Autenticación LDAP en GNU/Linux

Alberto Molina Coballes <alberto.molina@hispalinux.es>
IES Gonzalo Nazareno. Dos Hermanas (Sevilla)

12 de enero de 2008

Resumen

En este documento se describen los pasos necesarios para configurar un equipo GNU/Linux para gestionar de forma centralizada las cuentas de usuarios de una red mediante un directorio LDAP. Todo el desarrollo se realiza utilizando OpenLDAP 2.3.30 sobre la rama estable de la distribución Debian GNU/Linux 4.0 (etch).

Índice

1. Introducción

La forma clásica de autenticar un usuario en un sistema GNU/Linux —en UNIX en general— es mediante la información existente en los ficheros `passwd`, `shadow` y `group`. Para el lector interesado, en el apéndice A se da una descripción breve de los pasos que se siguen para autenticar un usuario en un sistema UNIX mediante estos ficheros.

El método anterior funciona muy bien para lo que ha sido pensado, sin embargo no es útil para tener un sistema centralizado de autenticación. Para estos casos normalmente se utiliza un doble sistema de autenticación: mediante ficheros para los usuarios locales del sistema —como el usuario `root`, los usuarios para los demonios, etc.— y mediante un segundo método para los usuarios normales que pueden autenticarse en cualquier equipo de la red —lo que podríamos denominar "usuarios de red"—. En este documento nos centramos en explicar la forma de configurar un sistema para poder utilizar estos "usuarios de red", guardando la información en un directorio.

En entornos UNIX durante bastante tiempo se ha utilizado NIS para la autenticación centralizada de usuarios en una red local, ya que se conoce muy bien y se configura de forma sencilla; pero es una opción cada vez más en desuso por diferentes problemas que presenta, en particular porque no funciona sobre TCP/IP —lo que limita su extensión a una red local— y además no envía la información entre el cliente y el servidor de manera cifrada —lo que puede provocar problemas de seguridad—.

En este manual presentamos la opción preferida actualmente para implementar un sistema centralizado de autenticación, que es almacenar dicha información en un directorio LDAP. Este método resulta más eficaz que NIS ya que:

- Funciona sobre TCP/IP.
- Es posible establecer la comunicación entre el cliente y el servidor de forma cifrada.
- Permite guardar muchos otros datos de los usuarios —como direcciones de correo, teléfonos, etc.— e incluso de otros objetos como impresoras u ordenadores.

Como es lógico tiene alguna contrapartida y esta es que es algo más complicado de implantar que NIS.

En las siguientes secciones presentamos la configuración de un servidor OpenLDAP, típicamente se haría sobre un equipo de una red, y la configuración de un cliente, que sería cualquiera de los otros equipos que quieran utilizar estos "usuarios de red". En este documento no se van a explicar las características que tiene un directorio, ni el formato de los ficheros que se utilizan para incluir las entradas (LDIF), para una lectura complementaria sobre estos aspectos recomendamos la lectura de *Introducción al Servicio de Directorio* de Rafael Calzada Pradas [1].

2. Instalación del servidor OpenLDAP

El equipo en el que va a realizarse la instalación tiene definido correctamente su FQDN y es `ldap.gonzalonazareno.org`. Esto no es imprescindible para realizar la configuración correcta del servidor LDAP, pero sí recomendable. Para combrobar si el FQDN de un equipo está bien definido hay que ejecutar:

```
hostname --fqdn
```

En primer lugar debemos instalar el paquete `slapd` y todas sus dependencias:

```
aptitude install slapd
```

A continuación —dependiendo de la configuración del paquete `debconf`— nos pedirá lo siguiente:

- Contraseña del Administrador del directorio

y creará un directorio con dos entradas, en nuestro caso:

```
dn: dc=ldap,dc=gonzalonazareno,dc=org
objectClass: top
objectClass: dcObject
objectClass: organization
o: IES Gonzalo Nazareno
dc: ldap
structuralObjectClass: organization
entryUUID: 8376c53e-4815-102c-97c9-d7aec873b177
creatorsName:
modifiersName:
createTimestamp: 20071226154648Z
modifyTimestamp: 20071226154648Z
entryCSN: 20071226154648Z#000000#00#000000

dn: cn=admin,dc=ldap,dc=gonzalonazareno,dc=org
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e2NyeXB0fUNpTkxkYi9IZ0l6WU0=
structuralObjectClass: organizationalRole
entryUUID: 83773ad2-4815-102c-97ca-d7aec873b177
creatorsName:
modifiersName:
createTimestamp: 20071226154648Z
modifyTimestamp: 20071226154648Z
entryCSN: 20071226154648Z#000001#00#000000
```

