



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

Instalación y configuración de OpenLDAP (85248 lectures)

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

Creado el 30/05/2002 18:48 modificado el 30/05/2002 18:48

OpenLDAP es un servicio de directorio que, entre otras cosas, nos permite contener los datos (logins, claves) de una serie de usuarios y realizar la autenticación en máquinas clientes a través de un único servidor OpenLDAP.

En este artículo veremos como instalar un servidor OpenLDAP para realizar este tipo de autenticación y en un próximo artículo veremos cómo configurar los clientes.

Disclaimer *Todo esto ha sido fruto de lo que he aprendido durante mis prácticas en empresa en la Escuela de Ingenieros en Informática de Sevilla, <http://www.eii.us.es>⁽¹⁾, dónde tuvimos que implementar un sistema de autenticación basado en OpenLDAP para que los clientes, [rembo](#)⁽²⁾, windows y linux, utilizasen el directorio LDAP y no sus ficheros locales. Ya que OpenLDAP es extenso y dado que tuvimos que dedicarnos a solventar varios tipos de problemas en la implementación, este artículo es una recopilación de todo lo que averiguamos durante las prácticas, más que nada para tener constancia de todo lo que hicimos y para ayudar a otra gente que se encuentre con el mismo problema.*

No soy un experto en OpenLDAP y puede que, para cuando leas este artículo, ya me haya olvidado de muchas cosas sobre OpenLDAP ;-). Así que aquí queda esto como pequeña guía rápida.

Introducción al sistema de directorio

Un directorio es, básicamente, una base de datos que está optimizada para lecturas, dónde se van a producir pocas escrituras. Un directorio es, por ejemplo, el DNS, dónde el número de consultas es mucho mayor que el de adiciones o modificaciones. Los típicos usos que se le dan incluyen el almacenamiento de la información del usuario, claves, direcciones de correo, etc...

El protocolo DAP, X.500, es el protocolo de acceso a directorio de la pila OSI. El LDAP, del inglés *Lightweight Directory Access Protocol*, es un protocolo que está implementado sobre la pila TCP/IP y que surge como versión ligera al DAP, ya que este último requiere bastantes más recursos que LDAP.

Tenemos varias alternativas a la hora de elegir un servidor LDAP, entre ellos los creados por la Universidad de Michigan y el servidor LDAP de Netscape. OpenLDAP es la versión libre de este tipo de servidores.

En este artículo veremos como hacer una instalación simple de LDAP y cómo configurarlo para que use transmisiones seguras gracias a las librerías OpenSSL. Si quieres más información sobre LDAP, puedes consultar el [RFC2251](#)⁽³⁾.

Entrada en materia.

La información que se suele guardar en un directorio está formada por entradas. Cada entrada es un conjunto de atributos (información) que se identifica a través de un nombre distinguido, DN, *Distinguished name* en inglés. Cada uno de estos atributos es de un tipo concreto y puede tener uno o varios valores. Estos atributos suelen ser nemónicos como por ejemplo *cn* (*common name*) o *mail* que indicaría el nombre o email de una entrada en el directorio.

La información se almacena en el directorio en forma de árbol jerárquico. Tradicionalmente se almacenaban utilizando como raíz los distintos países, siendo los hijos las distintas provincias, e hijos de estos las distintas organizaciones o compañías que podía haber en una provincia.

También se puede utilizar la jerarquía de internet para definir la estructura de un directorio LDAP utilizando, por



ejemplo, los nombres de dominio. Así, la raíz podría ser **net**, un hijo, **bulmalug** y dentro de este los diversos usuarios que pudiera tener, etc.

LDAP posee un atributo, *objectClass*, que permite definir que datos van a ser obligatorios y cuales opcionales a través de los esquemas (schemas). O sea, los esquemas van a ser como definiciones de los datos que va a contener el directorio. OpenLDAP incluye varios esquemas, pero podremos añadir o modificar los ya existentes para cumplir con nuestras necesidades.

El acceso a la información del directorio, se hace a través de los nombres distinguidos, DN, y de los RDN, nombres distinguidos relativos. Estos se construyen a partir de una entrada, concatenandole el DN de todos sus padres. Por ejemplo, para el ejemplo de bulma, imaginemos que tenemos un autor que tiene de *uid* autor1, su DN sería "uid=autor1, ou=autores, dc=bulmalug, dc=net". Para encontrar más información sobre el formato DN, puedes consultar el [RFC2253](#)⁽⁴⁾.

OpenLDAP permite, además, definir ACL (access control lists) para el acceso a los datos, a parte de que permite transmitir los datos sobre una capa segura como OpenSSL. Así mismo, OpenLDAP permite la replicación de los datos entre varios servidores OpenLDAP para descongestionar los servidores.

