Autenticación/Identificación de usuarios con LDAP

Teoria 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 llamados "entradas" o "nodos", 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 su sistema de almacenamiento es diferente y su manera de consultar y manipular la información contenida en él también.

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 e identificación centralizada de usuarios (nombre, contraseña, uid, grupo, permisos, etc), o con la autenticación e identificación 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/s “Regla/s de Esquema” concreta/s utilizada/s por una entrada determinada (porque una entrada puede tener asociadas varias reglas, acumulándose entonces en la entrada todos los atributos presentes en cada una de ellas) se indica/n mediante un atributo especial de la entrada llamado "objectClass"; el valor de este atributo representa una Regla elegida (si hay varias, cada una se especificará en un atributo "objectClass" diferente). Lo más normal es asignar al atributo "objectClass" de una entrada el nombre de una Regla ya predefinida a escoger de entre un conjunto de Reglas estandarizadas que ofrecen todos los servidores LDAP (como por ejemplo las que definen los objetos de tipo "usuario Linux" o "grupo Linux"); también podríamos crear nuestras propias Reglas de Esquema (o modificar alguna existente; para más información sobre cómo hacer ambas cosas consultar por ejemplo http://www.zytrax.com/books/ldap/ch3), pero no suele ser necesario porque las Reglas predefinidas cubren la mayoría de casos prácticos y son oficiales.

Además del atributo *objectClass*, que sirve para indicar el tipo de entrada (es decir, la Regla que la define), no existe ningún otro atributo obligatorio porque, tal como hemos dicho, todos los que pudiera tener la entrada dependerán de la Regla (o Reglas) utilizada/s. No obstante, en el caso habitual de manejar entradas cuya Regla/s asignada/s represente/n empleados de una empresa, es muy habitual encontrar que dichas entradas tienen atributos tales como los siguientes (entre otros):

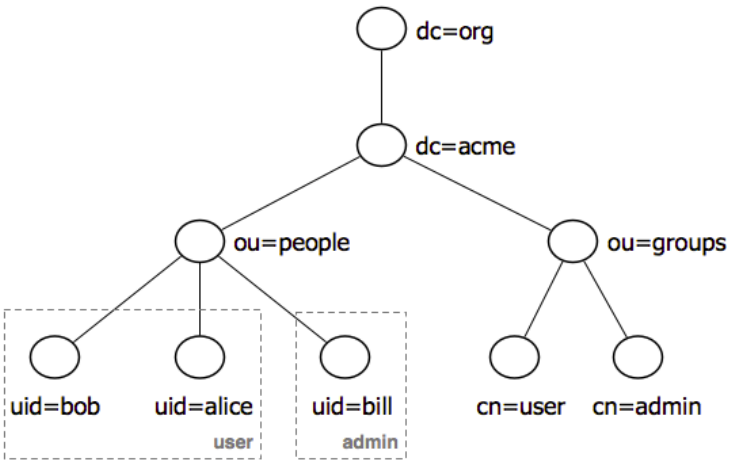
uid (user id): Identificador de la entrada. ¡No confundir con el uid de un usuario Linux!
cn (common name): Nombre de la persona representada en la entrada
sn (surname): Apellido de la persona.
mail: dirección de correo electrónico de la persona.
o (organization): Departamento al que pertenece la persona.
ou (organizational unit): Contenedor estructural (a modo de “carpeta”) dentro del cual está categorizada esta entrada. Este contenedor deberá ser creado previamente como otra entrada

NOTA: En el caso de carecer de atributo “uid”, el atributo que suele hacer de identificador de la entrada suele ser “cn”.

NOTA: Una Regla que TODOS las entradas de un árbol de directorio han de implementar (mediante su correspondiente atributo "objectClass"), independientemente del resto de Reglas que se deseen añadir en cada una de ellas, es una Regla llamada "top". Al implementar esta Regla en una entrada lo que se consigue es que esa entrada pueda disponer precisamente del atributo "objectClass" (es decir, el atributo "objectClass" está definido en la Regla "top", a modo de definición recursiva).