En la salida anterior podemos ver dos entradas en formato LDIF, identificadas cada una con un *Distinguished Name* (dn) que es único para cada entrada y una serie de atributos. La salida anterior la hemos obtenido con la instrucción `slapcat` que está incluída en el paquete `slapd` y que nos muestra los objetos del directorio desde el propio equipo sin necesidad de establecer una conexión LDAP propiamente.

En caso de que la base del directorio no esté correctamente definida —que será lo más habitual—, podríamos configurar de nuevo el directorio con:

```
dpkg-reconfigure slapd
```

En este caso nos preguntará más detalles de la configuración, en concreto:

- Nombre de dominio DNS: ldap.gonzalonazareno.org
- Nombre de la Organización: IES Gonzalo Nazareno
- Contraseña del administrador
- Motor de base de datos a utilizar: BDB
- ¿Permitir el protocolo LDAPv2?: No (salvo que sea necesario)

En caso de que esto no fuese suficiente —como ocurre en algunas versiones de Ubuntu— podríamos ir al directorio `/var/lib/ldap` y borrar todo su contenido y a continuación crear los dos objetos iniciales de la base a partir de un fichero LDIF.

Ya por último, recordar que todos los valores que hemos introducido y otros para los que se asumen los valores, se guardan en el fichero `/etc/ldap/slapd.conf`

2.1. Estructura básica del directorio

La estructura del árbol del directorio variará en función de la información que queramos almacenar y de la complejidad de ésta; en nuestro caso vamos utilizar el directorio como sistema de autenticación en una red local y por tanto, queremos almacenar información de los usuarios y los grupos a los que pertenecen. En estos casos habitualmente se utiliza una estructura del estilo a la que aparece en la figura 1.

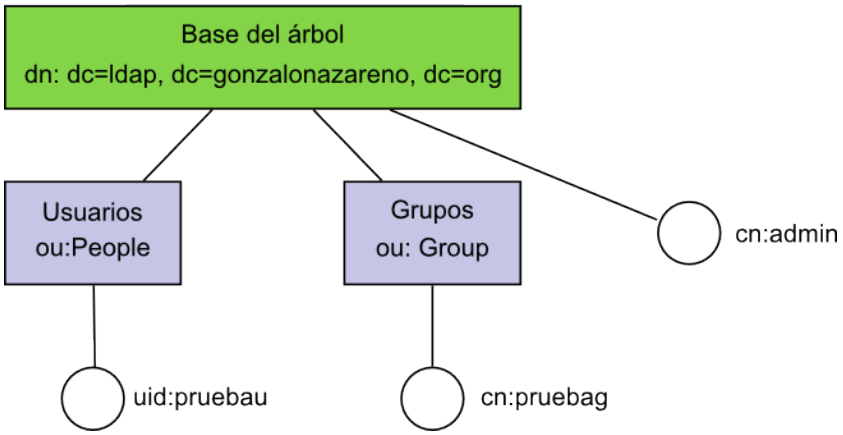

Figura 1: Esquema básico del árbol LDAP.

Como puede verse en la figura, se han creado dos unidades organizativas `(ou)` llamadas `People` y `Group` donde obviamente se incluirán las entradas de los usuarios y los grupos respectivamente, además de un objeto en cada unidad organizativa —el usuario `pruebau` y el grupo `pruebag`—. En el caso de que quisiéramos almacenar información de otros objetos como impresoras, ordenadores, etc. deberíamos añadir nuevas unidades organizativas.

