorio ~/debian-ldap los ficheros .deb que deberemos instalar tal que

```
$ sudo dpkg -i *.deb
```

y reiniciar el sistema LDAP con

```
$ sudo /etc/init.d/slaped restart
```

Como hemos configurado nosotros mismos el paquete con soporte SSL, deberemos controlar que, si sale una nueva versión, no nos machaque el que tenemos instalado. De manera que congelaremos el paquete de esta manera:

```
$ sudo echo "slaped hold" | dpkg --set-selections
```

Pero, debemos de estar **muy al tanto** de posibles fallos de seguridad para poder recompilar otra vez nuestros paquetes.

Instalando OpenLDAP desde paquetes precompilados con soporte SSL

Si no queremos pasar por la molestia de compilar nuestros propios paquetes, podemos descargarlos ya compilados con soporte para SSL y para woody. Para ello, debemos añadir la línea:

```
deb http://www.fs.tum.de/~bunk/debian woody/bunk-1 main contrib non-free
```

al fichero /etc/apt/sources.list.

Deberemos instalar los siguientes paquetes con apt-get:

```
apt-get install libldap2 libldap2-tls slapd
```

que contendrán las bibliotecas y el demonio.

Instalación de programas auxiliares

Es conveniente tener instalado estos programas: ldap-utils: que contiene las utilidades ldapsearch, ldapadd, etc para manejar el directorio desde la línea de comandos.

Configuración de OpenLDAP

Como hemos dicho, configuraremos el demonio de LDAP para que se ejecute con un usuario que no sea root. Normalmente cuando se instala OpenLDAP usando los paquetes de debian, se puede configurar la estructura básica del sistema gracias a debconf, es decir, a las preguntas que nos hará el instalador de debian. Nosotros nos saltaremos ese paso y configuraremos desde cero.

Procedemos con lo siguiente:

```
adduser slapd

chown -R slapd.slaped /etc/ldap
chmod 770 /etc/ldap
find /etc/ldap -type f -exec chmod 440 {} \;
find /etc/ldap -type d -exec chmod 770 {} \;
chown -R slapd.slaped /var/lib/ldap
chmod 750 /var/lib/ldap
rm /var/lib/ldap/*
chown -R slapd.slaped /var/spool/slurpd
rm /var/spool/slurpd/*

cd
/etc/ldap/
```

Con esto habremos cambiado todos los permisos al usuario slapd y borrado las bases de datos actuales.



Generaremos la clave del administrador con

```
$ /usr/sbin/slappasswd -h {CRYPT}
New password:
Re-enter new password:
{CRYPT}F×ThcXLxmMTw.
```

que está cifrada y que utilizaremos en el fichero de configuración slapd.conf que crearemos a continuación.

Un /etc/ldap/slapd.conf básico sería así:

```
# This is the main slapd configuration file. See slapd.conf(5) for more
# info on the configuration options.

# Schema and objectClass definitions
# Incluir en este orden
include      /etc/ldap/schema/core.schema
include      /etc/ldap/schema/cosine.schema
include      /etc/ldap/schema/nis.schema
include      /etc/ldap/schema/inetorgperson.schema
# Este schema lo utilizaremos despues
#include     /etc/ldap/schema/authldap.schema

# Schema check allows for forcing entries to
# match schemas for their objectClasses's
schemacheck  on

# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile      /var/run/slapd.pid

# List of arguments that were passed to the server
argsfile     /var/run/slapd.args

# Where to store the replica logs
repllogfile  /var/lib/ldap/repllog

# Read slapd.conf(5) for possible values
loglevel     256

# Password Format
password-hash {CRYPT}

#####
# ldbm database definitions
#####

# The backend type, ldbm, is the default standard
database     ldbm

# The base of your directory
suffix       "o=bulma,c=es"

# The admin password
rootpw       {CRYPT}F×ThcXLxmMTw. # La clave que obtuvimos antes

# Where the database file are physically stored
directory    "/var/lib/ldap"

# Indexing options
index objectClass eq

# Save the time that the entry gets modified
lastmod on

# The userPassword by default can be changed
# by the entry owning it if they are authenticated.
# Others should not be able to see it, except the
```



```
# admin entry below
access to attribute=userPassword
    by dn="cn=admin,o=bulma,c=es" write
    by anonymous auth
    by self write
    by * none

# The admin dn has full write access
access to *
    by dn="cn=admin,o=bulma,c=es" write
    by * read
```

Como vemos, utilizaremos una estructura base en el LDAP así: o=bulma, c=es, sobre la que guardaremos todos los datos del LDAP. Tendremos un usuario administrador con nombre distinguido "cn=admin, o=bulma, c=es" que será el que pueda acceder a todo el directorio.