Las entradas se organizan en una estructura jerárquica en forma de árbol invertido. Tradicionalmente, la parte superior de esta estructura refleja la jerarquía de los dominios DNS (incluso regionales) 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 (dc=”pepsi” y dc=”com” para “pepsi.com”, dc=”unicef” y dc=”org” para “unicef.org”, etc). Dentro de las últimas es donde se encuentra 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, de hecho, se construye a partir del identificador de la entrada en sí misma (lo que se llama “Nombre Relativo Distinguido” -”Relative Distinguished Name” (RDN)-, y que 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”) que contiene información sobre el empleado Pperez perteneciente a la sección española de Nike. Para conocer toda esa información, deberíamos observar el resto de atributos de esa entrada (objetClass, cn, givenname, sn, o,mail ...).



NOTA: Una explicación más pormenorizada sobre cómo se organiza la estructura en árbol de un directorio LDAP se encuentra en https://fy.blackhats.net.au/blog/html/pages/ldap_guide_part_1_foundations.html

LDAP tiene definidas las operaciones necesarias para interrogar y actualizar el directorio (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 busquen 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 (ruta de su carpeta personal, shell preferido, grupo/s de usuarios al que pertenece, etc) para poder asignarle los permisos adecuados. Es decir, usaremos el servidor LDAP como centro de autentificación e identificación de los usuarios de nuestra res.

Implementaciones de servidores LDAP:

Existen varios servidores LDAP libres:

389 Directory Server (http://www.port389.org) . Desarrollado por RedHat

OpenLDAP (http://www.openldap.org)

Apache DS (http://directory.apache.org/apacheds). Incluye servidor Kerberos integrado.

OpenDJ (https://github.com/OpenIdentityPlatform/OpenDJ)

También hay que destacar la existencia de soluciones integradas formadas por un servidor LDAP más otros servidores que complementan la funcionalidad de dicho servidor LDAP, facilitando en gran medida la integración de todos estos servicios (muy habitualmente utilizados en conjunto) entre sí. Ejemplos son:

FreeIPA (http://www.freeipa.org) : Servidor LDAP 389 Directory Server + Servidor Kerberos (tipo MIT) + Servidor DNS propio + Servidor NTP propio + Autoridad Certificadora propia +Servidor HTTP propio (para gestionarlo todo vía web). Es la solución elegida por Fedora

Samba4 (http://www.samba.org) : Servidor LDAP propio + Servidor Kerberos (tipo Heimdal) + Servidor DNS propio + Servidor NTP propio + Servidor de compartición de carpetas.

Active Directory (https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/active-directory-domain-services) : "Suite" de servidor LDAP + Kerberos + DNS + NTP (entre otros) desarrollada por Microsoft e integrada en las versiones "Server" de sus sistemas Windows para poder almacenar y gestionar de forma centralizada la información de sus dominios de administración (usuarios, equipos, configuraciones, permisos, etc).

Autenticació/Identificació a la màquina "client":

Un cop tinguem un servidor LDAP funcionant i amb un directori omplert i accessible a través de la xarxa, ja podrà ser consultat i/o modificat amb les eines client pertinents (tant de tipus terminal com gràfiques) des d'una màquina remota qualsevol sense gaires dificultats. No obstant, si el que volem és fer servir el contingut del directori (concretament, nodes de tipus "usuari Linux" -contenint tots els atributs necessaris per definir-lo: UID, contrasenya, etc- i nodes de tipus "grup Linux" -contenint tots els atributs necessaris per definir-lo: GID, etc-) com a origen de dades durant procés d'inici de sessió en una màquina client, caldrà alterar aquest procés per a què canvïi el seu origen de dades per defecte, que sol ser normalment els arxius "/etc/passwd, "/etc/group" i "/etc/shadow". Més concretament, "canviar l'origen de dades del procés d'inici de sessió a la màquina client" significa a la pràctica canviar dues coses en aquesta màquina client:

*Modificar la configuració de la pila de mòduls PAM utilitzats per tal de què s'empri algun mòdul específic (n'hi ha varis que estudiarem properament: pam_ldap, pam_sssd...) que faci la consulta d'usuari<->contrasenya pertinent contra un servidor LDAP convenientment indicat.