Para añadir las entradas anteriores, debemos crear un fichero en formato LDIF —lo denominaremos `base.ldif`— con el siguiente contenido[1]:

[1]Para crear el contenido del atributo userPassword hemos utilizado `slappasswd -h {MD5}`

```
dn: ou=People,dc=ldap,dc=gonzalonazareno,dc=org
ou: People
objectClass: top
objectClass: organizationalUnit

dn: ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
ou: Group
objectClass: top
objectClass: organizationalUnit

dn: cn=pruebag,ou=Group,dc=ldap,dc=gonzalonazareno,dc=org
objectClass: posixGroup
objectClass: top
cn: pruebag
gidNumber: 2000

dn: uid=pruebau,ou=People,dc=ldap,dc=gonzalonazareno,dc=org
uid: pruebau
cn: Usuario de prueba
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: {MD5}qPXxZ/RPSWTmyZje6CcRDA==
loginShell: /bin/bash
uidNumber: 2000
gidNumber: 2000
homeDirectory: /home/pruebau
gecos: Usuario de prueba
host: *
```

Puesto que todavía no hemos visto las herramientas del cliente LDAP, lo haremos con `slapadd`, para lo que debemos parar antes el demonio slapd:

```
/etc/init.d/slapd stop
```

y añadir las dos nuevas entradas con:

```
slapadd -l base.ldif
```

si no se ha producido ningún error, podremos ver las nuevas entradas utilizando `slapcat`.

Puesto que no existe el directorio home del usuario `pruebau`, tendremos que crearlo y darle el propietario adecuado:

```
mkdir /home/pruebau
cp /etc/skel/.* /home/pruebau
chown -R 2000:2000 /home/pruebau
```

Si hacemos un listado del directorio home del usuario:

```
ls -al /home/pruebau/
total 20
drwxr-xr-x 2 2000 2000 4096 2007-12-28 09:29 .
drwxr-xr-x 9 root root 4096 2007-12-28 09:28 ..
-rw-r--r-- 1 2000 2000  220 2007-12-28 09:29 .bash_logout
-rw-r--r-- 1 2000 2000  414 2007-12-28 09:29 .bash_profile
-rw-r--r-- 1 2000 2000 2227 2007-12-28 09:29 .bashrc
```

podemos ver que el sistema no es capaz de asociar los UID/GID a ningún usuario existente puesto que todavía no está configurado para hacer la búsqueda en el directorio.

2.2. Definición de entradas destacadas

Como hemos mencionado anteriormente cada entrada del directorio se define por un `dn` único, que la identifica dentro del árbol y define la ruta que hay que seguir hasta ella. Hay algunas entradas que se utilizan frecuentemente y por ello las definimos a continuación:

`base dn:` *Distinguished Name* de la raiz del directorio.

`root dn:` *Distinguished Name* del administrador del directorio LDAP.

`bind dn:` *Distinguished Name* del usuario que establece la conexión, si el usuario que se conecta es el administrador, entonces coincide con el `root dn`.

3. Configuración del cliente LDAP

Esta configuración deberá realizarse en cada equipo de la red que vaya a utilizar los usuarios del servidor LDAP. Aquí presentamos la configuración necesaria en el propio servidor, para realizarlo en otro equipo habría que cambiar la dirección IP 127.0.0.1 por la que corresponda al servidor.

En primer lugar configuramos el fichero `/etc/ldap/ldap.conf`, en caso de que no exista habrá que instalar el paquete `libldap2`. Lo editamos y configuramos de la siguiente manera:

```
BASE    dc=ldap,dc=gonzalonazareno,dc=org
URI     ldap://127.0.0.1
```

Debemos configurar el sistema para que pueda obtener del directorio LDAP los siguientes servicios:

- Autenticación: Validando la correspondencia entre un nombre de usuario y una contraseña suministrados por un programa `login`, `ssh`, `gdm`, etc. tal como haría el sistema con un usuario local a través del fichero `/etc/shadow`.
- Información de usuario: Asignando a un usuario que ha ingresado en el sistema su directorio home, UID, shell, etc. tal como haría el sistema con un usuario local a través del fichero `/etc/passwd`.

- *Name service switch*: Estableciendo —cada vez que sea requerido por el sistema— la relación entre el UID/GID de un usuario y su correspondiente nombre, como por ejemplo al crear un fichero, hacer un listado, etc. tal como haría el sistema con un usuario local a través de los ficheros `/etc/passwd` y `/etc/group`.
- Asignación de grupos: Asignando los grupos a los que pertenece cada usuario tal como haría el sistema con un usuario local a través del fichero `/etc/group`.

