OpenLDAP et Kerberos

mars 2013

Index

Description de l'environnement

OS : Ubuntu 12.10

OpenLDAP

OpenLDAP fournit les services d'annuaires.

Il s'agit d'un service qui permet de stocker des informations destinées à être souvent consultées et qui repose sur tcp/ip. C'est une version simplifiée du protocole x500.
Un annuaire peut être assimilé à une base de données spécialisée dans l'accès et la présentation des informations. Il possède une structure arborescente constituée d'une racine, de branches et de feuilles. La nomenclature des différents éléments permet une représentation spatiale des éléments du réseau.

La principale limitation dans un réseau réside dans la difficulté d'échanger des informations entre différents systèmes.

Un annuaire est optimisé pour la recherche (lecture) et le stockage d'informations, et donc ne correspond pas à des données trop dynamiques.

Le site de référence : http://www.openldap.org/

Introduction

Historique

```
1989 :      x500
1992 :      LDAP (University of Michigan)
1993 :      RFC 1487
1995 :      Premiers serveurs LDAP
1998 :      OpenLDAP version 1.0
2000 :      OpenLDAP version 2.0
```

Terminologie

```
ldap :      Lightweight Directory Access Protocol
dc :        Domain Component
cn :        Common Name
dn :        Distinguished Name
rdn :       Relative Distinguished Name
```

Tcp/ip et openLDAP

Un client se connecte sur un serveur ldap qui normalement sur le port tcp 389, ou sur le port 636 en mode sécurisé. Il peut alors envoyer plusieurs requêtes et attend les réponses du serveur. Une fois terminé, il coupe la connexion.

Installation sous Ubuntu 12.10

Voici la liste minimale des paquets à installer :

- slapd # le serveur
- ldap-utils # des outils ldap

```
$ sudo apt-get install slapd ldap-utils
```

Il faut spécifier le mot de passe de l'administrateur.

Pour obtenir la liste des outils ldap installés, on utilise

```
$ dpkg -L ldap-utils | grep bin
/usr/bin
/usr/bin/ldapexop
/usr/bin/ldappasswd
/usr/bin/ldapdelete
/usr/bin/ldapcompare
/usr/bin/ldapsearch
/usr/bin/ldapwhoami
/usr/bin/ldapurl
/usr/bin/ldapmodrdn
/usr/bin/ldapmodify
/usr/bin/ldapadd
```

Première configuration

Arbre de base

```
dc=local
|
dc=gull
|
cn=admin
```

On reconfigure le paquet à l'aide de la commande suivante.

```
$ sudo dpkg-reconfigure slapd
```

Il faut alors répondre à une série de questions :

```
Voulez-vous omettre la configuration d'OpenLDAP? Non
Nom de domaine : gull.local
Nom d'entité : linux
Mot de passe de l'administrateur : gull (2x)
Module de base de données à utiliser : HDB
Faut-il supprimer la base de données à la purge du paquet? Oui
Faut-il déplacer l'ancienne base de données? Oui
Faut-il autoriser le protocole LDAPv2? Non
```

Extension de l'arbre

On désire agrandir notre structure de base et ajouter deux conteneurs et des utilisateurs/groupe

```
dc=local
|
dc=gull
|
+----------------------+----------------------------+
|                      |                            |
cn=admin               ou=users                     ou=groups
                       |                            |
                       ?                            linux
```

On peut utiliser des fichiers textes ldif pour remplir la base de données.

```
$ cat base.ldif
dn: ou=users,dc=gull,dc=local
objectClass: organizationalunit
ou: users
description: Les Utilisateurs

dn: ou=groups,dc=gull,dc=local
objectClass: organizationalunit
ou: groups
description: Les Groupes
```

Voici comment ajouter les enregistrements.

```
$ ldapadd -x -D "cn=admin,dc=gull,dc=local" -W -f base.ldif
Enter LDAP Password:
adding new entry "ou=users,dc=gull,dc=local"
```

```
adding new entry "ou=groups,dc=gull,dc=local"
```

