LDAP Documentation

By: Kotari Vijay (B2004035)
C. Koundinya (B2004021)

Contents

I. Basics

Lightweight Directory Access Protocol or LDAP, is a networking protocol which is used for checking the level of directory access a user has. It is used for querying and modifying entries running over TCP/IP. In other words, it decides the information that should be made available to a user.

It is meant to provide a single means of storing authentication information and for storing other essential information about the users. It mostly used for authentication purposes but it can also be used for storing data about the users themselves such their names, addresses, email addresses, etc.

A very simple LDAP implementation would involve UserID's and passwords' of the users being stored so that when they log in, the UserID and password provided could be used to verify the identity of the user.

In the OSI model, the X.500 directory services were meant to be accessible via the X.500 Directory Access Protocol.
An LDAP database structure usually follows the X.500 model: it is a tree of entries, each of which consists of a set of named attributes with values.

An LDAP directory often reflects various political, geographic, and/or organizational boundaries, depending on the model chosen. LDAP deployments today tend to use Domain Name System (DNS) names for structuring the most simple levels of the hierarchy. Further into the directory might appear entries representing people, organizational units, printers, documents,

groups of people or anything else which represents a given tree entry, or multiple entries.

II. Installation

A). OpenLDAP

OpenLDAP Software is an open source implementation of the **L**ightweight **D**irectory **A**ccess **P**rotocol. OpenLDAP Software can be downloaded from

http://www.openldap.org/software/download/

B). phpLDAPadmin

phpLDAPadmin is a graphical interface that is commonly used with the Openldap software, to easily add, remove or edit the entries. However, it is not very useful for dealing with a large number of entities.

III. Configuration

A). OpenLDAP

Configuring openLDAP begins and ends with its lone configuration file slapd.conf

```
include     /etc/openldap/schema/core.schema
include     /etc/openldap/schema/cosine.schema
include     /etc/openldap/schema/new/ibm-auxAccount.schema
include     /etc/openldap/schema/inetorgperson.schema
include     /etc/openldap/schema/yast.schema
include     /etc/openldap/schema/nis.schema
include     /etc/openldap/schema/new/qmail.schema
include     /etc/openldap/schema/new/ml.schema
```

First we include all the schemas that we will need for our entities. This largely depends on the kind of attributes that we wish to store about our users. For example, the qmail.schema is used to store information specific about mail like email addresses of the person, the storage area of the mail of the user.

pidfile /var/run/slapd/slapd.pid

The pidfile is used to contain the pid of the currently running slapd.

argsfile /var/run/slapd/slapd.args

The argsfile is used to store the services to start when slapd is started. For example to start only the ldap service, use

./slapd – h ldap:///

To run ldaps(Secure LDAP) as well append ldaps:/// to the above.

modulepath /usr/lib/openldap/modules

To add any new modules to the existing LDAP, this is the directory where the modules have to be stored.

```
access to dn.base=""
        by * read

access to dn.base="cn=Subschema"
        by * read

access to attr=userPassword,userPKCS12
        by self write
        by * auth

access to
attr=shadowLastChange,telephoneNumber,postalAddress,homePostalAddre
```

ss,mobile,roomNumber,homePhone,deliveryMode,mailForwardingAddress, mailReplyText
by self write
*by * read*

*access to **
*by * read*

The Access Control List above enlists the access to the attributes of the users. By default, the Administrator has read and write access to all the attributes and entities.

The ACL are of the form
Access to attribute1
By who what

Attribute1 is the attribute to be controlled.
Who denotes the person who is being controlled, self is the person to whom the attribute belongs, what denotes the rights enabled.

IV. LDAP SERVICES AT IIIT-A

Object Tree Structure:

Data is represented in an LDAP enabled directory as a hierarchy of objects, each of which is called an entry. The resulting tree structure is called a Data Information Tree (DIT). The top of the tree is commonly called the **root** (a.k.a **base** or the **suffix**).

Each **entry** in the tree has one parent entry (object) and one or more child entries (objects). Each child entry (object) is a sibling of its parent's other child entries.

Each entry is composed of (is an instance of) one or more objectClasses. **Objectclasses** contain zero or more attributes. Attributes have names (and sometimes abbreviations or aliases) and typically contain data .