3.1. Name Service Switch (nss)

En primer lugar vamos a configurar el sistema para que sea capaz de establecer la correspondencia entre los UID/GID y los nombres de los usuarios y los grupos que estén en el directorio LDAP. Esto se hace con los siguientes dos pasos.

3.1.1. Modificación de /etc/nsswitch.conf

Editamos este fichero y modificamos las líneas correspondientes a passwd, group y shadow, que originalmente son:

```
passwd: compat
group: compat
shadow: compat
```

que significa que el sistema va a buscar correspondencia entre UID/GID y nombres en servidores NIS y en los ficheros `passwd`, `group` y `shadow`; las modificamos de la siguiente manera:

```
passwd: compat ldap
group: compat ldap
shadow: compat ldap
```

que indica al sistema que en caso de que no encuentrar el correspondiente usuario (grupo) de un determinado UID (GID), lo busque en el directorio LDAP.

3.1.2. libnss-ldap

Para que el sistema sea capaz de consultar a un directorio LDAP, debemos instalar el paquete:

```
aptitude install libnss-ldap
```

y configurar los siguientes puntos:

- Identificador del servidor LDAP: ldap://127.0.0.1/
- Nombre distinguido (dn): dc=ldap,dc=gonzalonazareno,dc=org
- Versión de LDAP: 3
- Cuenta del administrador: cn=admin,dc=ldap,dc=gonzalonazareno,dc=org

- Contraseña del administrador

Para comprobar que todo está configurado correctamente, repetimos el listado del directorio home del usuario de prueba:

```
ls -al /home/pruebau/
total 20
drwxr-xr-x 2 pruebau pruebag 4096 2007-12-28 09:29 .
drwxr-xr-x 9 root    root    4096 2007-12-28 09:28 ..
-rw-r--r-- 1 pruebau pruebag  220 2007-12-28 09:29 .bash_logout
-rw-r--r-- 1 pruebau pruebag  414 2007-12-28 09:29 .bash_profile
-rw-r--r-- 1 pruebau pruebag 2227 2007-12-28 09:29 .bashrc
```

donde comprobamos que se ha cambiado el UID/GID por el correspondiente nombre de usuario y grupo que se ha obtenido consultando el directorio LDAP.

Otra forma de comprobación, quizás más elegante, es mediante la utilización de `getent`:

```
getent passwd pruebau
getent group pruebag
```

Ya por último, recomendar la instalación del paquete `nscd`, que no necesita configuración, y que es simplemente un demonio que cachea las correspondencias entre los UID/GID y los nombres, con el fin de evitar consultas repetidas al directorio y agilizar así la respuesta.

3.2. Pluggable Authentication Module (PAM)

PAM es el sistema modular que se encarga de las tareas de autenticación en el sistema, cada aplicación que necesite comprobar la autenticación de usuarios tendrá que hacer uso de él y habitualmente tendrá un fichero de configuración en el directorio `/etc/pam.d`; podemos ver el contenido de este directorio en un sistema muy simple:

```
chfn  common-account  common-password  cron   other   ssh  sudo
chsh  common-auth     common-session   login  passwd  su
```

En los cuatro ficheros `common-*` se incluyen las directivas comunes para todas las aplicaciones, mientras que en el resto de ficheros se incluyen las directivas que son aplicables sólo a ese programa.

Recomendación: Puesto que vamos a tocar los ficheros de autenticación del sistema es posible dejar el sistema inaccesible, por lo que es recomendable guardar una copia de todo el directorio `/etc/pam.d`, por ejemplo haciendo:

```
cp -r /etc/pam.d /etc/pam.d.old
```

3.2.1. Modificación de los ficheros common-*

Vamos a permitir que todas las aplicaciones sean capaces de autenticar contra el directorio LDAP, por lo que haremos modificaciones sólo en los anteriormente mencionados ficheros `common-*` del directorio `/etc/pam.d`. Una configuración muy simple sería:

common-password

Contenido inicial:

```
password required pam_unix.so nullok obscure min=4 max=8 md5
```

