Autenticació d'usuaris via LDAP

NSS :

NSS (“Name Service Switch”) es un conjunto de librerías que permiten a las aplicaciones que hagan uso de ellas acceder de forma coherente y común a varios posibles orígenes de datos que tienen una característica común: contener equivalencias de nombres de ítems diversos (como máquinas, redes, nombres de protocolos, usuarios, grupos, etc) con sus valores reales. En otras palabras: las “aplicaciones NSS” son capaces de entender el fichero /etc/nsswitch.conf, el cual indica qué orígenes de datos se pueden utilizar para resolver los distintos tipos de nombres que dichas aplicaciones necesitan.

Por ejemplo, para obtener la información sobre los nombres de usuarios reconocidos por el sistema, las “aplicaciones NSS” consultarán /etc/nswitch.conf para saber si la han de obtener de los ficheros locales (/etc/passwd y /etc/shadow) o bien de alguna base de datos remota (como pudiera ser un servidor LDAP), o bien complementar ambos orígenes para que si falla el predeterminado se utilice el otro. Otro ejemplo: para conocer la correspondencia entre nombres de máquinas y su ip, las “aplicaciones NSS” consultarán /etc/nswitch.conf para saber si primero han de consultar en un servidor DNS o bien pueden obtener la información del fichero /etc/hosts, o bien pueden complementar ambos orígenes en un orden determinado, etc.

El objetivo de NSS es que las aplicaciones que hagan uso de este sistema puedan obtener este tipo de información (nombres de usuarios y sus contraseñas, nombres de máquinas, de redes, de protocolos, etc) preguntando directamente a NSS sin tener que conocer el lugar exacto donde está almacenada, (ya que para obtenerla ya se encarga NSS).

Un ejemplo de archivo /etc/nswitch.conf muy simple sería el siguiente:

```
passwd: files
group: files
shadow: files
hosts: files dns
networks: files
protocols: files
services: files
ethers: files
```

...donde se indica que la base de datos de usuarios que se va a utilizar es el fichero local /etc/passwd (por el valor “files”), la de grupos es el fichero local /etc/group, la de contraseñas es el fichero local /etc/shadow, la de nombres de máquinas será en primera instancia el fichero local /etc/hosts (ver *man hosts* para más información sobre este fichero, o también http://en.wikipedia.org/wiki/Hosts_%28file%29), y, si no se encuentra el nombre allí, entonces se procederá (ya que tras “files” está el valor “dns”) a realizar una búsqueda DNS en los servidores configurados en /etc/resolv.conf; los nombres de redes se buscarán en el fichero local /etc/networks (ver *man networks* para más información sobre este fichero), los de los protocolos en el fichero local /etc/protocols (ver *man protocols* para más información sobre este fichero), los de los servicios en el fichero local /etc/services (ver *man services* para más información sobre este fichero) y la correspondencia estática entre direcciones MAC e Ips en el fichero local /etc/ethers (ver *man ethers* para más información sobre este fichero).

PAM :

Existeixen moltes maneres d'autenticar usuaris: consultant els logins/passwords locals dins la parella d'arxius /etc/passwd i /etc/shadow, o bé consultant-los en un servidor LDAP, o en una base de dades relacional convencional, o bé a través de tarjetes hardware, o via reconeixement d'empremtes dactilars, etc.

Això pels desenvolupadors és un problema, perquè per a què els seus programes (com ara *login*,

gdm, sudo, ftp, nfs, ssh,...) suportin tots aquests diferents mètodes d'autenticació, han de programar explícitament cada mètode per separat (a més de què si aparegués un altre mètode nou, s'hauria de recompilar el programa per a què el soportés -en el cas que fos possible-, o bé reescriure de nou el programa sencer.

PAM és un conjunt de llibreries que fan d'intermediàries entre els mètodes d'autenticació i els programes que els fan servir, permetent la possibilitat de desenvolupar programes de forma independent a l'esquema d'autenticació a utilitzar. La idea és que el programa desenvolupat per utilitzar PAM utilitzi un determinat "mòdul d'autenticació", i que aquest "mòdul" s'encarregui de la "feina bruta". El mòdul concret a utilitzar pel programa en un determinat moment l'escollirà l'administrador del sistema, el qual podrà canviar de mòdul (de /etc/shadow a LDAP, per exemple), si així ho desitgés, sense que el programa corresponent notés la diferència (en realitat, la configuració del sistema normalment fa utilitzar als programes no només un mòdul, sinò un conjunt "en cascada", per si un falla).

Resumint: PAM facilita la vida als desenvolupadors perquè en comptes d'escriure una complexa capa d'autenticació pels seus programes simplement han d'escriure un "hook" (un "ganxo") a PAM, i també facilita la vida als administradors perquè poden configurar l'autenticació dels programes que la necessiten d'una forma centralitzada i comú sense necessitat de recordar cada model separat per cada programa.