Esquema del directorio LDAP

El esquema que utilizaremos para el LDAP va a ser un árbol en el que la raíz estará compuesta por la organización bulma (o=bulma,c=es) y a partir del cual tendremos dos *organizationalUnits* que se encargarán de almacenar toda la información pertinente:

- Una rama **People** que contendrá información de las cuentas de usuario. Aquí, de momento, almacenaremos todos los datos obligatorios sobre las cuentas: direcciones de correo, directorio maildir, etc.
- Una rama **Postfix** que contendrá información necesaria para postfix. En esta rama crearemos un hijo para almacenar los alias de correo, de manera que podamos tener varias direcciones que apunten a una misma cuenta o a otra cuenta externa.

Introducción del esquema inicial

Vamos a crear un fichero LDIF que contendrá el esquema inicial del directorio LDAP. Esto lo haremos utilizando el siguiente fichero y el comando ldapadd.

```
# bulma, es
dn: o=bulma,c=es
objectClass: organization
o: subnet

# admin, bulma, es
dn: cn=admin, o=bulma,c=es
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: admin
description: LDAP Administrator
userPassword: {CRYPT}FxFhcXLxmMTw.

# People, bulma, es
dn: ou=People, o=bulma,c=es
objectClass: organizationalUnit
ou: People

# Postfix, bulma, es
dn: ou=Postfix, o=bulma, c=es
ou: Postfix
objectClass: organizationalUnit
userPassword: {CRYPT}FxFhcXLxmMTw.

# Alias, Postfix, bulma, es
dn: ou=Alias,ou=Postfix, o=bulma,c=es
objectClass: organizationalUnit
```

Si os fijáis, le hemos introducido una clave al ou=Postfix. Esto es por si queremos restringir esa información, pero de momento no la usaremos.



Para añadir esa información al servidor utilizaremos el siguiente comando:

```
ldapadd -x -D "cn=admin, o=bulma, c=es" -W -h ldap.dominio.com \
-f fichero.ldif
```

Configuración de phpldapadmin

[Phpldapadmin](#)⁽⁴⁾ es tan fácil de instalar como bajarse el fichero tar.gz de su web, descomprimirlo en un directorio en el servidor web, y modificar el fichero config.php que trae. En él hay que añadir la información de nuestro servidor ldap, métodos de autenticación, etc.

Configuración del sistema de correo

Veámos en esta sección cómo configurar Postfix para que trate con LDAP.

Configuración del directorio donde almacenar los correos

Como dijimos, tendríamos un directorio dónde almacenar todos los correos de los usuarios. Para ello utilizaremos el directorio /var/vmail de la siguiente manera:

```
/var/vmail/dominio1.com/usuario1
/var/vmail/dominio1.com/usuario2
/var/vmail/dominio2.com/usuario1
/var/vmail/dominio2.com/usuario2
/var/vmail/dominio2.com/usuario3
```

y pertenecerán al usuario vmail del grupo vmail. Este usuario tendrá un uid y gid alto para no mezclarse con los usuarios locales del sistema. Utilizaremos el 5000. Además, al directorio /var/vmail le daremos los mismos permisos que /var/mail. Veámoslo:

```
# addgroup --gid 5000 vmail
# adduser --ingroup vmail --uid 5000 vmail
# mkdir /var/vmail
# chmod 2775 /var/vmail
# chown vmail.vmail /var/vmail
```

Nota: si quisiésemos que el usuario fuese del sistema, lo añadiríamos con el comando adduser --system, y habría que utilizar su uid y gid en vez de 5000.