Contenido final:

```
password sufficient pam_ldap.so md5
password required pam_unix.so nullok obscure min=4 max=8 md5 try_first_pass
```

common-auth

Contenido inicial:

```
auth required pam_unix.so nullok_secure
```

Contenido final:

```
auth sufficient pam_ldap.so
auth required pam_unix.so nullok_secure try_first_pass
```

common-account

Contenido inicial:

```
account required pam_unix.so
```

Contenido final:

```
account sufficient pam_ldap.so
account required pam_unix.so try_first_pass
```

common-session

Contenido inicial:

```
session required pam_unix.so
```

Contenido final:

```
session sufficient pam_ldap.so
session required pam_unix.so
```

En todos los casos existe una línea para la biblioteca `pam_unix.so` que inicialmente utiliza el tipo de control `required` que implica que debe dar un resultado exitoso para que se pueda acceder al servicio. En la configuración final se ha modificado el tipo de control para esta biblioteca a `sufficient`, para que consulte en primer los usuarios locales y después los del directorio ldap, poniendo una nueva línea ahora con el tipo de control `required` para la biblioteca `lib_pam.so`.

3.2.2. libpam-ldap

Para que el sistema sea capaz de consultar a un directorio LDAP, debemos instalar el paquete:

```
aptitude install libpam-ldap
```

y configurar los siguientes puntos:

- *Make local root Database admin*: Sí
- *Does the LDAP database require login?*: No
- Cuenta del administrador: cn=admin,dc=ldap,dc=gonzalonazareno,dc=org
- Contraseña del administrador

Debería pedir algunas cosas más, por lo que reconfiguramos el paquete mediante:

```
dpkg-reconfigure libpam-ldap
```

y contestamos a las preguntas anteriores y a las siguientes:

- Identificador del servidor LDAP: ldap://127.0.0.1/
- Nombre distinguido (dn): dc=ldap,dc=gonzalonazareno,dc=org
- Versión de LDAP: 3
- *Local crypt to use when changing passwords*: exop

Ahora debaríamos ser capaces de ingresar en el sistema a través de `login`, `gdm` o `ssh`, o cambiar la contraseña del usuario con `passwd`.

4. Herramientas de gestión de usuarios

Ahora ya tenemos configurado el servidor LDAP y el cliente para la autenticación de los usuarios que se encuentran en el directorio, pero para la utilización normal debemos aprender cómo se añaden nuevos usuarios o se modifican los atributos de los existentes —es necesario recordar que si utilizamos las instrucciones habituales como `adduser` o `useradd` se creará un usuario local y no una entrada en el directorio—.

La forma que hemos utilizado hasta ahora para crear usuarios ha sido escribir un fichero en formato LDIF que incluya todos los atributos de la entrada y añadirla al directorio con `slapadd`. Esto es válido para los objetos iniciales del directorio, pero totalmente inadecuado para la utilización cuando el directorio está activo. Existen varias formas de actualizar las entradas del directorio o añadir nuevas, y en primer lugar veremos la utilización de las herramientas incluídas en el paquete `ldap-utils`.

4.1. ldap-utils

Es muy interesante poder hacer todo tipo de modificaciones de los objetos del directorio directamente desde la línea de comandos, ya que permite hacer modificaciones con todo detalle y en caso de tener que modificar gran cantidad de entradas —como por ejemplo en un alta masiva— puede automatizarse utilizando *scripts*. En los siguientes apartados comentamos algunas de ellas, aunque también se incluyen en el mismo paquete las siguientes: `ldapmodrdn`, `ldappasswd`, `ldapcompare` y `ldapwhoami`.

4.1.1. ldapsearch

Obviamente se utiliza para hacer búsquedas en el directorio. La búsqueda más sencilla es mediante:

```
ldapsearch -x
```

(el parámetro `-x` será necesario mientras no se configure el acceso con SASL al directorio), que realiza una búsqueda genérica de forma anónima y que no tiene acceso a cierta información como son las contraseñas de los usuarios.

Si queremos realizar una búsqueda con un determinado usuario debemos especificarlo con el parámetro `-D` y además incluir `-W` para que solicite la contraseña del usuario:

```
ldapsearch -x -D "uid=pruebau,ou=People,dc=ldap,dc=gonzalonazareno,
dc=org" -W
```

ahora podemos comprobar que sí aparece la contraseña del usuario `pruebau` ya que sí tiene permiso para leerlo (y modificarlo), pero como es lógico no aparece ninguna otra contraseña.