LDAP :

LDAP ("Lightweight Directory Access Protocol") es un protocolo de nivel de aplicación que permite acceder a un "servidor de directorio". Por "directorio" se entiende un conjunto de datos organizados de una manera lógica y jerárquica en forma de elementos de información llamados "entradas", los cuales poseen diversos atributos. Cada entrada representa un objeto que puede ser abstracto o real (una persona, un mueble, una función en la estructura de una empresa, etc). La utilidad de un servidor de directorio radica en ofrecer dichos objetos a la red de una forma centralizada (y, opcionalmente, transparentemente distribuida). Se puede entender que un servidor de directorio pueda ser equivalente a un servidor de bases de datos, pero tanto el sistema de almacenamiento como la forma de consultar/manipular la información contenida en él es distinta.

Aunque no tiene por qué ser así siempre, el tipo de información que suele encontrarse en la mayoría de ocasiones en un servidor LDAP es típicamente aquella relacionada con la autenticación centralizada de usuarios (nombre, contraseña, uid, grupo, permisos, etc), o con la autenticación centralizada de máquinas (nombre, dirección MAC, dirección IP, etc). También puede contener información complementaria de usuarios (correo, teléfonos, dirección, etc) o configuraciones centralizadas de aplicaciones y certificados, etc.

Para definir los atributos que tendrán las entradas almacenadas en un servidor LDAP podemos hacer uso de las llamadas "Reglas de Esquema". Éstas son plantillas que especifican qué atributos formarán la entrada de forma obligatoria y cuáles de forma optativa (a modo de "esqueleto" de la entrada). La "Regla de Esquema" concreta utilizada por una entrada determinada se indica en su atributo especial objectClass; esta Regla se puede escoger de entre un conjunto de Reglas estandarizadas que ofrece todos los servidores LDAP. También podríamos crear nuestras propias Reglas de Esquema (o modificar alguna existente), pero no suele ser necesario porque las Reglas predefinidas cubren la mayoría de casos prácticos.

Entre los atributos que suelen emplearse para definir una entrada, habitualmente, encontraremos los siguientes, (aunque puede haber muchos más y diferentes, dependiendo de la Regla utilizada):

uid (user id): Identificador de la entrada. ¡No confundir con el uid de usuario!
objectClass: Indica el tipo de entrada (la Regla que la define)
ou (organizational unit): Contenedor estructural (a modo de "carpeta") dentro del cual está categorizada esta entrada particular en el directorio
cn (common name): Nombre de la persona representada en la entrada
sn (surname): Apellido de la persona.
mail: dirección de correo electrónico de la persona.
o (organization): Departamento (de la empresa, organización...) al que pertenece la persona.

Tal como se puede ver, en el caso anterior se ha utilizado una Regla (muy típica, por otro lado) que hace referencia a objetos representando empleados de una empresa. En el caso de carecer de atributo “uid”, el atributo que suele hacer de identificador de la entrada suele ser “cn”.

Las entradas se organizan en una estructura jerárquica en forma de árbol invertido. No es obligatorio, pero es costumbre que la parte superior de esta estructura refleje la jerarquía de los dominios DNS de la organización, de manera que las entradas que representan a la compañía (como “pepsi.com”, “unicef.org” o “yahoo.es”) aparecen en el árbol por encima de otras entradas que representan unidades organizativas internas. Las primeras suelen identificarse por la presencia del atributo “dc” ("domain component"), y para cada subdominio hay una (por ejemplo, dc=”pepsi” y dc=”com” para “pepsi.com”, dc=”unicef” y dc=”org” para “unicef.org”, etc). Dentro de las últimas entradas de la jerarquía será donde se encuentre la información relativa a usuarios, máquinas, documentos o cualquier otra cosa que queramos.

Sea del tipo que sea (“domain component” o no) y represente lo que represente, toda entrada posee un único “Nombre Distinguido” -“Distinguised Name” (DN)-, que sirve para identificarla de manera unívoca. El DN se construye a partir del identificador de la entrada en sí misma (lo que se llama “Nombre Relativo Distinguido” -”Relative Distinguished Name” (RDN)-, el cual suele ser el valor de su atributo “uid” o bien “cn” -o “ou” en el caso de las unidades organizativas- concatenado con los identificadores de las entradas de sus antecesores separados por comas. Por ejemplo: si el DN de una entrada es “uid=pperez,ou=empleados,dc=nike,dc=es”, nos estaremos refiriendo a una entrada cuyo RDN es “uid=pperez” y que contiene información sobre el empleado Pperez perteneciente a la sección española de Nike. Para conocer más información sobre ese empleado, deberíamos observar el resto de atributos de esa entrada (objetClass, cn, givenname, sn, o,mail ...).

LDAP tiene definidas las operaciones necesarias para interrogar y actualizar el directorio (es decir, adicionar y borrar una entrada, modificar una entrada existente, cambiar el nombre de una entrada, etc). No obstante, la mayor parte del tiempo LDAP se utiliza para buscar información almacenada en el directorio: las operaciones de búsqueda permiten que en una porción del directorio se encuentren entradas que cumplan con algún criterio especificado en el filtro de búsqueda.

Algunos servidores de directorio no tienen protección y permiten que cualquier persona consulte la información que contienen, pero LDAP provee un mecanismo para que los clientes se autentiquen, (o al menos confirmen su identidad) para garantizar un control de acceso y proteger así la información que el servidor contiene.

Nosotros utilizaremos la infrastructura LDAP para poder loguearnos en un PC mediante un usuario y contraseña guardados en forma de entrada dentro de un servidor de directorio, obteniendo además información adicional sobre dicho usuario para poder asignarle los permisos adecuados.