*Modificar la configuració del backend NSS utilitzat per la llibreria libc (i, per tant, per la gran majoria de programes de Linux) per tal de què contacti també amb el servidor LDAP adient i així obtingui la resta de dades necessàries per completar la identificació de l'usuari (UID, GID, grups addicionals, ruta de la seva carpeta personal, ruta del shell per defecte, data de caducitat de la contrasenya, etc)

A continuació s'explica una mica més la funcionalitat d'aquests dos passos:

***PAM (fase de autenticación):**

PAM (de "Pluggable Authentication Modules") es, tal como ya sabemos, un conjunto de librerías especializadas en el ámbito de la autenticación que permite a los programas que hagan uso de ellas (por ejemplo: *ssh*, *su*, *sudo*, *login*, *gdm*, etc) poder utilizar distintos métodos de autenticación a conveniencia de forma totalmente transparente para dichos programas. Es decir: PAM permite que, sin realizar modificaciones en el código de los programas, podamos utilizar métodos que vayan desde el uso típico de un nombre de usuario y una contraseña, hasta dispositivos que faciliten identificación biométrica (lectores de huellas, de voz, de imagen, etc.) simplemente indicándoselo en determinados ficheros de configuración.

***NSS (fase de identificación):**

PAM solo se encarga de la autenticación: es decir, de decidir si el usuario en cuestión "puede entrar" en el sistema o no según si determinadas condiciones resultan correctas o no (condiciones que dependerán del método utilizado pero que pueden ser, por ejemplo, -y sería el caso más habitual- una combinación de nombre de usuario + contraseña). No obstante, para que el inicio de sesión de un usuario sea completo y dicho usuario sea reconocido como tal sin errores, el sistema debe conocer más datos de ese usuario. Por ejemplo, ha de saber a qué grupo/s pertenece ese usuario, o qué shell por defecto tiene, o cuál es la ruta de su carpeta personal, o la fecha de caducidad de su contraseña, etc. De obtener toda esta información complementaria del usuario recién autenticado se encarga otro conjunto de librerías llamado NSS (de "Name Service Switch", con ruta en /lib/libnss_* o /lib/x86_64-linux-gnu/libnss_*). En otras palabras: PAM se dedica a responder a la pregunta "¿Fulanito puede entrar?" y NSS se dedica a responder a la pregunta "¿Qué sabemos de Fulanito?"

Normalment, NSS obté la informació necessària dels arxius /etc/passwd, /etc/group i /etc/shadow perquè així està indicat a l'arxiu /etc/nsswitch.conf (a les línies "passwd", "shadow" i "group", respectivament). Per tant, si no canviem res del subsistema NSS i només configuréssim el mòdul PAM per a què l'autenticació es produís, per exemple, contra un servidor LDAP remot, "ens hauríem quedat a mig camí" perquè només en el cas de què el mateix nom de l'usuari existeixi tant dins del servidor LDAP com dins dels arxius /etc/passwd i /etc/shadow l'inici de sessió funcionaria ja que, tal com hem dit, es faria servir el servidor LDAP per autenticar (via PAM) i la informació NSS estàndar (via fitxers "/etc/passwd" i família) per completar la informació sobre aquest usuari. Si el que volem és tenir un sistema que inclogui informació NSS, independent de la estàndar (la qual, per què no, podria estar emmagatzemada igualment en el mateix servidor LDAP), serà necessari instal.lar un mòdul NSS que afegeixi a aquest subsistema la possibilitat de buscar aquest tipus d'informació en aquest altre origen diferent de l'estàndar. Per decidir quin és l'origen escollit entre varis possibles, les aplicacions que fan ús de NSS consulten primer l'arxiu "/etc/nsswitch.conf" per tal de saber si la informació sobre els usuaris l'han d'obtenir dels fitxers locals ("/etc/passwd" i "/etc/shadow") o bé, mitjançant l'ús d'algun mòdul NSS concret prèviament instal.lat, l'han d'obtenir d'alguna base de dades remota com podria ser un servidor LDAP (o, per què no, complementar ambdós orígens per a què si falla el predeterminat s'utilitzi l'altre).