Normalmente no se hacen búsquedas genéricas, sino que se buscan determinados atributos o cadenas de texto y se puede especificar incluso la rama del árbol en la que se quiere buscar con el parámetro `-b`. Por ejemplo:

```
ldapsearch -x -D "uid=pruebau,ou=People,dc=ldap,dc=gonzalonazareno,
dc=org" -W -b "ou=People,dc=ldap,dc=gonzalonazareno,dc=org"
"uidNumber=2000"
```

buscará entradas que tengan el atributo `uidNumber` igual a 2000 en las entradas que pertenecen a la unidad organizativa `People`.

4.1.2. ldapadd

Esa instrucción se utiliza para añadir nuevas entradas al directorio, y deberemos por tanto tener los permisos necesarios. Para añadir una nueva entrada al directorio, debemos escribirla previamente en un fichero en formato LDIF, la diferencia con `slapadd` es que se pueden añadir con el servidor LDAP ejecutándose. Por ejemplo añadiríamos un nuevo usuario creando el fichero `usuario2.ldif` con este contenido:

```
dn: uid=prueba2,ou=People,dc=ldap,dc=gonzalonazareno,dc=org
uid: prueba2
cn: Segundo usuario
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: {MD5}qPXxZ/RPSWTmyZje6CcRDA==
loginShell: /bin/bash
uidNumber: 2001
gidNumber: 2000
homeDirectory: /home/prueba2
gecos: Segundo usuario
host: *
```

y lo añadiríamos con:

```
ldapadd -x -D "cn=admin,dc=ldap,dc=gonzalonazareno,dc=org" -W
-f usuario2.ldif
```

Es necesario recordar que esto no crea el usuario completamente, sólo crea la entrada en el directorio, debemos crear además el directorio home del usuario como ya hicimos con `pruebau`.

4.1.3. ldapdelete

Lógicamente necesitamos una instrucción que nos permita borrar entradas completas del directorio, si por ejemplo quisiéramos borrar la entrada creada anterior del usuario `prueba2` escribiríamos:

```
ldapdelete -x -D "cn=admin,dc=ldap,dc=gonzalonazareno,dc=org" -W
"uid=prueba2,ou=People,dc=ldap,dc=gonzalonazareno,dc=org"
```

4.1.4. ldapmodify

Se utiliza para modificar las entradas existentes de tres maneras diferentes: añadiendo un nuevo atributo, borrando un atributo existente o modificando el valor de un atributo. Por ejemplo para cambiar el `uidNumber` de un usuario podríamos hacer:

```
ldapmodify -x -D "cn=admin,dc=ldap,dc=gonzalonazareno,dc=org" -W
```

especificando a continuación qué entradas queremos modificar y de qué manera:

```
dn: uid=prueba2,ou=People,dc=ldap,dc=gonzalonazareno,dc=org
changetype: modify
replace: uidNumber
uidNumber: 2010
```

Esta información bien la podemos introducir directamente desde la entrada estándar —terminando con un EOF (CTRL-D)— o desde un fichero y pasarlo como parámetro a `ldapmodify` con `-f`.

También se aceptan entradas del tipo:

```
add: mail
mail: usuario2@gonzalonazareno.org
```

para añadir nuevos atributos o del tipo:

```
delete: hosts
```

para borrarlos.

4.2. ldapscripts

Debido a la incomodidad de manejo directo de las herramientas de `ldap-utils` es frecuente la utilización de las mismas a través de algún *script*. En particular, tenemos un paquete en el que se incluyen algunos bastante útiles y que se denomina `ldapscripts`. El listado de *scripts* de este paquete es:

```
ldapaddmachine
ldapsetprimarygroup
_ldapdeletemachine
_ldaprenamemachine
```

```
ldapdeletegroup
ldapaddgroup
ldapadduserto group
_ldaprenamegroup
_lsldap
ldaprenameuser
_ldapinit
ldapdeleteuserfromgroup
ldapadduser
ldapdeleteuser
```

Para que `ldapscripts` funcione correctamente debe conocer varios parámetros: nombre del host, dn de la base del directorio, cn del administrador, etc. Todos estos valores son configurables en el fichero `ldapscripts.conf` del directorio `/etc/ldapscripts`, pero por defecto asume los valores más comunes y en nuestro caso lo único necesario ha sido crear el fichero `/etc/ldap.secret` que contiene la conraseña del administrador y darle lógicamente sólo permisos de lectura y escritura para el usuario `root`[2].