Instalación y configuración del servidor OpenLDAP:

Nosotros utilizaremos el software OpenLDAP (http://www.openldap.org) sobre Ubuntu por ser uno de los más extendidos y con mayor historia y documentación, pero existen varios servidores LDAP más en el ecosistema Linux como por ejemplo 389 Directory Server o Apache Directory Server, entre otros. También existen soluciones integradas que intentan aglutinar diferentes servicios relacionados para funcionar de forma equivalente al Active Directory de Windows como es FreeIPA (servidor LDAP+DNS+NTP+Kerberos) o el propio Samba4. Por otro lado, clientes LDAP aquí no veremos más que las herramientas ldapadd y família però también existen clientes gráficos como JxExplorer o Apache Directory Studio, entre otros:

1.-a) Crea una máquina virtual llamada "ServidorLDAP" y otra máquina virtual diferente llamada "ClienteLDAP", ambas con un sistema Ubuntu Server instalado. Asegúrate también que ambas máquinas tengan, además de una tarjeta en modo NAT para poder conectarse a Internet, una tarjeta en modo "red interna".
NOTA: En este documento supondremos que las tarjetas en modo "red interna" de ambas máquinas serán identificadas por sus respectivos sistemas Ubuntu con el nombre de "enp0s8". Para que esto sea así, en el panel de configuración de red del VirtualBox de cada máquina se deberá añadir dicha tarjeta en la 2ª pestaña (la 1ª será para la tarjeta NAT, la cual se identificará por ello con el nombre de "enp0s3").

b) Arranca ambas máquinas y tras, comprobar que puedes iniciar sesión correctamente (con usuario "usuari" y contraseña "usuari"), asígnale a sus respectivas tarjetas enp0s8 una ip fija (en este documento supondremos que "ServidorLDAP" tiene la 192.168.0.1 y "ClienteLDAP" la 192.168.0.2. Una vez comprobado, apágalas.
NOTA: Para una mayor comodidad en el uso de las ips internas, se recomienda añadir las siguientes líneas a los archivos /etc/network/interfaces de ambas máquinas (donde "X" será "1" ó "2" según corresponda) y reiniciar:

```
auto enp0s8
iface enp0s8 inet static
address 192.168.0.X
netmask 255.255.255.0
```

c) Arranca la máquina "ServidorLDAP" y dale un nombre que tenga una estructura similar a un nombre DNS (por ejemplo, en este documento usaremos el de "miservidor.midominio.local"). Para ello modifica manualmente el contenido del archivo **/etc/hostname** sustituyendo el valor que haya en ese momento por miservidor.midominio.local ,modifica manualmente el contenido del archivo **/etc/hosts** para que 127.0.0.1 está asociado al "miservidor.midominio.local" en vez de a "localhost" y reinicia la máquina.
NOTA: Que el nombre del servidor sea de tipo DNS es necesario para que las entradas "dc" de nuestro servidor LDAP se correspondan con los subdominios de dicho nombre. Es decir, al generar la base de datos de nuestras entradas en forma de árbol invertido, todas ellas deberán colgar de un DN base que en nuestro caso será "dc=midominio,dc=local". De hecho, la infrastructura básica de una red con servidor LDAP integrado se debería completar con un servidor DNS más pronto que tarde (para, entre otras cosas, no tener que configurar el archivo /etc/hosts de los clientes individualmente), pero para no complicar más las cosas, nos conformaremos con realizar los pasos anteriores y ya está.

d) Instala el software básico: ***sudo aptitude install slapd ldap-utils*** (el primer paquete es el servidor propiamente dicho, el segundo son las herramientas de administración; en Fedora reciben el nombre de "openldap-servers" y "openldap-clients", respectivamente). Durante el proceso de instalación aparecerá un mensaje que solicita la contraseña necesaria para administrar el servidor LDAP: introduce una cualquiera PERO ACUÉRDATE DE ELLA a partir de ahora.
NOTA: El comando aptitude no está instalado por defecto en Ubuntu: puedes instalarlo o bien, si lo prefieres, puedes utilizar el comando *apt-get* en su lugar

e) Instala el software necesario para convertir el servidor LDAP (actualmente generalista) en un servidor específico de autenticación: ***sudo aptitude install ldap-auth-config*** (se instalarán como dependencias, entre otros, los paquetes "libnss-ldap" y "libpam-ldap"). Durante el proceso de instalación aparecerá un asistente que irá solicitando diversa información:

*Lo primero que pide es la dirección IP del servidor IP (en nuestro caso, 192.168.0.1 -¡es importante cambiar el principio para que sea "ldap://"!-)

*Después introduciremos el DN base de nuestro servidor ("dc=midominio,dc=local")

*Tras elegir la versión 3 del protocolo, a continuación indicaremos que sí queremos que las utilidades que utilicen PAM se comporten del mismo modo que si utilizáramos contraseñas locales (esto hará

que las contraseñas se guarden en un archivo independiente que sólo podrá ser leido por el administrador)

*Seguidamente, diremos que no es necesario identificarnos para hacer consultar a la base de datos (es mucho más sencillo realizar consultas anónimas al no tener que incluir ninguna contraseña en los clientes de la red) pero sí indicaremos el nombre de la cuenta LDAP que tendrá privilegios para realizar cambios en el directorio: debemos escribir un DN tal como "cn=admin,dc=midominio,dc=local" (donde el nombre "admin" podría ser cualquier otro) y a continuación introducir la misma contraseña que introdujimos en el apartado *e)*.