De forma que todo el sistema funcionará de la siguiente forma: El postfix cuando recibe un correo consultará con el servidor LDAP si es un correo para uno de sus dominios y usuarios o si es para uno de los alias, en cuyo caso se encargará del mensaje con lo que le indique el directorio LDAP (reenviarlo o almacenarlo en el HD). En caso contrario, el mensaje será rechazado.

Introducción de un usuario de correo

Vamos a crear un usuario de correo de uno de nuestros dominios para que lo almacene correctamente en su sitio dentro de /var/vmail. Para ello necesitamos configurar el OpenLDAP con un esquema específico que viene en el paquete courier-ldap. Hay que coger el fichero /usr/share/doc/courier-ldap/authldap.schema que contiene las definiciones de las clases necesarias para manejar el correo. Haremos:

```
cp /usr/share/doc/courier-ldap/authldap.schema /etc/ldap/schema/
chmod 440 /etc/ldap/schema/authldap.schema
chown slapd.slapd /etc/ldap/schema/authldap.schema
```

y descomentamos la línea del fichero /etc/ldap/slapd.conf que dejamos comentada anteriormente, para incluir este fichero. Después reiniciamos slapd.

Para añadir un usuario al sistema utilizaremos esta plantilla:

```
dn: cn=Nombre Apellido,ou=People, o=bulma,c=es
```



```

cn: Nombre Apellido
gidNumber: 5000
mail: login@dominio.com
sn: Apellidos
uidNumber: 5000
objectClass: couriermailaccount
objectClass: person
objectClass: top
objectClass: inetOrgPerson
homeDirectory: /var/vmail
quota: 0
mailbox: dominio.com/login/
userPassword: {CRYPT}clave

```

Siendo:

- **dn**: El nombre distinguido, que utilizará el cn (*common name*) como identificador.
- **gidNumber, uidNumber**: Los gid y uid del usuario. En realidad no los necesitamos, pero como authldap.schema indica que son obligatorios los ponemos. Podrían quitarse si se modificase authldap.schema.
- **mail**: el correo electrónico.
- **sn**: Los apellidos (*surname*).
- **objectClass**: las clases que usaremos son estas tres:
 - ◆ courierMailAccount: especifica los atributos relacionados con el correo.
 - ◆ person e inetOrgPerson: Clase que especifica los atributos de una persona. Usamos pocos de ellos, pero se pueden rellenar a gusto con información extra (tfno, dirección, etc.)
 - ◆ top
- **homeDirectory**: indica el directorio a partir del cual se va a acceder relativamente a los buzones.
- **quota**: Atributo que originalmente se usa para controlar una cuota de usuario. Nosotros lo vamos a utilizar para indicar si una cuenta está activa o no, reservándonos el posible uso de limitación de cuota para el futuro. Para ello, si tenemos el valor -1 la cuenta estará desactivada y en cualquier otro caso, activada.
- **mailbox**: camino relativo dónde se va a almacenar los mensajes. Notas:
 1. El directorio se almacenará de forma dominio.com/usuario/
 2. El directorio debe estar creado con prioridad a ser usado.
 3. Debe acabar en "/" para indicar que es formato Maildir.
- **userPassword**: Una clave obtenida con el comando slappasswd.

Por ejemplo, si quiero añadir la dirección de correo: jesus@bulmeros.com tendría este archivo:

```

dn: cn=Jesus Roncero,ou=People, o=bulma,c=es
cn: Jesus Roncero
gidNumber: 5000
mail: jesus@bulmeros.com
sn: Roncero Franco
uidNumber: 5000
objectClass: couriermailaccount
objectClass: person
objectClass: top
objectClass: inetOrgPerson
homeDirectory: /var/vmail
quota: 0
mailbox: bulmeros.com/jesus/
userPassword: {CRYPT}Clave

```

y lo añadiría de manera habitual con el comando ldapadd:

```

ldapadd -x -D "cn=admin, o=bulma, c=es" -W -h ldap.dominio.com \
-f cuenta.ldif

```

De esta manera podemos añadir todas las cuentas que queramos.