A la hora de crear usuarios, `ldapscripts` nos da como opción crear el directorio home del usuario a través de la directiva:

```
CREATEHOMES="yes"
```

Veremos a continuación la utilización de algunos de ellos de forma sencilla. Por ejemplo si queremos añadir un nuevo usuario al directorio y que pertenezca al grupo `pruebag`, bastará con ejecutar:

```
ldapadduser prueba3 pruebag
```

y obtendremos la siguiente entrada en el directorio:

```
dn: uid=prueba3,ou=People,dc=ldap,dc=gonzalonazareno,dc=org
objectClass: account
objectClass: posixAccount
cn: prueba3
uid: prueba3
uidNumber: 2002
gidNumber: 2000
homeDirectory: /home/prueba3
loginShell: /bin/bash
gecos: prueba3
description: prueba3
userPassword:: e1NTSEF9MHNDMjZOQnk0emxEMGN0T3FCalQ0ek16WGhabGRmbEE=
```

La contraseña se ha creado de forma aleatoria y hay dos opciones, cambiarla desde una sesión de root con:

```
passwd prueba3
```

[2] chmod 600 /etc/ldapscripts/ldapscripts.conf

o bien configurar `ldapscripts` para que nos dé como salida un fichero con el passwd generado.

Para borrar el usuario anterior basta con hacer:

```
ldapdeleteuser prueba3
```

Todo bastante fácil, ¿no?

4.3. Herramientas gráficas de administración LDAP

Existen bastantes herramientas gráficas de administración de un directorio LDAP, en particular algunas de las incluídas en la rama estable de Debian actualmente son: gq, gosa, phpldapadmin, luma, directory-administrator o lat.

Es especialmente cómoda phpldapadmin, ya que permite su utilización desde cualquier equipo a través de un navegador. En la figura 2 podemos ver una captura donde se pueden intuir sus posibilidades.