*Finalmente (si lo pregunta), deberemos elegir el método "crypt" para la gestión de la encriptación de las contraseñas de los usuarios que guardaremos.

NOTA: Si más adelante observamos algún error o necesitamos efectuar alguna modificación en los datos introducidos, sólo tendremos que ejecutar el siguiente comando: *sudo dpkg-reconfigure ldap-auth-config* o bien editar directamente el archivo */etc/ldap.conf* (que es donde se han guardado todos estos datos) y reiniciar el servidor.

f) Modifica los archivos de configuración de NSS y PAM del servidor para configurar cómo se realizará el proceso de autenticación de los clientes. Concretamente:

*Modifica las siguientes líneas de /etc/nsswitch.conf para que queden así:

passwd: files ldap
group: files ldap
shadow: files ldap

*Ejecuta el comando ***sudo pam-auth-update*** y elige para habilitarlos todos los módulos PAM (especialmente el "Create Home directory when login", que aparece desmarcado por defecto).

g) Configura mediante el comando ***sudo dpkg-reconfigure slapd*** el servidor LDAP propiamente dicho (para poderlo empezar a utilizar). En el asistente que aparece, se nos pedirá confirmar (dos veces) el dominio DNS de nuestro servidor ("midominio.local", sin comillas) ya que este sera será utilizado como referencia para generar el DN base de nuestro directorio. Además nos preguntará la contraseña de administración (la misma que introdujimos en el primer punto), el formato interno de base de datos a usar (elegiremos "HDB"), si deseamos eliminar la base de datos en el caso de desinstalar el servidor (contestaremos que sí) y un par más de preguntas que contestaremos con la opción por defecto.

Administración básica del directorio:

Ya tenemos el servidor LDAP funcionando (***sudo service slapd status***) y escuchando en el puerto 389 TCP. Ahora deberíamos generar la estructura de entradas de nuestro directorio y rellenarlas de datos. La forma más básica de añadir información al directorio es utilizar ficheros de texto cuyo contenido está escrito en el formato LDIF (LDAP Data Interchange Format). El formato básico de una entrada es:

```
# comentario
dn: <nombre global único>
<atributo>: <valor>
<atributo>: <valor>
...
```

Entre dos entradas consecutivas debe existir siempre una línea en blanco. Por otro lado, si una línea es demasiado larga, podemos repartir su contenido entre varias, siempre que las líneas de continuación comiencen con una tabulación o un espacio en blanco.

Una vez creados estos ficheros, para añadirlos al directorio (incluso con el servidor en marcha) podemos utilizar el comando *ldapadd* (del paquete "ldap-utils"). La mayoría de veces necesitaremos indicar cuatro parámetros: -f fichero.ldif (para indicar el fichero cuyo contenido se desea agregar), -D "cn=admin,dc=midominio,dc=local" (para indicar la cuenta con la que nos autenticaremos en el servidor LDAP para realizar la modificación del directorio; esta cuenta ha de tener privilegio para ello, y por tanto, ha de ser la cuenta indicada en el punto f) del ejercicio anterior), -W (para que se solicite la contraseña de dicha cuenta interactivamente) y -x (para indicar que esta cuenta se autenticará de forma simple).

NOTA: Todos los comandos del paquete "ldap-utils" disponen además del parámetro *-H ldap://ipServidor* para poder indicar el servidor LDAP contra el cual se van a ejecutar; esto es muy útil para utilizar estos comandos en un ordenador diferente del propio servidor

2.- a) Crea un fichero llamado “base.ldif” con el contenido mostrado a continuación y seguidamente agrégalo al directorio con el comando: ***ldapadd -x -D cn=admin,dc=midominio,dc=local -W -f base.ldif*** . Con esto habrás generado dos entradas de tipo “unidad organizativas” que servirán para contener (a modo de “carpetas”) los usuarios y grupos que generaremos en siguientes apartados.

```
dn: ou=usuarios,dc=midominio,dc=local
objectClass: organizationalUnit
ou: usuarios

dn: ou=grupos,dc=midominio,dc=local
objectClass: organizationalUnit
ou: grupos
```

b) Crea un fichero llamado “grupos.ldif” con el siguiente contenido y seguidamente agrégalo al directorio con un comando similar al del apartado anterior:

```
dn: cn=grupoldap,ou=grupos,dc=midominio,dc=local
objectClass: posixGroup
cn: grupoldap
gidNumber: 10000
```