Creación de los directorios /var/vmail



Los subdirectorios de /var/vmail deben de pertenecer al usuario vmail y estar creados con anterioridad a que se vaya a usar. Para ello se puede usar el [siguiente script para crearlos](#)⁽⁵⁾:

```
#!/bin/sh
# Program to create virtual mail folders
# (c) Jesús Roncero Franco jesus (at) roncero.org 2004
# This program is GPL
#
# Depends on:
#         - ldapsearch
#         - maildirmake ( from courier )
#
# Based on http://jeroen.protheus.com/postfix-courier-ldap-howto.html
# (c) J.Vriesman

# Password to bind to ldap server
systempass="password"
# Bind dn
binddn="ou=Postfix,o=bulma,c=es"
# Account leave
accountleave="ou=People,o=bulma,c=es"
# Vmail parent dir
vmaildir="/var/vmail/"
# ldap host
ldaphost="ldap.dominio.com"

ldappersonaldirs=`ldapsearch -h $ldaphost -x -w $systempass -D "$binddn" \
    -b "$accountleave" "(!(quota=-1))" \
    mailbox | grep "^[^#]" | grep mailbox | awk '{ print $2 }'`

# create personal mailfolders

for ldappersonaldir in $ldappersonaldirs
do
    if [ ! -d $vmaildir/$ldappersonaldir ]
    then
        # debug
        #echo $vmaildir/$ldappersonaldir
        mkdir -p `dirname $vmaildir/$ldappersonaldir`
        maildirmake $vmaildir/$ldappersonaldir
    fi
done
```

Este programa creará los directorios de los usuarios si no existen. Necesita el programa maildirmake del paquete courier para crear un directorio de tipo Maildir.

Recuerda que cada vez que añadamos un usuario al sistema, habrá que ejecutar este script como el usuario **vmail**.

Creación de un alias de correo

Para crear un alias de correo, tendremos que introducir un elemento dentro de la hoja ou=Alias, ou=Postfix, o=bulma, c=es e indicar qué hacer. Como en el ejemplo anterior, se puede utilizar la plantilla siguiente:

```
dn: mail=alias@dominio.com,ou=Alias,ou=Postfix, o=bulma,c=es
mail: alias@dominio.com
maildrop: direccion@destino.com
objectClass: couriermailalias
objectClass: top
```

dónde:

- Utilizaremos como **dn** el propio atributo mail, usádo la sintaxis típica.
- **mail**: la dirección de alias de uno de los dominios virtuales que tenemos.
- **maildrop**: la dirección de destino. Puede ser uno de las direcciones de dominios virtuales que tenemos u otra en otro sitio. Si tenemos más de una línea con maildrop, el mensaje se enviará a todas esas direcciones.



- **objectClass: courierMailAlias**: Clase que especifica que es de tipo alias.

Por ejemplo:

```
dn: mail=criticas@bulmeros.com,ou=Alias,ou=Postfix, o=bulma,c=es
mail: criticas@bulmeros.com
maildrop: bill@microsoft.com
maildrop: jesus@bulmita.com
objectClass: couriermailalias
objectClass: top
```

y lo añadiríamos con el comando típico `ldapadd`, tal como vimos antes.

Configuración del Postfix

Llegados a este punto estamos preparados para modificar la configuración del sistema de correo postfix. Os recomiendo que tengáis una terminal monitorizando `/var/log/mail.info` para saber en todo momento que está pasando. En concreto os recomiendo el paquete `colorize` (`apt-get install colorize`) y :

```
tail -f /var/log/mail.info | colorize
```

Utilizaremos la versión 2.x de Postfix por estar mejor preparada para este tipo de soportes y tener mejor control del relay de correo. Todo esto se puede hacer con la versión de woody (1.1.11), pero a mi me ha costado más trabajo y no he sido capaz. Así que utilicé una versión compilada sacada a partir de esta línea para el `/etc/apt/sources.list`:

```
deb http://people.debian.org/~hnh/woody/ hnh/postfix/
```

Debemos instalar los paquetes: `postfix`, `postfix-ldap`, `postfix-doc`, `postfix-pcre`.