Ajout des utilisateurs

Pour faciliter la tâche, voici deux scripts, et une liste de noms d'utilisateur.

```
$ cat useradd_ldap.sh
#!/bin/sh
DEFAULT_SHELL=/bin/bash
i=10000                 # start uidNumber
DEFAULT_GID=10000       # default gidNumber
DEFAULT_GROUPNAME=linux
DEFAULT_HOME=/home/
DEFAULT_PASSWD=gull
DEFAULT_ENCRYPT_PASSWD=$(/usr/sbin/slappasswd -h {CRYPT} -s $DEFAULT_PASSWD)
# Users Organizational Unit
USERS_OU="ou=users, dc=gull, dc=local"
GROUPS_OU="ou=groups, dc=gull, dc=local"
#
while read text
do
  UID=$(echo $text | cut -d \  -f 1)
  NAME=$(echo $text | cut -d \  -f 2)
  MAIL="${UID}.${NAME}@gull.local"
  UP_NAME=$(echo $NAME | sed 's/^./\u&/')
  # Affichage
  echo "dn: cn=$UID $UP_NAME, $USERS_OU" # Distinguished Name (Unique)
  echo "cn: $UID $UP_NAME"               # Common Name
  echo "uid: $UID"
  echo "sn: $UP_NAME"                   # Surename (Last Name)
  echo "mail: $MAIL"                    # Email
  echo "objectClass: top"
  echo "objectClass: person"
  echo "objectClass: organizationalPerson"
  echo "objectClass: inetOrgPerson"
  echo "objectClass: posixAccount"
  echo "objectClass: shadowAccount"
  echo "userPassword: $DEFAULT_ENCRYPT_PASSWD"
  echo "uidNumber: $i"
  echo "gidNumber: $DEFAULT_GID"
  echo "gecos: $UID"
  echo "homeDirectory: $DEFAULT_HOME$UID"
  echo "loginShell: $DEFAULT_SHELL"
  echo "shadowLastChange: 11547"
  echo "shadowMax: 99999"
  echo "shadowWarning: 7"
  echo
  # On augmente l'uidNumber
  i=$(expr $i + 1)
  # Membres du groupe
  MEMBER="$MEMBER,$UID"
done < $1
#
echo "dn: cn=$DEFAULT_GROUPNAME, $GROUPS_OU"
echo "cn: $DEFAULT_GROUPNAME"
echo "objectClass: posixGroup"
echo "objectClass: top"
echo "gidNumber: $DEFAULT_GID"
#
```

```
echo "memberuid: $(echo $MEMBER | cut -b 2-)"
```

... pour supprimer ...

```
$ cat userdel_ldap.sh
#!/bin/sh
# Users Organizational Unit
USERS_OU="ou=users, dc=gull, dc=local"
GROUPS_OU="ou=groups, dc=gull, dc=local"
#
while read text
do
  UID=$(echo $text | cut -d \  -f 1)
  NAME=$(echo $text | cut -d \  -f 2 | sed 's/^./\u&/')
  # Affichage
  echo "dn: cn=$UID $NAME, $USERS_OU"
  echo "changetype: delete"
  echo
done < $1
echo "dn: cn=linux, $GROUPS_OU"
echo "changetype: delete"
```

... et voici une liste de noms.

```
$ cat userlist.txt
frodo baggins
samwise gamgee
meriadoc brandybuck
peregrin took
gandalf thewizard
bilbo baggins
```

Voici comment générer un fichier ldif à l'aide du script.

```
$ ./useradd_ldap.sh userlist.txt > users.ldif
```

Puis on ajoute les données.

```
$ ldapadd -x -D "cn=admin,dc=gull,dc=local" -W -f users.ldif
Enter LDAP Password:
adding new entry "cn=frodo Baggins, ou=users, dc=gull, dc=local"

adding new entry "cn=samwise Gamgee, ou=users, dc=gull, dc=local"

adding new entry "cn=meriadoc Brandybuck, ou=users, dc=gull, dc=local"

adding new entry "cn=peregrin Took, ou=users, dc=gull, dc=local"

adding new entry "cn=gandalf Thewizard, ou=users, dc=gull, dc=local"

adding new entry "cn=bilbo Baggins, ou=users, dc=gull, dc=local"

adding new entry "cn=linux, ou=groups, dc=gull, dc=local"
```

