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

Configuración de un cliente y servidor OpenLDAP para autentificación

Per **Carles Pina i Estany**, ***cpina*** (http://pinux.info)

Creado el 07/03/2004 11:15 modificado el 07/03/2004 11:15

En este artículo veremos como instalar un servidor LDAP para autentificar a los clientes y también como configurar los clientes LDAP para que se autentifiquen contra este servidor

Recordar que LDAP es un "Directorio Ligero", que puede servir tanto para autentificar a los clientes como para recopilar otros tipos de información (desde colecciones de distribuciones Linux a resolver nombres).

Versió en català disponible aquí[1] (Catux[2])

Introducción

Usaremos la implementación OpenLDAP, que podemos encontrar en http://www.openldap.com[3]. De todas formas aquí nos basaremos en los paquetes Debian (sid, pero es parecido a todas) para hacerlo.

Como nota general, no soy un experto de LDAP sinó que he tenido que instalarlo un par de veces y esos son mis apuntes para conseguirlo, espero que sean útiles a alguien.

Podemos ver a LDAP como una base de datos optimizada para hacer un número muy alto de lecturas y muy pocas escrituras o modificaciones. Además está organizado de modo jerárquico.

Muchas veces (como en este artículo) se usa LDAP para guardar información de los usuarios. Podemos guardar lo típico de login, UID, GID, contraseña, etc. y además añadirle una foto, teléfono de casa u otras informaciones qué queramos. Recalcar que en este artículo guardaremos la información necesaria para un servidor de autentificación; no para un servidor de ficheros. Si necesitamos un servidor de ficheros tendremos que usar NFS (o Coda, Intermezzo, etc.)

Instalación del servidor

En el servidor necesitaremos los paquetes *libldap2 slapd*. Podemos instalar también el paquete *ldap-utils* que nos servirá para hacer ciertas pruebas.

Una vez instalado el servidor OpenLDAP iremos al directorio */etc/ldap* y modificaremos el fichero *slapd.conf*. En este fichero modificaremos al menos el suffix poniendo por ejemplo nuestro dominio (no hace falta que esté registrado, es para referirnos a él desde LDAP). Lo pondremos con el formato `suffix "dc=pinux,dc=info"`.

Añadiremos también:

```
rootdn "cn=admin,dc=pinux,dc=info"  #para autentificar al administrador
rootpw secret                       #la contraseña
```

(la contraseña la podemos encriptar, al final del documento vemos como)

En el resto del fichero de configuración sólo tendremos que cambiar el cn= y el dc=pinux,dc=info donde veamos que hace falta.

En este momento podremos hacer un `/etc/init.d/slapd restart` para reiniciar el servidor LDAP.

Si ejecutamos `slapcat` nos tendría que dar información sobre nuestro LDAP (normalmente imprime por pantalla todoo el LDAP, de momentos sólo veremos algo de estructura).

Alta, búsqueda y eliminación de usuarios

Ahora lo que tendremos que hacer es dar de alta a algunos usuarios para después poder usarlos en la autentificación.

Antes de dar de alta a los usuarios y grupos, tendremos que dar de alta al administrador, un usuario para "buscar" la información sin privilegios y ya después algun usuario normal para verificar que nos funciona correctamente el sistema.

La forma de dar de alta a usuarios (o datos en general) en LDAP suele ser mediante ficheros .ldif. Un fichero .ldif es un fichero que contiene la información que vamos a añadir, para después añadirla a nuestro directorio.

antes de empezar a dar de alta a los usuarios crearemos la "estructura" mediante este fichero .ldif:

```
dn: ou=Group,dc=pinux,dc=info
objectClass: top
objectClass: organizationalUnit
ou: Group

dn: ou=People,dc=pinux,dc=info
objectClass: top
objectClass: organizationalUnit
ou: People
```

De esta forma tenemos a dos "Organization Units" que son los grupos (Groups) y los usuarios (People).

Vemos que estamos usando el *objectClass: organizationalUnit*, de esa forma LDAP ya sabe qué campos tendrás nuestros usuarios/grupos. Para decirle a LDAP que inserte los datos del fichero .ldif en la base de datos ejecutaremos:

```
ldapadd -x -D 'cn=admin,dc=pinux,dc=info' -w secret -f fichero.ldif
```

Notas:

- Con -x nos autentificamos de forma simple a LDAP sin usar SASL
- Al -D le decimos el Distinguised Name, el mismo que le pusimos en el fichero de configuración
- Al -w le pasaremos la contraseña. Con -W nos la pediría de forma interactiva
- Al -f sirve para pasarle el fichero de configuración

Seguidamente, daremos de **alta un usuario**. Seguiremos el mismo esquema que hasta ahora, haciendo un fichero .ldif y después ejecutando la utilidad `ldapadd`. El fichero .ldif para dar de alta a un usuario puede ser uno como este:

```
#
# New Tester user
#
dn: uid=tester,ou=People,dc=pinux,dc=info
objectClass: top
objectClass: account
objectClass: posixAccount
uid: tester
cn: Test User
userPassword: hola
gecos: Test User
uidNumber: 2000
gidNumber: 2000
homeDirectory: /home/tester
loginShell: /bin/bash
```

Y a continuación:

```
ldapadd -x -D 'cn=admin,dc=pinux,dc=info' -w secret -f fichero.ldif
```

Ahora sí que podemos ejecutar `slapcat` y ver si el usuario tester está correctamente listado.