Configuración de postfix para los dominios virtuales

Explico a continuación cuales son las declaraciones clave en el fichero `/etc/postfix/main.cf` con respecto al tema de dominios virtuales:

```
#Dominios Virtuales
virtual_maps = ldap:valiases
virtual_transport = virtual
virtual_mailbox_base = /var/vmail/
virtual_mailbox_maps= ldap:ldapvirtualmap
virtual_mailbox_domains = $virtual_mailbox_maps hash:/etc/postfix/vmaildomains
virtual_minimum_uid = 100
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000
```

- **virtual_maps**: Tabla con los alias.
- **virtual_transport**: Especificamos el agente de transporte por defecto que se encargará de los dominios definidos en `$virtual_mailbox_domains`.
- **virtual_mailbox_base**: aquí especificaremos el directorio a partir del cual tendremos las distintas cuentas de correo en subdirectorios (dominio/usuario/)
- **virtual_mailbox_maps**: Aquí se buscan los destinatarios de los mensajes junto con el path relativo dónde se encuentra los directorios Maildir dónde almacenar los mensajes. Es decir, las direcciones de correo válidas, indicando además el camino relativo a partir de `virtual_mailbox_base`.
- **virtual_mailbox_domains**: Aquí se especifican los dominios virtuales que vamos a gestionar.
- **virtual_minimum_uid**: El uid mínimo que el sistema aceptará como retorno de una búsqueda en `$virtual_uid_maps`.
- **virtual_uid_maps**: El uid usado para escribir en el mailbox.
- **virtual_gid_maps**: El gid usado para escribir en el mailbox.

Como vemos, en alguno de ellos utilizamos el ldap para buscar los datos (`ldap:nombre`) y en otros usamos un fichero local (`hash:/fichero`).



En este caso, tendremos el fichero `/etc/postfix/vmaildomains`, que contendrá los dominios para los cuales tenemos cuentas virtuales. Deberá tener una estructura similar a esta:

```
bulma.net      requerido
dominio2.com   requerido
```

Donde tendremos una línea por cada dominio virtual. Para que postfix reconozca estos dominios, habrá que hacer:

```
postmap vmaildomains
```

para crear la tabla hash correspondiente.

Para el caso de las búsquedas ldap, cuando especificamos `ldap:nombre`, *nombre* va a ser el prefijo que se va a utilizar en una serie de variables de postfix para especificar la configuración e interrogación del ldap. Así, por ejemplo, para la sentencia `ldap:nombre`, habrá que definir las siguientes líneas:

- **nombre_server_host**: Servidor de LDAP.
- **nombre_search_base**: Base de búsqueda. A partir de dónde hay que buscar en el LDAP.
- **nombre_query_filter**: Filtro para la búsqueda. Especificamos qué tipo de restricciones tiene lo que estamos buscando.
- **nombre_result_attribute**: Atributos que queremos leer de los resultados de la búsqueda.
- **nombre_bind**: ¿Debe autenticarse o no? Especificamos no porque esa parte del directorio LDAP es accesible anónimamente.

Entonces, por ejemplo, para configurar los alias virtuales, tendríamos una cosa como esta:

```
#Alias virtuales
virtual_maps = ldap:valiases
valiases_server_host = ldap.dominio.com
valiases_search_base = ou=Alias,ou=Postfix,o=bulma,c=es
valiases_query_filter = (&(mail=%s)(objectClass=CourierMailAlias))
valiases_result_attribute = maildrop
valiases_bind = no
```

Vemos que en este caso, buscaríamos en `ou=Alias,ou=Postfix,o=bulma,c=es`, para los elementos que cumplan que el atributo **mail** sea igual a la dirección de correo que se busca y que el tipo de *objectClass* sea del tipo `CourierMailAlias`, y por último, se obtiene lo que hay en el atributo *maildrop*.

Nota: `_query_filter` utiliza una notación prefija, tal como podemos ver, del tipo `(&(condicion)(condicion))`. Otras opciones podrían ser:

- `(!(condicion)(condicion))`
- `(&(!(condicion)(condicion))(condicion))`
- etc.

Cuando veámos el fichero de configuración final, lo entenderemos bien.