Descarga, compilación e instalación de OpenLDAP

Aunque en debian podemos hacer un apt-get directamente, vamos a explicar aquí como instalar a partir de los fuentes, ya que para utilizar la capa segura deberemos recompilar el OpenLDAP con soporte OpenSSL.

Lo primero que haremos es apuntar a <http://www.openldap.org/software/download/>⁽⁵⁾ y descargaremos la última versión que tengamos disponible de uno de sus mirrors. A fecha del artículo, trabajaremos con la versión 2.0.23.

Descomprimiremos el fichero con las fuentes en un directorio con la orden

```
$ tar xvfz openldap-2.0.23.tgz
```

Para configurar el ldap haremos el típico configure, make y make install. El configure lo haremos con las siguientes opciones:

```
$ ./configure --with-tls --enable-debug --enable-cache --enable-referrals --with-threads
--enable-slaped --enable-slurpd
```

--with-tls nos permitirá tener soporte seguro a través de OpenSSL, por lo que deberemos tener instaladas las bibliotecas de OpenSSL en nuestro sistema.

--enable-slaped y **--enable-slurpd** nos crearán los ejecutables para el servidor OpenLDAP y para el encargado de la replicación. Aunque este último no lo trataremos en este artículo, nos puede venir bien tenerlo para un futuro próximo. Puedes ver la documentación para otros tipos de opciones, pero yo creo que con esta es suficiente para lo que tenemos pensado montar.

Despues haremos :

```
$ make depend
$ make
$ cd tests
$ make tests
$ cd ..
$ su
# make install
```

para compilar el software, testearlo e instalarlo en el sistema. Si todo ha ido bien y no ha fallado ni la compilación ni los testeos, tendremos instalados los binarios en /usr/local/libexec y los ficheros de configuración en /usr/local/etc/openldap.

Procedemos ahora a configurar el fichero de configuración del openLDAP, que es /usr/local/etc/openldap/slapd.conf. Este fichero está estructurado en dos partes, una primera con las opciones globales del OpenLDAP y otra con las definiciones de cada base de datos (directorio) que queramos tener. Hay que resaltar que las opciones de este fichero



deben comenzar en la primera columna del fichero, ya que si no lo hacen así, serán considerados continuaciones de la línea anterior.

Veamos pues un fichero `slapd.conf` de ejemplo para definir un directorio que contendrá una base de datos de usuario para un sistema de autenticación:

```
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
include      /usr/local/etc/openldap/schema/core.schema
include      /usr/local/etc/openldap/schema/cosine.schema
include      /usr/local/etc/openldap/schema/inetorgperson.schema
include      /usr/local/etc/openldap/schema/nis.schema
#include     /usr/local/etc/openldap/schema/redhat/autofs.schema
#include     /usr/local/etc/openldap/schema/redhat/rfc822-MailMember.schema
#include     /usr/local/etc/openldap/schema/redhat/kerberosobject.schema
#####

#Los siguientes valores indican la ruta de los pid y los args.
pidfile      /var/run/slapd.pid
argsfile     /var/run/slapd.args

# permisos para nuestra base de datos

access to attr=userpassword
    by self write
    by * read

access to *
    by self write
    by dn="."+ read
    by * read

#####
# DEFINICION DE LAS BASES DE DATOS (DIRECTORIOS)
#####

# Base de datos por defecto a usar ldbm
database ldbm

# Sufijo o raiz(root) del directorio. Es el nodo raiz superior
# de nuestro directorio ldap
suffix "dc=bulmalug, dc=net"

# El nombre distinguido del directorio manager
rootdn "cn=Manager, dc=bulmalug, dc=net"

# No es muy buena idea guardar la contraseña en texto llano,
# pero la dejaremos asi al principio hasta que controlemos mas sobre LDAP
rootpw secret

# Aqui es donde se guardara los datos del directorio
directory /var/lib/ldap
```

Veamos lo que definimos con este fichero de configuración. Primeramente, la parte de configuración global, donde le indicamos los schemas que queremos que siga OpenLDAP por medio de la orden *include*. Definimos los ficheros dónde se guardarán el pid y los argumentos con qué se ha llamado al programa con *pidfile* y *argsfile* y por último, en la sección global, definimos los ACLs para la clave.

A continuación tenemos la definición de la base de datos que vamos a usar. Si queremos tener más de una base de datos podemos añadir más sentencias como estas que vienen a continuación. Veámoslas:

- **database ldbm**: Con esto definimos, a parte del inicio de la configuración de la base de datos, el tipo de base de datos que queremos usar. Lo normal es usar `ldbm`, pero otras opciones son `passwd` y `shell`.
- **suffix "dc=bulmalug, dc=net"**: Con esto definimos la jerarquía que usaremos, o sea, `bulma.net`.
- **rootdn "cn=Manager, dc=bulmalug, dc=net"**: Aquí definiremos quién va a ser el usuario administrador de



esta base de datos, es como el root de un sistema unix.