Contrôle de l'installation

Pour tester, on peut le faire en local ...

```
$ ldapsearch -x -b "dc=gull, dc=local" "objectclass=*"
```

```
$ ldapsearch -x -b "dc=gull, dc=local" "objectclass=inetorgperson"
$ ldapsearch -x -b "dc=gull, dc=local" "ou=users"
$ ldapsearch -x -b "dc=gull, dc=local" "dc=*local"
$ ldapsearch -x -b "dc=gull, dc=local" "cn=*"
```

... mais aussi de loin à l'aide du switch -H ...

```
$ ldapsearch -H ldap://<ip_ldap_srv> -x -b "dc=gull, dc=local" "objectclass=*"
$ ldapsearch -H ldap://<ip_ldap_srv> -xLLL -b "dc=gull, dc=local" uid=frodo sn
givenName cn
```

Pour obtenir l'arbre entier, on peut utiliser la commande slapcat

```
$ sudo slapcat
```

Configuration du client Linux

Sur le client, il faut ajouter les paquets suivants :

- auth-client-config
- libnss-ldap

```
$ sudo apt-get install auth-client-config libnss-ldap
```

Le paquet auth-client-config facilite la gestion de l'authentification.

```
$ sudo auth-client-config -a -p ldap_example
```

La commande ne fait que de modifier les modules pam d'authenfication.

On corrige /etc/ldap.conf

nss_base_passwd ou=users,dc=gull,dc=local?one

nss_base_shadow ou=users,dc=gull,dc=local?one

nss_base_group ou=groups,dc=gull,dc=local?one

On corrige juste un peu... cette ligne permet la création automatique des répertoires utilisateurs.

```
$ sudo vim /etc/pam.d/common-session
...
session    required     pam_limits.so
session    required     pam_unix.so
session    required     pam_mkhomedir.so skel=/etc/skel/
session    optional     pam_ldap.so
```

On peut alors voir le résultat à l'aide de

```
$ getent passwd
```

La liste doit contenir les enregistrements LDAP

Kerberos

Système d'authentification réseau qui permet une vérification unique de connexion.

```
sécurité
signature unique
serveur de confiance
authentification mutuelle
```

Constituant de base des systèmes Windows 2000 ADS + Windows 2003 ADS, il a été classé secret défense et l'exportation de certains algorithmes est sous contrôle des autorités américaines.

Aussi, il existe deux branches distinctes, Kerberos du MIT, et la version européenne, développée par l'Université de Heimdal. La deuxième version provient du travail de réécriture des fonctions interdites d'exportation.

On comprend mieux maintenant pourquoi certaines distributions (Debian par exemple) possèdent des branches US et non-US.

Notions comlémentaires

```
LDAP, Lightweight Directory Access Protocol
SASL, Simple Authentification and Security Layer
GSSAPI, Generic Security Services API
PAM, sert de couche d'abstraction d'authentification au niveau des
applicatifs.
NSS, Name Service Switch
Principal, une entrée dans la base de Kerberos
Distinguished Name, une entrée dans la base LDAP
```

Principes

KDC

Le principe d'encryption des mots de passe est basée sur une partie salt. C'est à dire une partie que l'on ajoute au nom d'utilisateur pour crypter le mot-de-passe. Dans Kerberos on utilise le nom de domaine.

Une authentification ne nécessite pas d'identifier le client au préalable, on envoie un TGT (Ticket Granting Ticket) crypté avec le mot-de-passe du client et seul sa clé permet de le décrypter.

On peut faire une demande pour un service précis à l'aide de son tiquet. Un autre tiquet sert alors d'autorisation.