Configuración de Postfix para la entrega local

Como también queremos que la entrega local sea tratada en el propio sistema a través de postfix, tenemos que especificarle que se encargue de los correos locales. Si no se habilita esta opción, habría que tratar las cuentas locales del sistema en el sistema virtual, ya que, si no lo hacemos se produciría un error al entregar correo a root, por ejemplo.

Para configurar la entrega local, necesitaremos lo siguiente:

```
local_transport = local
mydestination = $myhostname $localhost.$mydomain localhost.dominio.com
local_recipient_maps = unix:passwd.byname $alias_maps
```

Esto debe ser suficiente para que consiga hacer la entrega local.



Configuración de Postfix para hacer de MX secundario

Si queremos que nuestro servidor de correo actúe como una estafeta de correo secundaria de un tercer dominio, deberemos configurarlo correctamente para que el correo de estos dominios lo acepte, pero no lo trate como si fuese un dominio virtual o local, y para que lo reenvíe al servidor principal. Para ello, debe de haber un registro MX configurado en el DNS del dominio en cuestión que apunte a nuestro sistema de correo, con una prioridad menor que la del sistema principal. Por ejemplo, en el caso de los correos para el dominio bulma.net, tenemos:

```
$ host -t mx bulma.net
bulma.net      MX      10 bergantells.bulma.net
bulma.net      MX      50 pdn.nucli.net
```

Lo que significa que los correos del dominio **bulma.net** los tratará el sistema **bergantells.bulma.net**, mientras que si este falla, el sistema **pdn.nucli.net** se encargará de almacenar temporalmente los correos del dominio bulma.net.

Supongamos que queremos hacer backup secundario al dominio **vger.kernel.org**, tendríamos que tener declarado esto:

```
relay_domains = $mydestination, vger.kernel.org
relay_recipient_maps = hash:/etc/postfix/relay_recipients
```

y en el fichero `/etc/postfix/relay_recipients` lo siguiente:

```
@vger.kernel.org      cualquier_valor
```

Como el fichero va a ser un hash, hay que prepararlo de la siguiente manera:

```
postmap relay_recipients
```

Y con ello debe de funcionar el sistema como MX secundario.

Pasos finales

Os recomiendo muy encarecidamente que monitoriceis el `/var/log/mail.info` para ver en todo momento que funciona como se espera y para corregir los errores que puedan surgir.

Por último, el fichero `/etc/postfix/main.cf` quedaría tal que así:

```
command_directory = /usr/sbin
daemon_directory = /usr/lib/postfix
program_directory = /usr/lib/postfix
setgid_group = postdrop
append_dot_mydomain = yes
smtpd_banner = $myhostname ESMTP $mail_name
biff = no

myhostname = correo.dominio.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
relayhost =
mynetworks = 127.0.0.0/8, 192.168.0.0/24

#Alias virtuales
virtual_maps = ldap:valiases
valiases_server_host = ldap.dominio.com
valiases_search_base = ou=Alias,ou=Postfix,o=bulma,c=es
valiases_query_filter = (&(mail=%s)(objectClass=CourierMailAlias))
valiases_result_attribute = maildrop
valiases_bind = no

#Dominios Virtuales
virtual_transport = virtual
virtual_mailbox_base = /var/vmail/
#virtual_mailbox_maps= hash:/etc/postfix/vmailbox
```



```

virtual_mailbox_maps= ldap:ldapvirtualmap
ldapvirtualmap_server_host = ldap.dominio.com
ldapvirtualmap_server_port = 389
ldapvirtualmap_bind = no
ldapvirtualmap_search_base = ou=People,o=bulma,c=es
ldapvirtualmap_query_filter = (&(mail=%s)!(quota=-1))(objectClass=CourierMailAccount)
ldapvirtualmap_result_attribute = mailbox

virtual_mailbox_domains = $virtual_mailbox_maps hash:/etc/postfix/vmaildomains

virtual_minimum_uid = 100
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000

# Todos los dominios y los usuarios entregados por el agente de entrega local
# local_recipient_maps es usado por el servidor SMTP para rechazar el correo
# de los usuarios no conocidos
local_transport = local
mydestination = $myhostname $localhost.$mydomain localhost.dominio.com
local_recipient_maps = unix:passwd.byname $alias_maps

# relay
relay_domains = $mydestination, vger.kernel.org
relay_recipient_maps = hash:/etc/postfix/relay_recipients

mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +

```

Comentemos éstas líneas:

```

ldapvirtualmap_search_base = ou=People,o=bulma,c=es
ldapvirtualmap_query_filter = (&(mail=%s)!(quota=-1))(objectClass=CourierMailAccount)
ldapvirtualmap_result_attribute = mailbox

```

Esto hace una búsqueda a partir de ou=People,o=bulma,c=es de todos los registros LDAP que contienen la dirección email que buscamos, que su campo quota es distinto de -1 y que objectClass es igual que CourierMailAccount. Se devuelve el contenido de mailbox.

Ejemplos por telnet de envío de correo

Veámos algunos ejemplos por telnet. (En rojo, lo que tecleo yo)

```

jesus@momona:~$ telnet ldap.midominio.com 25
Trying x.x.x.x...
Connected to x.Red-x-x-x.pooles.rima-tde.net.
Escape character is '^]'.
220 ldap.midominio.com ESMTP Postfix
helo prueba
250 ldap.midominio.com
mail from: <jesus@midominio.com>
250 Ok
rcpt to: <jesus@bulmeros.com>
250 Ok
rcpt to: <jose@bulmeros.com>
550 : User unknown in virtual mailbox table
rcpt to: <bill@microsoft.com>
554 : Relay access denied
rcpt to: <leni@vger.kernel.org>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
prueba
.
250 Ok: queued as EB02617B04
quit
221 Bye
Connection closed by foreign host.

```



Instalación y configuración de Courier

Una vez que tenemos instalado todo el sistema de correo y que nos aseguramos de que funciona (los correos se entregan en /var/vmail), sólo tenemos que instalar courier y sus paquetes asociados (pop3, imap, etc.).

Courier utiliza un servicio de autenticación para todos sus servicios. Este servicio se puede configurar de varias maneras, de manera que haga la autenticación de varias fuentes (PAM, LDAP, MYSQL, etc.), con lo que una vez configurado este servicio, todos los servicios adicionales courier tirarán de éste para hacer la autenticación (pop3, pop3s, imap, etc). Este servicio está formado por un demonio llamado authdaemon, que se configura en el fichero /etc/courier/authdaemonrc, el cual viene configurado por defecto para autenticar contra PAM. Instalaremos los paquetes siguientes: courier-base courier-authdaemon courier-ldap y courier-pop, para tener soporte ldap y pop3.

Modificaremos los siguientes ficheros:

- /etc/courier/authdaemonrc para que tenga la siguiente línea:

```
authmodulelist="authldap authpam"
```

- /etc/courier/authldaprc para que contenga lo siguiente:

```
LDAP_SERVER          ldap.dominio.com
LDAP_PORT            389
LDAP_BASEDN          ou=People, o=bulma,c=es
LDAP_BINDDN          ou=Postfix, o=bulma,c=es
LDAP_BINDPW          clave
LDAP_TIMEOUT         15
LDAP_AUTHBIND        1
LDAP_MAIL            mail
LDAP_FILTER           (!(quota=-1))
LDAP_GLOB_UID         vmail
LDAP_GLOB_GID         vmail
LDAP_HOMEDIR          homeDirectory
LDAP_MAILDIR          mailbox
LDAP_FULLNAME         cn
LDAP_CRYPTPW          userPassword
LDAP_DEREF            never
LDAP_TLS              0
```

Creo que llegados a este punto, no hace falta explicar para qué significa cada cosa.

Si no es así, lo único que hay que tener en cuenta es que cuando se quiere recoger el correo de un usuario, se utilizará como login la dirección entera de correo, ya que podemos tener dos cuentas con distinto dominio pero igual login.