c) Crea un fichero llamado “usuarios.ldif” con el siguiente contenido y seguidamente agrégalo al directorio con un comando similar al del apartado anterior:
NOTA: Tal como se puede ver, cada objeto “usuario” está formado a partir de la unión de diferentes tipos predefinidos de objeto (posixAccount, shadowAccount, inetOgPerson), donde cada uno aporta un determinado conjunto de atributos: posixAccount incluye la información que encontraríamos en el archivo /etc/passwd clásico, shadowAccount incluye la información que encontraríamos en el archivo /etc/shadow clásico y inetOrgPerson incluye información extra del usuario dentro de la organización (como el correo, código postal, etc).
NOTA: Hay que tener **muy en cuenta** que al añadir nuevos usuarios los valores -marcados en color verde- de los atributos uidNumber y homeDirectory (además de, lógicamente, userPassword, señalado en rojo) deben ser diferentes para cada usuario. Lo mismo ocurre conel atributo gidNumber para los grupos. Además, los valores de uidNumber y gidNumber no deben coincidir con el uid y gid de ningún usuario y grupo local de los clientes.
NOTA: Hay que tener mucho cuidado en NO añadir por despiste ningún espacio en blanco al final de cualquier línea del fichero. Si se hiciera, al ejecutar el comando *ldapadd* nos encontraríamos con errores

```
dn: uid=usu1ldap,ou=usuarios,dc=midominio,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: usu1ldap
sn: Lopez
givenName: Juan
cn: Juan Lopez
displayName: Juan Lopez
uidNumber: 3000
gidNumber: 10000
userPassword: XXX
gecos: Es muy tonto
loginShell: /bin/bash
homeDirectory: /home/jlopez
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
shadowLastChange: 10877
mail: juan.lopez@gmail.com
```

postalCode: 29000
####################LINEA EN BLANCOOOOO####################
dn: uid=usu2ldap,ou=usuarios,dc=midominio,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: usu2ldap
sn: Perez
givenName: Perico
cn: Pedro Perez
displayName: Pedro Perez
uidNumber: 3001
gidNumber: 10000
userPassword: XXX
gecos: Es un crack
loginShell: /bin/bash
homeDirectory: /home/pperez
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
shadowLastChange: 10877
mail: pedrito@yahoo.es
postalCode: 29001

d) Ahora podemos comprobar si el contenido anterior se ha añadido correctamente. Para ello podemos usar el comando *ldapsearch* (del paquete "ldap-utils"), el cual permite hacer una búsqueda en el directorio. Concretamente, ejecuta ***ldapsearch -x -LLL -b "dc=midominio,dc=local" uid=usu1ldap sn givenName*** Con este comando de ejemplo estaremos buscando un usuario con uid=usu1ldap y pediremos que nos muestre el contenido de los atributos sn y givenName (se puede no pedir ningún atributo en concreto: en ese caso se muestran todos). El parámetro -LLL sirve para utilizar ldapsearch en modo "no verboso" y el parámetro -b sirve para indicar el DN base a partir del cual se empezará la búsqueda por las entradas inferiores del árbol.
NOTA: Fijarse que ahora no ha sido necesario utilizar los parámetros -D y -W porque en el apartado *f)* del primer ejercicio indicamos que para realizar consultas no se necesitaba realizar ninguna autenticación.

Otros comandos importantes del paquete "ldap-util" son *ldapdelete* y *ldapmodify*. Un ejemplo del primero (bastante evidente) podría ser: ***ldapdelete -x -D cn=admin,dc=midominio,dc=local -W "uid=usu2ldap,ou=usuarios,dc=midominio,dc=local"*** . El comando *ldapmodify*, por su parte, tiene tres formas de modificar una entrada: añadiendo un nuevo atributo, eliminando un atributo existente o modificando el valor de un atributo. Estas tres modificaciones se realizan mediante el mismo comando, así: ***ldapmodify -x -D cn=admin,dc=midominio,dc=local -W -f fichero.cambios*** donde es el contenido de "fichero.cambios" lo que deberá ser diferente según queramos añadir un nuevo atributo, eliminar uno existente o modificar su valor.

Concretamente, para modificar el valor de un atributo ya existente, el contenido de "fichero.cambios" podría ser el siguiente (ahí se puede ver que se especifica qué entradas se quieren modificar y de qué manera; esta información la podríamos haber introducido directamente desde la entrada estándar si no hubiéramos especificado el parámetro -f en el comando):

dn:uid=usu2ldap,ou=usuarios,dc=midominio,dc=local
replace:uidNumber
uidNumber:3002

Para añadir un atributo nuevo deberíamos escribir, en cambio:

```
dn:uid=usu2ldap,ou=usuarios,dc=midominio,dc=local
add:jpegPhoto
jpegPhoto:file:///tmp/foto.png
```

Y para borrarlo:

```
dn:uid=usu2ldap,ou=usuarios,dc=midominio,dc=local
delete:jpegPhoto
```

3.-a) Modifica el atributo “gecos” del usuario Juan López para que muestre la descripción: “Ronca”. Comprueba mediante *ldapsearch* que el cambio lo has realizado correctamente
b) Añade el atributo “jpegPhoto” del usuario Juan López indicando la ruta (ficticia) de su foto identificativa. Comprueba mediante *ldapsearch* que el cambio lo has realizado correctamente
c) Elimina el atributo “jpegPhoto” anterior. Comprueba mediante *ldapsearch* que el cambio lo has realizado correctamente

Autenticación mediante LDAP en máquinas cliente:

4.-a) Arranca la máquina "ClienteLDAP" y modifica su fichero **/etc/hosts** para que 192.168.0.1 apunte a “miservidor.midominio.local” y reinicia. Comprueba que haciendo "ping" a "miservidor.midominio.local" se obtiene respuesta.