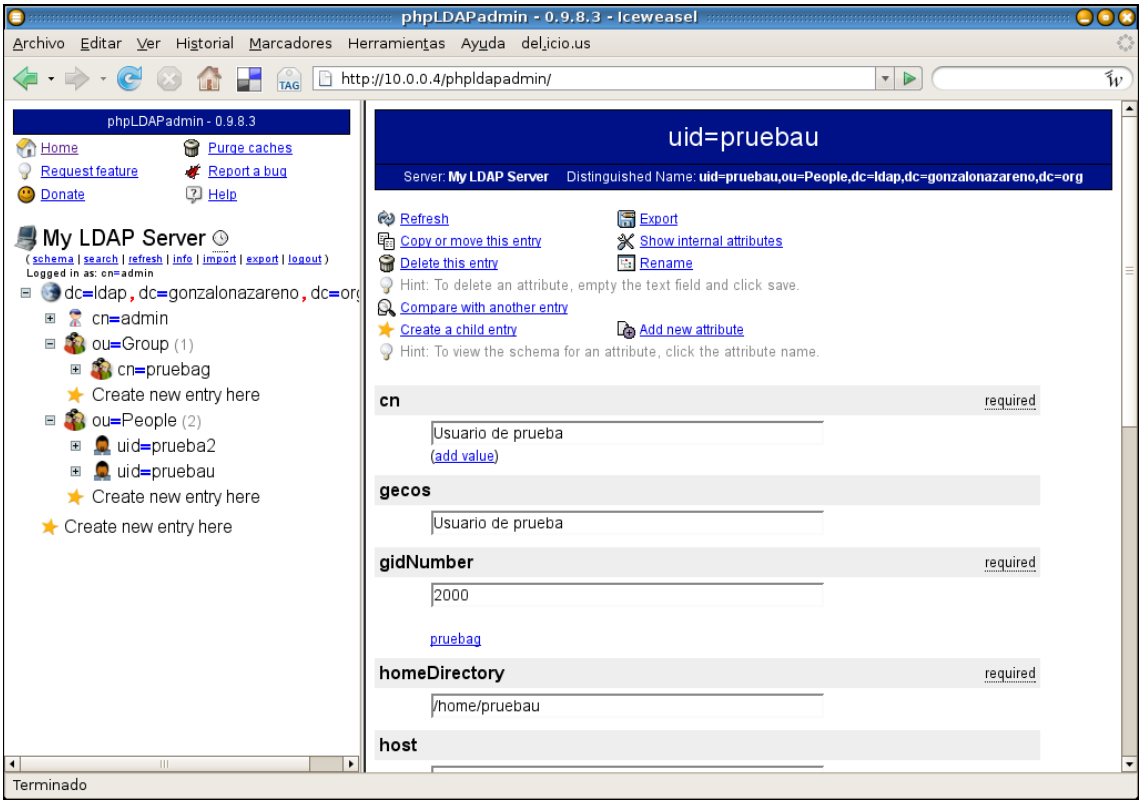

Figura 2: phpLDAPadmin en funcionamiento.

Referencias

[1] Rafael Calzada Pradas. Introducción al Servicio de Directorio
http://www.rediris.es/ldap/doc/ldap-intro.pdf

[2] LDAP Authentication on Debian Sarge HOWTO
http://moduli.net/sysadmin/sarge-ldap-auth-howto.html

[3] Ldap Authentication on Debian
http://www.jukie.net/~bart/ldap/ldap-authentication-on-debian/index.html

[4] CNICE. Redes de Área Local. Aplicaciones y servicios en Linux. Capítulo 14.
http://formacion.cnice.mec.es/materiales/85/cd/REDES_LINUX/indice.htm

[5] Antonio Villalón Huerta. Seguridad en UNIX y redes
http://www.rediris.es/cert/doc/unixsec/

A. Autenticación clásica en UNIX (*shadow passwords*)

Vamos a comentar brevemente los pasos que da el sistema para recabar toda la información necesaria de los ficheros `shadow`, `passwd` y `group`[3] ya que puede resultar útil entenderlo para ver las similitudes y diferencias con la autenticación a través de un directorio.

1. Se recoge el nombre de usuario y la contraseña y se validan con la información existente en el fichero `/etc/shadow`[4]; si la validación es correcta se sigue en el siguiente paso y en caso contrario se repite éste. Una línea de ejemplo de un fichero `shadow` sería:

   ```
   borrame:$1$nXzBxaBL$2eCiIBskzccblg/dTHFDj1:13875:0:99999:7:::
   ```

   donde sólo los dos primeros campos son relevantes para nuestra discusión, el primero es el nombre de usuario y el segundo es su contraseña cifrada.

2. Se consulta en `/etc/passwd` los valores de UID, GID, directorio home y shell de ese usuario, una línea tipo sería la siguiente:

   ```
   borrame:x:1012:1012:,,,:/home/borrame:/bin/bash
   ```

   que nos indica que el usuario `borrame` tiene UID y GID 1012, su directorio home es `/home/borrame` y la shell que utilizará al ingresar en el sistema será `bash`.

3. Cada vez que el usuario cree o modifique un fichero —salvo que haya cambiado de grupo principal, como por ejemplo con `newgrp`— se le asignará como propietario y grupo propietario los valores de UID y GID anteriores. Por ejemplo:

   ```
   touch fichero_nuevo
   ls -nl
   -rw-r--r-- 1 1012 1012 0 2007-12-28 21:00 fichero_nuevo
   ls -l
   -rw-r--r-- 1 borrame borrame 0 2007-12-28 21:00 fichero_nuevo
   ```

4. Si el usuario pertenece a algún otro grupo del sistema deberá reflejarse en el fichero `etc/group`, donde además se establece la relación entre los GID y los nombres de los grupos.

[3] Muy recomendable la lectura de *Seguridad en UNIX y redes* de Antonio Villalón[5], en particular la sección "Autenticación de usuarios en UNIX"

[4] Salvo que el sistema no tenga este fichero y guarde la contraseña en el fichero `/etc/passwd`, algo poco frecuente hoy en día