Veámos una sesión (En rojo lo que tecleo yo):

```
golan@terminus:~$ telnet bulmeros.com 110
Trying 192.168.0.1...
Connected to ldap.dominio.com.
Escape character is '^]'.
+OK Hello there.
user jesus@bulmeros.com
+OK Password required.
pass clavecorreo
+OK logged in.
list
+OK POP3 clients that break here, they violate STD53.
1 411
2 408
3 1213
4 559
5 560
.
quit
+OK Bye-bye.
Connection closed by foreign host.
```



Añadir más servicios (como pop3s o imap) es sólo cuestión de instalar los paquetes. Automáticamente autenticarán contra el ldap :-)

Conclusión

Con esto hemos visto una manera no trivial de montar un sistema completo de cuentas de correo virtuales. Almacenar los datos en el LDAP nos da varias ventajas sobre otros sistemas como MySQL. LDAP está especializado en responder con rapidez a muchas lecturas, se puede replicar el directorio LDAP y, sobre todo, los datos de LDAP se pueden utilizar para muchas otras cosas más, como ser la base de datos central de claves de todo un sistema (login, web, correo, etc.).

LDAP es un poco difícil de entender, pero una vez que te superas la curva de aprendizaje te darás cuenta de lo versátil y potente que es/puede llegar a ser.

Espero que, al menos, hayas podido aprender de todo esto y que os sirva el artículo, porque yo cuando empecé a investigar sobre esto, no me aclaraba mucho de como iba la cosa.

Cosas que quedan por hacer

Hay varias cosas que no están implementadas y que estaría bien tenerlas activadas:

- **Cifrado:** Activar el soporte TLS para las conexiones OpenLDAP y también acceder a phpmldapadmin a través https para evitar que las contraseñas vayan en claro.
- **Dominios en LDAP:** Conseguir que los dominios que hacen relay no estén en un fichero en /etc/postfix, sino en el LDAP. Yo todavía no lo he conseguido. Estoy en ello :-)
- **Quotas:** Conseguir que las cuotas funcionen. De momento habría que aplicar un parche a Postfix o buscar métodos alternativos.
- **Scripts:** Hacer una serie de scripts para modificar los datos y que no se tenga que hacer el acceso a través de phpldapadmin.

Más lecturas

Obviamente, esto ha sido el esfuerzo de muchos días de búsqueda por internet, de lectura de manuales y de páginas web. No hay un sitio concreto para mirar toda la documentación, así que os dejo una serie de enlaces con artículos en los que me he basado o he leído, así como una serie de lugares con documentación.

- [Postfix](#)⁽⁶⁾: El sitio web de Postfix y su documentación en /usr/share/doc/postfix (contenida en el paquete debian postfix-doc).
- [OpenLDAP](#)⁽⁷⁾: El sitio web de OpenLDAP.
- [Configuración de cuentas y samba con LDAP](#)⁽⁸⁾: Cómo configurar el equipo para que la autenticación local la haga por LDAP.
- [Postfix Courier LDAP Howto](#)⁽⁹⁾: Un howto para montar un sistema similar al que describo aquí aunque de distinta manera. Aunque este sistema lo monté inicialmente, me dió algunos problemas, porque no me funcionaba junto con el MX secundario, y lo descarté.
- [Understanding LDAP Directories](#)⁽¹⁰⁾: Un RedBook de IBM sobre LDAP en general (Recomendable lectura).
- Los artículos de Bulma sobre LDAP, correo y Postfix

Agradecimientos

Este artículo no habría sido posible sin la ayuda de varias personas.

- A [Elena González](#)⁽¹¹⁾, por su ayuda en el proceso (betatester oficial ;-)) y en las configuraciones.
- A [Carlos Perelló](#)⁽¹²⁾, por su ayuda con las configuraciones.
- A Rafa Martín, de ADALA.
- A la gente de la lista postfix-es

Lista de enlaces de este artículo:



1. <http://www.postfix.org/>
2. <http://www.openldap.org/>
3. <http://www.courier-mta.org>
4. <http://phpldapadmin.sourceforge.net/>
5. <http://roncero.org/variost/ldap/makevirtualdirs>
6. <http://www.postfix.org>
7. <http://www.openldap.org>
8. <http://aqua.subnet.at/~max/ldap/>
9. <http://jeroen.protheus.com/postfix-courier-ldap-howto.html>
10. <http://www.redbooks.ibm.com/abstracts/sg244986.html>
11. <http://www.vampipollo.com>
12. <http://carlos.pemas.net>

E-mail del autor: jesus_ARROBA_roncero.org

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2013>