b) Repite los apartado *e)* y *f)* del primer ejercicio, esta vez para instalar y configurar los paquetes necesarios que permitan utilizar NSS y PAM como clientes de autenticación y conectarlos al servidor LDAP ya funcional. Las respuestas a dar en el asistente que aparecen son exactamente las mismas que las que se dieron en su momento. OJO: además has de instalar los paquetes libpam-ldapd y libnss-ldapd !! Reinicia.
NOTA: Es interesante observar en este sentido los cambios producidos automáticamente tras este proceso en los ficheros /etc/pam.d/common-auth, /etc/pam.d/common-account y /etc/pam.d/common-password.

c) Descomenta todas las líneas del archivo /etc/ldap/ldap.conf (correspondiente a la configuración del cliente) y modifica las líneas BASE y URI para que queden así:

```
BASE    dc=midominio,dc=local
URI     ldap://miservidor.midominio.local
```

d) Ejecuta el comando ***getent passwd*** (o ***getent shadow***) para ver si todo ha ido bien. Este comando obtiene las entradas disponibles en los diversos orígenes especificados dentro de /etc/nsswitch.conf para la categoría indicada (“passwd” o “shadow”, respectivamente). La gracia está en que este comando junta la información local (“files”) con la obtenida a través de la red (“ldap”), por lo que deberíamos ver al mismo tiempo los usuarios locales y los usuarios LDAP.

e) El cliente ya está listo para que nos autentiquemos con una cuenta del servidor LDAP. Sin embargo, si ahora nos identificáramos en el cliente con la cuenta usu1ldap, por ejemplo, encontraríamos que no existe su carpeta personal (/home/jlopez) en el equipo cliente. Lógicamente, podríamos crear dicha carpeta a mano, pero habría que repetir el proceso en cada uno de los clientes en los que el usuario vaya a iniciar sesión. Si queremos que la carpeta se cree automáticamente cuando el usuario inicie sesión por primera vez en un equipo, deberemos hacer uso de un módulo PAM llamado “pam_mkhomedir”. Esto lo conseguimos haciendo una pequeña modificación en el archivo **/etc/pam.d/common-session** del cliente; concretamente, tenemos que añadir una nueva línea al principio de ese archivo con este contenido:

```
session required pam_mkhomedir.so skel=/etc/skel umask=0022
```

f) Ya podemos loguearnos con los usuarios LDAP en un terminal virtual de la máquina cliente. Pruébalo.

g) Un inconveniente que aún tenemos es que con la configuración actual los usuarios LDAP no pueden cambiar sus propias contraseñas (prueba de ejecutar el comando *passwd* para comprobarlo). Para solucionarlo, en la máquina cliente deberemos cambiar (siendo "usuari" porque si no no podremos) el archivo **/etc/pam.d/common-password**; concretamente debemos borrar las palabras use_authtok try_first_pass de la línea que usa pam_ldap.so. Vuelve a iniciar sesión con un usuario LDAP y comprueba que ahora sí que puedes utilizar el comando *passwd* para cambiar su contraseña

h) Ya podemos loguearnos también con un display manager. El problema es que tanto Gdm como Lightdm sólo incluyen en la lista de usuarios para seleccionar a aquellos que ya conoce (es decir, que han iniciado sesión al menos alguna vez). En el caso de Gdm esto no es demasiado problema porque él mismo ofrece la posibilidad mediante el botón "¿No estás listado?" de introducir el nombre y contraseña que deseemos y así añadir ese nuevo usuario a la lista en próximos inicios, pero en el caso de LightDM, para obligar a que pregunte un nombre de cuenta debemos editar el archivo **/usr/share/lightdm/lightdm.conf.d/50-unity-greeter.conf** y añadir las líneas allow-guest=false y greeter-hide-users=true y reiniciar.

Perfiles móviles con NFS :

Tal como hemos dejado la configuración de los usuarios LDAP hasta ahora, aunque inician sesión correctamente, todavía hay algo que no queda demasiado elegante: un usuario que vaya itinerando entre varios equipos cliente acabará teniendo una carpeta personal en cada uno de los equipos y su contenido no se sincronizará. Es decir, si crea un archivo en la máquina A, no lo encontrará en su carpeta cuando inicie sesión en la máquina B. Esto es porque LDAP sólo se encarga de autenticar a los usuarios.

Tenemos que conseguir unir las posibilidades de autenticación centralizada en el servidor que ofrece LDAP con la capacidad de almacenamiento centralizado que aporta NFS. El resultado serán los perfiles móviles de usuario. Es decir, un usuario encontrará su carpeta personal en todos los equipos cliente donde inicie sesión. Para ello deberemos recordar cómo funcionen els servidors i clients NFS.