The DIT structure of IIIT-A LDAP is shown in the fig 1.1

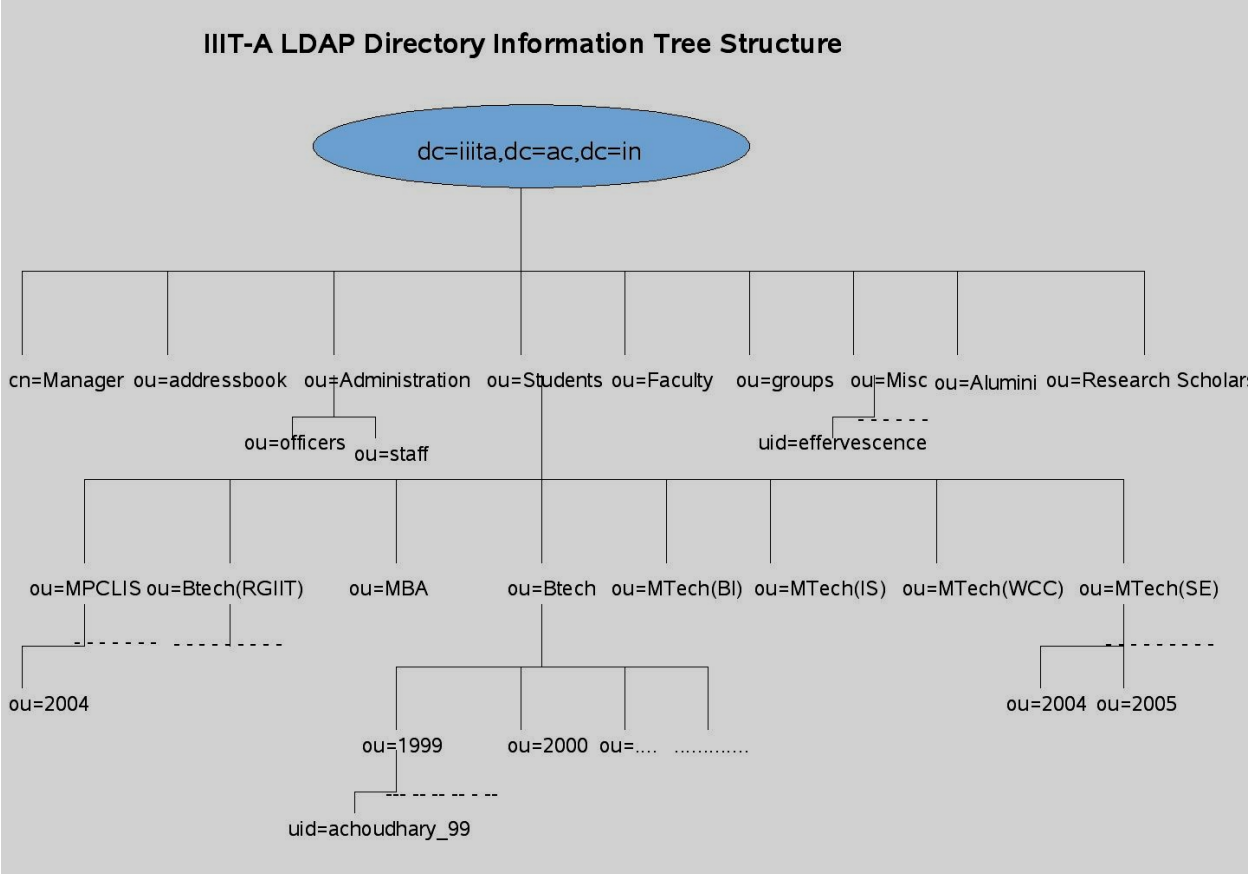

fig 1.1

The root (base or suffix) is dc=iiita, dc=ac, dc=in. The root has child entries like ou=Students, ou=Faculty which contain one or more object Classes like top, OrganisationalUnit. At each level in the hierarchy one or more of the attributes must contain data that somewhat uniquely identifies each entry.

By constructing paths that comprise these named attributes we can get to our desired entry or search start position.

These paths are quaintly called Distinguished Names (DN). Each unique data attribute that is a part of this DN is called a Relatively Distinguished Name (RDN). For example in the fig 1.1, in order to get the entry with uid=achoudary_99, the Distinguished Name is dn:uid=achoudary_99,ou=1999,ou=Btech,ou=Students,dc=iiita,dc=ac,dc=in. And the RDN is uid=achoudary_99.

LDAP is used for authentication by services like MAIL, PROXY, MYIIITA, FORUMS, People's SERVER, PROFILE SERVER.