Les tiquets sont des données encryptées qui ont pour buts de confirmer l'identité échanger une clé volatile de cryptage

Ils contiennent les informations suivantes

```
nom de l'utilisateur
nom du serveur principal
date de validité
liste d'adresses IP valides
clé secrète de cryptage
```

Ils ont une durée limitée, en général entre 10 et 24 heures. Il existe un répertoire de cache des tiquets.

Installation

```
$ sudo apt-get install krb5-kdc krb5-admin-server
```

Configuration

On modifie le fichier /etc/krb5.conf... voici une version épurée de tous commentaires.

```
# cat /etc/krb5.conf
[libdefaults]
      default_realm = GULL.LOCAL
[realms]
      GULL.LOCAL = {
            kdc = kdc.gull.local
            admin_server = kdc.gull.local
      }
[domain_realm]
      .gull.local = GULL.LOCAL
      gull.local = GULL.LOCAL
[login]
      krb4_convert = true
      krb4_get_tickets = false
```

Initialisation

```
# kdb5_util create -s
Loading random data
Initializing database '/var/lib/krb5kdc/principal' for realm 'GULL.LOCAL',
master key name 'K/M@GULL.LOCAL'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
```

Gestion des principals

```
# kadmin.local
Authenticating as principal root/admin@GULL.LOCAL with password.
kadmin.local:  list_principals
K/M@GULL.LOCAL
kadmin/admin@GULL.LOCAL
kadmin/changepw@GULL.LOCAL
kadmin/localhost@GULL.LOCAL
krbtgt/GULL.LOCAL@GULL.LOCAL
```

Ajout d'un principal

```
kadmin.local:  addprinc frodo/admin
WARNING: no policy specified for frodo/admin@GULL.LOCAL; defaulting to no
policy
Enter password for principal "frodo/admin@GULL.LOCAL":
Re-enter password for principal "frodo/admin@GULL.LOCAL":
Principal "frodo/admin@GULL.LOCAL" created.
```

Pour finir la session...

```
kadmin.local:  quit
```

Démarrage du service

```
# /etc/init.d/krb5-kdc start
```

Obtenir un ticket

```
# kinit frodo/admin
Password for frodo/admin@GULL.LOCAL:
# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: frodo/admin@GULL.LOCAL

Valid starting       Expires              Service principal
04. 03. 13 06:54:14  04. 03. 13 16:54:14  krbtgt/GULL.LOCAL@GULL.LOCAL
      renew until 05. 03. 13 06:54:10
```

Modifier l'ACL

```
# nano /etc/krb5kdc/kadm5.acl
frodo/admin@GULL.LOCAL *
```

On redémarre le serveur...

```
# /etc/init.d/krb5-admin-server restart
```

Pour obtenir la liste des commandes de kadmin.

```
# kadmin
Authenticating as principal frodo/admin@GULL.LOCAL with password.
Password for frodo/admin@GULL.LOCAL:
kadmin:  ?
Available kadmin requests:

add_principal, addprinc, ank
                         Add principal
delete_principal, delprinc
                         Delete principal
modify_principal, modprinc
                         Modify principal
rename_principal, renprinc
                         Rename principal
change_password, cpw     Change password
get_principal, getprinc  Get principal
list_principals, listprincs, get_principals, getprincs
                         List principals
add_policy, addpol       Add policy
modify_policy, modpol    Modify policy
delete_policy, delpol    Delete policy
get_policy, getpol       Get policy
list_policies, listpols, get_policies, getpols
                         List policies
get_privs, getprivs      Get privileges
ktadd, xst               Add entry(s) to a keytab
ktremove, ktrem          Remove entry(s) from a keytab
lock                     Lock database exclusively (use with extreme caution!)
```

```
… escamotage d'une ligne
purgekeys                Purge previously retained old keys from a principal
get_strings, getstrs     Show string attributes on a principal
set_string, setstr       Set a string attribute on a principal
del_string, delstr       Delete a string attribute on a principal
list_requests, lr, ?     List available requests.
quit, exit, q            Exit program.
```