> NOTA: En realitat, el subsistema NSS és un mecanisme molt més global que serveix per escollir altres orígens d'informació que no tenen res a veure amb "característiques d'usuaris". Per exemple, 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. Així doncs, en general, el que permet el subsistema NSS es que las aplicaciones que hayan sido programadas haciendo uso de él puedan utilizar de forma coherente y común un conjunto de orígenes de datos conteniendo los nombres de distintos ítems, (como máquinas, redes, nombres de protocolos, usuarios, grupos, etc.) que dichas aplicaciones necesitan.

> NOTA: Una altra avantatge de què una aplicació faci servir NSS és que poden obtenir la informació que necessiten (nombres de usuarios y sus contraseñas, nombres de máquinas, de redes, de protocolos, etc) preguntando directamente a NSS (y éste utilizando la librería concreta adecuada según lo que se encuentre en /etc/nsswitch.conf) sin tener que conocer el lugar exacto donde está almacenada, (ya que para obtenerla ya se encarga NSS).

Un ejemplo de la parte más relevante para nosotros del archivo "/etc/nswitch.conf" sería el siguiente:

```
passwd: files sssd
group: files sssd
shadow: files sssd
```

...donde principalmente se indica (mediante la lína *passwd:*...) que la base de datos de usuarios que primero se va a consultar es el fichero local "/etc/passwd" (gracias al valor *files*) y que, si allí no se encontrara el usuario buscado se pasará a utilizar (gracias al valor *sssd*) una librería especial llamada SSSD (de la cual hablaremos más adelante) que permite acceder a una base de datos LDAP; además también se indica que la base de datos de grupos que primero se irá a consultar es el fichero local "/etc/group" (mediante la línea *group:*...) pero que, igualmente, si allí no se encontrara el grupo buscado se pasará a utilizar la librería SSSD; finalmente, también se indica que la base de datos de contraseñas que primero se irá a consultar es el fichero local "/etc/shadow" (mediante la línea *shadow:*...) pero que si allí no se encontrara lo buscado se pasará también a utilizar la librería SSSD.

NOTA: El archivo "/etc/nsswitch.conf" contiene más líneas además de las indicadas anteriormente. Por ejemplo, es muy habitual que disponga además de una línea similar a hosts: files dns , la cual sirve para definir los orígenes donde buscar los nombres de máquinas: en primera instancia estos nombres se consultarán en el fichero local "/etc/hosts" (gracias al valor *files*, ver *man hosts* para más información o también http://en.wikipedia.org/wiki/Hosts_%28file%29), pero, si allí no se encontrara el nombre en cuestión, entonces se procederá (gracias al valor *dns*) a realizar una búsqueda DNS en los servidores configurados en "/etc/resolv.conf"

NOTA: Hay otras líneas del archivo "/etc/nsswitch.conf" que no son tan utilizadas pero que las comentamos también: la línea networks: files indica que los nombres de redes se buscarán en el fichero local "/etc/networks" (ver *man networks* para más información sobre este fichero), la línea protocols: files indica que los nombres de los protocolos se buscarán en el fichero local "/etc/protocols" (ver *man protocols* para más información sobre este fichero), la línea services: files indica que los nombres de los servicios se buscarán en el fichero local "/etc/services" (ver *man services* para más información sobre este fichero) y la línea ethers: files indica que la correspondencia estática entre direcciones MAC e IPs se buscará en el fichero local "/etc/ethers" (ver *man ethers* para más información sobre este fichero).