Al hacer un `slapcat` él mismo nos está haciendo un listado en format .ldif (lo podriamos aprovechar para lo que haga falta).

Ya que estamos dando de alta a usuarios, aprovechamos y damos de alta a un grupo. Para hacerlo, crearemos un fichero .ldif con este contenido:

```
#
# Testing Group#
dn: cn=testing,ou=Group,dc=pinux,dc=info
objectClass: top
objectClass: posixGroup
cn: testing
gidNumber: 2000
```

Ahora ya tenemos a un grupo con GID 2000.

También podemos aprovechar para **hacer búsuqedas:**

```
ldapsearch -x -b 'uid=tester,ou=People,dc=pinux,dc=info'
```

Y por último, si queremos **eliminar un usuario** podemos hacerlo de esa forma:

```
ldapdelete -x -D 'cn=admin,dc=pinux,dc=info' -w secret\
  'cn=Local Root,ou=People,dc=pinux,dc=info'
```

Configurando un cliente LDAP

El objetivo de esta parte es poder hacer un `slapcat` desde una máquina que no sea el servidor y funcione correctamente.

Para conseguir eso tendremos que intalar al menos los paquetes *libldap2, ldap-utils*.

Una vez instalados estos paquetes, editaremos el fichero `/etc/ldap/ldap.conf`. Este fichero es el de configuración del cliente LDAP, no lo tenemos que confundir confundir con el fichero `slapd.conf`, que es de la configuración del servidor.

El fichero podemos dejarlo así:

```
host 192.168.1.2
base dc=pinux,dc=info
```

De esa forma le decimos donde tiene que conectarse y el dc.

Ahora desde la máquina cliente podemos hacer uso del `slapcat` igual que antes. La configuración por defecto permite la lectura de varios campos por "todo el mundo", así que aunque no veremos la contraseña podremos ver otra información del usuario.

Configurando un cliente, parte nsswitch

Para esta parte necesitaremos instalar el paquete `libnss-ldap`.

Cuando trabajamos con el sistema (`ls -l`, p. ej.) normalmente vemos los nombres de los usuarios propietarios de los ficheros. En cambio guardado en el disco hay el "número" (UID) del usuario.

Para que los programas sepan el nombre que corresponde a los UID's (y otras cosas, como grupos, hosts, etc.) hacen

unas llamadas a funciones de la librería GLIBC, y es esta quien "averigua" la relación.

Es en el fichero `/etc/nsswitch.conf` donde le decimos al sistema de donde averiguar el propietario sabiendo el UID. Normalmente contiene algo como:

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

hosts:          files dns
networks:       files
```

Lo que más nos interesa es la parte de passwd, group y shadow. Ahora lo dejaremos así:

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

Por tanto cuando un programa le pide a la GLIBC "dime el nombre del usuario 1005", la GLIBC primero mira en `/etc/passwd` y sinó hará la consula al servidor LDAP.

Para que el nsswitch pueda hacer las consultas en el LDAP tendremos el fichero `/etc/libnss-ldap.conf` algo como:

```
host 192.168.1.2
base dc=pinux,dc=info
```

Es decir, la información necesaria para llegar a nuestro servidor LDAP y lanzar la consulta.
Si hacemos `man libnss-ldap.conf` veremos las opciones que podemos ponerle (p. ej. port, ldap_versions, etc.)

Entre otras cosas, a veces necesitaremos que se haga una conexión autentificada contra el servidor. Para eso se usará la contraseña que encuentre en `/etc/ldap.secret` (tiene que estar con permisos 600, propietario y grupo root)

Configurando el cliente PAM

Para poder configurar el cliente PAM tendremos que instalar el paquete `libpam-ldap` Hay varios programas que pueden usar (y usan por defecto) un método de autentificación "centralizado" y por módulos llamado PAM (Pluggable Authentication Modules). Eso son unas librerias que los programas pueden soportar que sirven de "interfaz" contra varios métodos de autentificación (p. ej. LDAP)

La configuración en Debian es en el directorio `/etc/pam.d/` y tenemos un fichero de configuración por cada servicio.

Si es necesario que la conexión sea con privilegios, se usará la contraseña que se encuentre en `/etc/ldap.secret`, y que estará con permisos 600 (como el punto anterior).

Tener la contraseña para autentificarse es necesario, con la configuración por defeto del `slapd.conf`, cuando el usuario root quiere cambiar la contraseña de otro usuario: si no es conexión autentificada, el servidor LDAP no le deja cambiarla. De otra forma cualquier usuario podría cambiar la contraseña de cualquier otro, solo lanzando la consulta al servidor LDAP.

Seguidamente, dejaremos el fichero `/etc/pam_ldap.conf` de forma parecida:

```
host 192.168.1.2
base dc=pinux,dc=info
ldap_version 3

rootbinddn cn=admin,dc=pinux,dc=info
# don't forget /etc/ldap.secret
```

Ahora ya tenemos la configuración general de PAM para funcionar con LDAP.
Pasemos a la parte específica de cada servicio (ssh, su, passwd, etc.).
Estaremos tocando los ficheros de configuración del cliente para que se autentifique contra el servidor.

ssh

Tenemos que ir al fichero `/etc/pam.d/ssh` y al menos añadir esas líneas:

```
auth       sufficient   pam_ldap.so
account    sufficient   pam_ldap.so
session    sufficient   pam_ldap.so
password   sufficient   pam_ldap.so
```

al inicio del fichero.

En el caso del ssh tendremos seguramente que modificar en el fichero `/etc/ssh/sshd_config`el parámetro `PAMAuthenticationViaKbdInt` a `yes`. De otra forma no autentificaría de forma correcta.

su

Sirve para poder ejecutar el `su` con usuarios que están dados de alta en el LDAP.

En el fichero `/etc/pam.d/su` lo dejaremos parecido a:

```
auth       sufficient pam_rootok.so
auth       sufficient pam_ldap.so
auth       required   pam_unix.so use_first_pass

account    sufficient pam_ldap.so
account    required   pam_unix.so

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

passwd

Este es para permitir cambiar las contraseñas de los usuarios. Lo podemos dejar así:

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

Es bastante útil dar de alta a los usuarios de forma normal y cambiarles la contraseña con el mismo passwd como root (si son pocos usuarios de pruebas, claro)

login

Lo dejaremos parecido a este:

```
auth        required      pam_nologin.so
auth        sufficient    pam_ldap.so
auth        sufficient    pam_unix.so shadow use_first_pass
auth        required      pam_deny.so
```

Hay otras maneras de dejarlo, pero tenemos que ir con cuidado con el `use_first_pass`: si no lo ponemos los usuarios que estan dados de alta en LDAP se les pediría dos veces la contraseña (una validaría con `/etc/passwd`, y al no encontrar el usuario se lo pediría otra vez para validarlo contra LDAP.

Con los ejemplos vistos hasta ahora sería fácil hacer lo mismo en otros servicios (p. ej. proftpd, xlock, etc.)

También es relativamente fácil modificar los ficheros `common-account common-auth common-password common-session` para no tener que tocar el fichero de cada servicio, a costa de tener menos "personalización" por servicio.

Podemos mejorar

Tenemos varias cosas a hacer, sólo comento algunas.

- Primero de todo, la contraseña "secret" que tenemos en texto "llano" en el fichero ponerla encriptada (y una contraseña de verdad). Para hacerlo ejecutaremos `slappasswd` y la que nos diga la pondremos en el fichero. Por ejemplo: `{SSHA}xAR4MvR0AByRx0gYCZGKeWUFAhNGZIud`.
- También deberíamos usar certificados para que nadie pueda suplantar nuestro servidor. Y ya que nos ponemos, que los datos transferidos de forma segura. El mismo OpenLDAP puede hacerlo, lo podemos leer en este otro artículo[4] de Bulma. Eso nos interesará sobretodo si clientes LDAP y servidor LDAP no están en la misma red (o que esa no sea de confianza). Otra forma de hacer eso mismo sería con freeswan[5] (una implementación de IPSEC) o haciendo túneles seguros con otras aplicaciones.
- Un caso bastante normal es querer migrar los usuarios que ya teniamos en el sistema a OpenLDAP. Lo podemos conseguir con el paquete migrationtools[6]
- Un último apunte importante, es instalar el paquete `nscd`. Este paquete hará de caché para OpenLDAP en la máquina local. De otra forma, cada vez que hagamos un "ls -l" el sistema irá a preguntar al servidor LDAP "¿este UID, qué nombre tiene?" (hecho que carga el servidor, la red y además el cliente responde más lento)

Enlaces

Varios enlaces interesantes:

- La Web de OpenLDAP[7]
- Una Web sobre OpenLDAP en Debian[8]
- Para saber como se empieza a llenar[9] OpenLDAP
- Los dos geniales artículos de Bulma de LDAP: servidor[10] y cliente[11]

Lista de enlaces de este artículo:

1. http://catux.org/index.php?contingut=articles&menu=6&num=33
2. http://www.catux.org
3. http://www.openldap.com
4. http://bulma.net/body.phtml?nIdNoticia=1343&nIdPage=3
5. http://www.freeswan.org
6. http://packages.debian.org/cgi-bin/search_packages.pl?keywords=migrationtools&am
7. http://www.openldap.org
8. http://homex.subnet.at/~max/ldap
9. http://sapiens.wustl.edu/~sysmain/info/openldap/openldap_populate.html
10. http://bulma.net/body.phtml?nIdNoticia=1343
11. http://bulma.net/body.phtml?nIdNoticia=1371

E-mail del autor: carles _ARROBA_ pinux.info
Podrás encontrar este artículo e información adicional en: http://bulma.net/body.phtml?nIdNoticia=1991