- **rootpw secret:** Aquí definimos cual va a ser la clave de root, secret en este caso. Vemos que aquí la clave está en modo texto plano, totalmente desaconsejada, pero que, para hacer nuestras pruebas, nos sirve de momento. Después se debe de cambiar a una clave encriptada con crypt, por ejemplo.
- **directory /var/lib/ldap:** Directorio dónde se guardarán todos los datos del directorio LDAP.

Os recomiendo que mireis la página de manual de slapd.conf para que veais todas las opciones que tiene y como proteger **la clave de root**. Si lo que quereis es usar otra jerarquía de internet, podeis cambiar las sentencias rootdn a alguna que se adecue a vuestro caso, por ejemplo: "dc=eii,dc=us,dc=es".

Una vez que tenemos hecho todo esto, podemos proceder a lanzar el demonio. Lo normal es lanzarlo a través de su script en /etc/init.d/, pero vamos a lanzarlo a mano hasta comprobar que todo funciona bien. Lo hacemos con:

```
/usr/local/libexec/slapd -f /usr/local/etc/openldap/slapd.conf
```

Esto lanzará el servidor en modo demonio, pero podemos añadirle la opción "-d n", para obtener el debug por pantalla, siendo n el nivel de debug que queremos.

Para comprobar que el servidor está funcionando, ejecutamos el siguiente comando:

```
$ ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
```

lo que debe devolver debe ser algo parecido a:

```
.....
dn:
namingContexts: dc=bulmalug,dc=net
.....
```

Si has obtenido algo parecido a eso, es que el servidor está funcionando correctamente y ya podemos agregar las entradas iniciales al directorio.

La primera que vamos a añadir es la que va a sostener toda la estructura de los usuarios. O sea, vamos a añadir una *organización* a la base de datos. Esta organización la llamaremos *bulma* y estará sobre "dc=bulmalug,dc=net".

Todo el intercambio de datos con LDAP se hace con unos ficheros con un formato especial. Este formato es el LDIF, *LDAP interchange format*. Puedes encontrar su RFC en [ftp://ftp.isi.edu/in-notes/rfc2849.txt](http://ftp.isi.edu/in-notes/rfc2849.txt)⁽⁶⁾ para tener más información. El fichero que utilizaremos para añadir la entrada original será este, llamemosle prueba.ldif:

```
#####
dn: dc=bulmalug,dc=net
objectclass: dcObject
objectclass: organization
o: bulma
dc: bulmalug

dn: cn=Manager,dc=bulmalug,dc=net
objectclass: organizationalRole
cn: Manager
#####
```

Lo añadiremos al directorio con el comando siguiente, que nos pedirá la clave del Manager:

```
$ ldapadd -x -D "cn=Manager,dc=bulmalug,dc=net" -W -f prueba.ldif
```

Una vez hecho esto ya podemos añadir el resto de datos al ldap. Una vez añadidos los datos, que veremos en la siguiente sección, podemos comprobarlos con este comando:

```
$ ldapsearch -x -b 'dc=bulmalug,dc=net' '(objectclass=*)'
```

que nos tiene que devolver una salida en formato LDIF con todos los datos en que están en el directorio.



Migración de los usuarios al LDAP

La migración de todos los datos de los usuarios nos supondría crear un fichero LDIF a partir de /etc/passwd. Gracias a <http://www.padl.com>⁽⁷⁾ podemos usar una serie de scripts en perl para la fácil migración de estos datos. Estos scripts los puedes encontrar en <http://www.padl.com/OSS/MigrationTools.html>⁽⁸⁾. Con estos podrás crear un fichero ldif que posteriormente podrás añadir al directorio con ldapadd.

Os recomiendo que os mireis la documentación de estos scripts, así como la documentación del LDAP, sin olvidar las páginas man del ldapadd y de ldapsearch

Configuración del servidor en modo seguro

Hasta ahora lo que hemos hecho ha sido configurar un servidor OpenLDAP para que funcione por el puerto estándar, es decir, el puerto 389, que realiza las conexiones sin cifrar. Para configurar el servidor en modo seguro, debemos tenerlo funcionando correctamente por el puerto 389, tal y como hemos explicado en los apartados anteriores, así como asegurarnos de que fue compilado con soporte OpenSSL.

OpenLDAP tiene dos formas de comunicación segura, SSL y TLS. SSL opera en otro puerto, el 636 y, desde el principio, realiza las comunicaciones encriptadas en modo seguro. TLS, por contra, es una solución más moderna que utiliza el puerto estándar 389 para iniciar las comunicaciones y que, después, cambia a modo seguro a través del citado puerto 636.