Per instal.lar un servidor NFS, a Ubuntu hem d'instal.lar el paquet "nfs-kernel-server" (com a dependència s'hauria d'instal.lar automàticament, entre altres paquets, un anomenat "nfs-common"). Per instal.lar el client NFS, a Ubuntu només cal instal.lar el paquet "nfs-common". Per comprovar si tot plegat està ben instal.lat, hauria d'aparèixer "nfs4" -i si és servidor, també "nfsd"- en executar *cat /proc/filesystems* (comanda que, en general, serveix per llistar els sistemes de fitxers reconeguts pel sistema)

Per a què el servidor comparteixi carpetes, hem de modificar el contingut del seu arxiu **/etc/exports**. El contingut d'aquest arxiu ha de ser una línia per cada carpeta que es vulgui compartir ("exportar"). El format de la línia és així:

/ruta/Carpeta/Compartida nomDNSmàquinaclientdesdonespotaccedir(opcions)

on: ***** En comptes del nom DNS de la màquina client permesa (el qual pot incloure els comodins *, ? i [ ]) es pot especificar la seva ip, o bé indicar xarxes senceres permeses amb el format *ipXarxa/Mascara*

***** Després de cada nom o ip de màquina/xarxa han d'aparèixer les opcions de compartició; si hi ha vàries es separen per comes, totes elles dins del parèntesi. L'opció més important és *rw*, la qual comparteix la carpeta segons els permisos que tingui aquesta (l'opció contrària seria *ro*, la qual comparteix la carpeta sempre en mode de només lectura sense considerar els permisos que tingui).Hi ha moltíssimes altres opcions, que les pots consultar a *man exports*

NOTA IMPORTANT: No s'ha de deixar cap espai en blanc entre el nom o ip de màquina/xarxa i el parèntesi d'obertura

Un cop configurat l'arxiu /etc/exports, ja podem iniciar el servei amb la comanda *service nfs-kernel-server start* . Recordem que cada cop que modifiquem l'arxiu /etc/exports haurem de reiniciar el servidor.

Per poder accedir des d'una màquina client a la carpeta compartida pel servidor, en aquesta màquina client primer hem d'instal.lar, recordem, el paquet "nfs-common". A partir d'aquí, per veure quines carpetes compartides té un servidor, es pot executar: ***showmount -e ipservidor*** i per accedir a una carpeta compartida concreta, el client abans l'ha de muntar, així: ***mount -t nfs ipservidor:/ruta/Carpeta/Compartida /punt/local/de/muntatge***

Per comprovar si el muntatge s'ha realitzat correctament, podem executar la comanda ***mount*** o bé la comanda ***df***. En tots dos casos haurem de veure la presència del nou punt de muntatge, al qual hi podrem accedir normalment via ***cd,*** llistar el seu contingut via ***ls***, etc.

No obstant, haver d'escriure la comanda *mount* a mà cada cop que volem accedir a una carpeta compartida no és gaire pràctic (a més de que ho hem de fer com a administrador). El més habitual és muntar durant l'arranc del client les carpetes compartides que es vulguin fer servir per tenir-les ja llestes en iniciar sessió, sense haver de fer res. Per fer això, al client hem de modificar un arxiu de text anomenat **/etc/fstab**. Aquest arxiu, com ja deveu saber, no és únicament utilitzat per NFS, sinò que és molt més general: conté tots els tipus de magatzems que s'han de muntar a l'arranc del sistema (particions locals, carpetes Samba, etc).

Cada línia d'aquest arxiu consta de sis camps separats per un espai en blanc o un tabulador (per més informació, consulteu *man fstab)*:

*El 1r camp indica la partició/carpeta compartida concreta que s'haurà de muntar durant l'arranc del sistema operatiu (per exemple, la partició local "/dev/sda1")

*El 2n camp indica la ruta de la carpeta que funcionarà com el seu punt de muntatge local

*El 3r camp indica el sistema de fitxers en què està formatada aquesta partició/carpeta compartida (valors possibles són: ext4, xfs, jfs, vfat -per "fat32"-, ntfs, swap, o auto -per no especificar cap sistema en concret i què l'fstab el determini ell sol en el moment del muntatge...aquesta opció és útil per lectors d'unitats extraïbles com CD/DVDs que poden contenir diferents sistemes de fitxers depenent de la unitat introduïda-)

*El 4r camp indica les opcions de muntatge s'hi apliquen (separades per comes)

*El 5è camp actualment no s'utilitza i quasi sempre val 0

*El 6è camp indica si la comanda *fsck* s'executarà en reiniciar el sistema quan es munti aquesta partició/carpeta compartida (un valor 0 vol dir que no es comprova el sistema de fitxers, un valor 1 vol dir que sí i un valor 2 vol dir que sí, però després d'haver comprovat les particions marcades amb valor 1; normalment la partició arrel del sistema ha de tenir un 1 mentre la resta de particions pot tenir 0 o 2).

Les opcions de muntatge poden ser qualsevol de les opcions específiques de NFS4 (les llistades a *man nfs*) o bé qualsevol de les generals de la comanda *mount* (és a dir, les que estan llistades a *man mount*) . Entre aquestes darreres, hi ha que tenen més sentit dins de /etc/fstab que no en una comanda *mount* normal:

Opció	**Funció**
auto	Indica que la partició/carpeta compartida corresponent es muntarà automàticament durant l'arranc. En altres paraules: és la manera "d'activar" aquesta línia. Cal tenir en compte que aquesta opció farà que la partició/carpeta compartida estigui disponible immediatament per qualsevol usuari del sistema client (sigui o no administrador). L'opció contrària és "noauto", que serveix per "desactivar" la línia sense haver d'esborrar-la, fent que només es pugui muntar la partició/carpeta compartida corresponent de forma manual (i com a administrador) amb la comanda *mount*
exec	Permet que es puguin executar els binaris existents a la partició/carpeta compartida corresponent. L'opció contrària és "noexec".
suid	Permet reconèixer el permís especial "suid" en els binaris existents a la partició/carpeta compartida corresponent. Aquest permís especial s'utilitza per permetre a usuaris que no són administradors executar binaris que en teoria només podria executar l'usuari administrador (com ara el binari *passwd*, per exemple). L'opció contrària és "nosuid"
user	Permet que qualsevol usuari (sense ser administrador) pugui muntar amb la comanda *mount* la partició/carpeta compartida corresponent (però només l'usuari que l'hagi muntat la podrà desmuntar). Aquesta opció implica "noexec" i "nosuid", a no ser que s'indiqui el contrari. L'opció contrària (és a dir, que només l'usuari root pugui muntar la partició/carpeta compartida corresponent) és "nouser"
users	Permet que qualsevol usuari (sense ser administrador) pugui muntar amb la comanda *mount* la partició/carpeta compartida corresponent, i també que qualsevol altre usuari (o el mateix) la pugui desmuntar. Aquesta opció implica "noexec" i "nosuid" , a no ser que s'indiqui el contrari.
sync	Fa que les escriptures a la partició/carpeta compartida corresponent es realitzin de forma síncrona (és a dir, que s'apliquin inmediatament).L'opció contrària és "async", la qual implica que les escriptures es realitzin de forma asíncrona (és a dir, que totes les escriptures s'apliquin de cop passat un temps determinat). Aquestes opcions en NFS no s'utilitzen.
defaults	Equival a "rw","suid", "exec","auto","nouser" i "async"

En el cas concret de muntar carpetes compartides per NFS, el primer camp (que recordem, representa el recurs a muntar) s'escriu indicant la ip o nom del servidor NFS seguit de ":" i de la ruta de la carpeta compartida a la què es vol accedir. Per tant, un exemple d'arxiu /etc/fstab per un client NFS podria ser aquest:

```
ipservidor:/ruta/Carpeta/Compartida  /punt/local/de/muntatge  nfs4  auto,rw  0  0
```

5.-a) Instala el servidor NFS en la máquina "ServidorLDAP" (***sudo aptitude install nfs-kernel-server***) y crea en ella una carpeta que alojará las carpetas personales de los usuarios. Si decidimos, por ejemplo, que esta carpeta sea "/opt/perfiles", los comandos a ejecutar son: ***sudo mkdir /opt/perfiles && sudo chown nobody:nogroup /opt/perfiles*** .

b) En esa máquina debes crear también las carpetas personales individuales de cada usuario (***sudo mkdir /opt/perfiles/jlopez, sudo mkdir /opt/perfiles/pperez***, etc) y asignar a cada una su propietario respectivo (***sudo chown usu1ldap:grupoldap /opt/perfiles/jlopez, sudo chown usu2ldap:grupoldap /opt/perfiles/pperez***, etc)
NOTA: Fijarse que podemos ejecutar el comando *chown* con los usuarios LDAP porque éstos también están reconocidos como usuarios válidos en el propio servidor.

c) Aún en el servidor, modifica el archivo **/etc/exports** para compartir el directorio anterior con permisos de lectura/escritura para todos los usuarios (debería quedar una línea parecida a ésta: /opt/perfiles *(rw)). Seguidamente reinicia el servidor

d) Inicia ahora la máquina "ClienteLDAP" e instala el paquete "nfs-common". Seguidamente, crea una carpeta que hará de punto de montaje de los perfiles móviles (por ejemplo, si decidimos que el punto de montaje sea la carpeta "/opt/punto", los comandos a ejecutar serán: ***sudo mkdir /opt/punto && sudo chmod 777 /opt/punto***).

e) Haz que el montaje de "/opt/perfiles" en "/opt/punto" se produzca nada más arrancar la máquina cliente. Para ello, debemos añadir la siguiente línea a su archivo **/etc/fstab** (¡y después reiniciarlo o ejecutar *mount -a*!): 192.168.0.1:/opt/perfiles /opt/punto nfs auto,rw,noatime 0 0

f) Modifica (en el servidor) las cuentas de usuario LDAP existentes para indicar que su carpeta personal (atributo homeDirectory) se encuentra dentro de la carpeta /opt/punto (por ejemplo, /opt/punto/jlopez, /opt/punto/pperez,...). Repasa el apartado *a)* del ejercicio 3 para recordar cómo se hacía esto.

g) Inicia sesión en un terminal virtual de la máquina cliente con un usuario LDAP y crea en su carpeta personal un archivo (con *touch* mismo). Comprueba que dicha carpeta personal es remota observando que efectivamente el fichero se ha creado en el servidor, dentro de la subcarpeta /opt/perfiles/carpeta_personal correspondiente.

h) Inicia sesión gráfica en la máquina cliente con un usuario LDAP, abre el Nautilus y crea una carpeta dentro del escritorio. Comprueba que dicha carpeta se ha creado en el servidor, dentro de la subcarpeta /opt/perfiles/carpeta_personal/Dekstop correspondiente.