Es conveniente tener los dos tipos activados, ya que no todas las aplicaciones cliente soportan este método, e incluso hay aplicaciones que no soportan las transmisiones seguras. Puedes encontrar más información en la web de OpenSSL en <http://www.openssl.org/>⁽⁹⁾.

Para las transmisiones seguras hace falta un certificado en formato PEM para la llave SSL. Normalmente, se necesitaría que una organización dedicada a generar certificados nos proporcionase una, pero como, normalmente, el OpenLDAP lo vamos a usar en la red local, no a través de internet, ni para hacer comercio electrónico, basta con que la generemos nosotros mismos. Para generar el certificado, debemos irnos al directorio /usr/share/ssl/certs y allí ejecutar el comando:

```
# make slapd.pem
```

Esto nos creará un certificado, con lo que nos irá preguntando: Nombre del equipo, e-mail del administrador, ORganización, etc. Esto es, simplemente, la ejecución de un archivo makefile que lo que hace es llamar a varias utilidades para generalo. Si lo queremos hacer a mano podemos hacer lo siguiente:

```
# /usr/bin/openssl req -newkey rsa:1024 -keyout temporal1 -nodes -x509 -days 365 -out temporal2
# cat temporal1 > slapd.pem
# echo "" >> slapd.pem
# cat temporal2 >> slapd.pem
# rm -f temporal1 temporal2
```

Este comando crea una llave que se firma por sí misma. Esto no es lo normal en los certificados seguros, pero, como hemos dicho, no pasa nada, ya que somos nosotros los que vamos a utilizar estos en la red local privada. (Aunque siempre se puede confiar en un *Certification Authority*, previo pago, claro).

Esta llave tiene que ser de sólo lectura para el usuario que ejecuta slapd. Si cualquier otro usuario tiene acceso a esta llave, la seguridad puede quedar comprometida.

Para configurar el servidor OpenLDAP para que utilice la llave correspondiente, tenemos que editar el fichero /usr/local/etc/openldap/slapd.conf y añadir estas tres líneas:

```
TLSCipherSuite HIGH:MEDIUM:+SSLv2
TLSCertificateFile /usr/share/ssl/certs/slapd.pem
TLSCertificateKeyFile /usr/share/ssl/certs/slapd.pem
```

Llegados a este punto, ya podemos reiniciar el servidor OpenLDAP para que utilice el puerto seguro. Para ello, llamamos a slapd con el parámetro -h de esta forma:



```
# /usr/local/libexec/slapd -h "ldap:/// ldaps://"
```

La opción -h le indica a OpenLDAP que tipo de transmisión queremos hacer, normal, segura o ambas. Teóricamente, cuando comprobemos que funciona en modo seguro, podemos cerrar el puerto inseguro, ejecutando slapd con -h "ldaps:///" solamente.

para comprobar que tenemos el servidor escuchando en el puerto podemos hacer varias cosas:

```
$ netstat -a |grep LISTEN
```

para ver los puertos de escucha y podemos ejecutar esta orden de openssl para ver el certificado que tengamos en el puerto 636 (openLDAP seguro):

```
$ openssl s_client -connect localhost:636 -showcerts
```

con lo que nos tiene que salir por pantalla los datos relativos al certificado que hemos creado.

Con esto ya tenemos constancia de que el servidor LDAP está funcionando en modo seguro. Ahora sólo queda configurar los clientes linux para que la autenticación la hagan a través del directorio LDAP en vez de en modo local con /etc/passwd, utilizando tanto el modo seguro como el no seguro. Pero esto lo veremos en un próximo artículo. Hasta entonces, aquí tenéis algunos cuantos enlaces dónde consultar:

- Web de OpenLDAP: <http://www.openldap.org>⁽¹⁰⁾
- Diversos enlaces y software sobre LDAP: <http://www.padl.com/>⁽¹¹⁾
- OpenSSL : <http://www.openssl.org/>⁽⁹⁾
- RFCs: <http://www.rfc-editor.org/>⁽¹²⁾
- Un interfaz web con el que podreis chequear vuestro directorio LDAP a través de internet: <http://www.web2ldap.de/>⁽¹³⁾.

Lista de enlaces de este artículo:

1. <http://www.eii.us.es>
2. <http://www.rembo.com>
3. <ftp://ftp.isi.edu/in-notes/rfc2251.txt>
4. <ftp://ftp.isi.edu/in-notes/rfc2253.txt>
5. <http://www.openldap.org/software/download/>
6. <ftp://ftp.isi.edu/in-notes/rfc2849.txt>
7. <http://www.padl.com>
8. <http://www.padl.com/OSS/MigrationTools.html>
9. <http://www.openssl.org/>
10. <http://www.openldap.org>
11. <http://www.padl.com/>
12. <http://www.rfc-editor.org/>
13. <http://www.web2ldap.de/>

E-mail del autor: jesus_ARROBA_roncero.org

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1343>