In this post let us look at LDAP, what it means and where it is used.

What is LDAP? LDAP stands for Lightweight Directory Access Protocol. This was initially used to refer to as Network Access **Protocol**. But as it became more popular and mature, it was more or less referred now to the directory architecture itself, instead of the protocol alone.

LDAP is now standardized. The standard includes the network protocols, the directory structure, and the services provided by the LDAP server. These are available as RFCs. In this post, we shall look into some basics and frequently asked newbie questions in LDAP. Let's begin.

What is an LDAP directory?

Think of a phone directory where you can find out the phone number of a given name. LDAP directory is similar in concept but not bound to a particular type of record, such as phone records. LDAP directories are generic in which you may opt to define a directory for phone numbers as well as a directory for users (and their attributes) in an organization. Literally, you can define any entity as a directory.

What is a directory server?

A directory server is an implementation that supports a centralized repository to store and manage information of such directories. OpenLDAP, Microsoft Active Directory (AD), and Apache Directory Server are few examples.

While working with LDAP directories, you would come across some common jargon. Let's understand some of the common keywords.

Unique Name

In a directory, there has to be a unique name for an entity to distinguish between two entries. Say for example, in a phone directory if you are looking for a name, John Carlson, and if you have two entries of a similar name, you can't distinguish between the two without additional information. Thus, in directory servers, there has to be a mechanism to uniquely identify an entity. LDAP adopts this strategy by defining a Distinguish Name (DN) for each entity. A directory designer should design what constitutes a DN. For example, the following may be a DN.

e.g: dn: o=Silicon Technologies, l=MountainView, st=California, c=US

DN's are not case sensitive

Relative Distinguish Name (RDN)

RDN is a unique name that identifies an entry within a container. DN is the grouping of RDN of the object entries to the directory root entry with all the parent container entity RDNs together. You may think of the RDN as the file names in filesystem directories and DN as the absolute pathname of it.

LDAP entry

An LDAP entry is a record in the directory server. It could be a record of any object type. Taking the above example, a record may be as follows.

```
dn: o=Silicon Technologies, l=San Francisco, st=California, c=US
o: Silicon Technologies
postalAddress: 323B Jo Street, San Francisco
l: San Francisco
st: California
postalCode: 61417-7734
c: US
telephoneNumber: +1 877 767 1234
telephoneNumber: +1 877 434 3456
objectclass: organization
```

Here the first line is the DN and others are attributes. Attribute names like telephoneNumber and st are referred to in a schema. It's basically a model that describes what an entry could have in terms of attributes and other details. That can't be changed as we need. Because everyone has to follow a common format.

Some important tips of attributes

- Attribute names are case insensitive — e.g: both ‘o’ and ‘O’ could mean the organization
- Attribute value may or may not be case sensitive. It depends on the attribute
- Attribute names can have synonyms. e.g c and countryName could be synonymous attribute names

ObjectClass attribute

This is a special attribute that must be present in a record. It defines the type of the record. Given an objectclass, determines what are the attributes it supports. It defines what attributes must present, can present, and can’t present. What attributes and corresponding values can be stated is determined by object class definition and the schema definition. There are three main types of objectClasses; STRUCTURAL (e.g: person, OrganizationalUnit), AUXILIARY (e.g: certificationAuthority) and ABSTRACT (e.g: top)

Directory Tree

A directory in reality is very synonymous with a file system structure, such as a tree. A parent element may have multiple child elements. Typically a child only has a single parent. In LDAP the entities are stored in a tree-like structure. Each element has a DN, to uniquely distinguish it from others.

DSE (DSA-Specific Entry)

DSE is the root or top-level entry in an LDAP directory. (DSA stands for Directory System Agent)

LDAP Interchange Format (LDIF)

LDIF is a standard text file format of LDAP configurations and directory contents. It can contain a collection of entries separated by a blank line. It also contains mostly attribute names and corresponding values. LDIF is a common way to add new data or modify existing data into a directory tree and it should be subjected to the schema of the entry. An example LDIF content of the topmost entry in the tree.

```
# LDIF listing for the entry dn: dc=example,dc=org
# Whatever on left to colon (:) is an ttribute name.
# Note there's a space after colon and the rest is the attribute value
dn: dc=example,dc=org
objectClass: domain
dc: example
```

Following are a few vocabulary that's specifically used in Microsoft Active Directory, but still conveys some important information in general. Microsoft AD is simply more than an LDAP.

Active Directory Domain

An Active Directory domain (which is a logical grouping of objects and containers) is made up of the following components:

- An X.500-based hierarchical structure of containers and objects
- A DNS domain name as a unique identifier
- A security service, which authenticates and authorizes any access to resources via accounts in the domain or trusts with other domains
- Policies that dictate how functionality is restricted for users or machines within that domain

Domain controller (DC)

DC can be authoritative for one and only one domain. It is not possible to host multiple domains on a single DC.

Domain Tree

A domain tree (which you can map with the directory tree that we stated above) is a collection of domains

Forests

Forests are a collection of domain trees that share a common schema and configuration. A forest is named after the first domain that is created, also known as the forest root domain.

Under a domain, you may have containers (that may house more containers and objects) or objects (which are not

containers). The primary type of container that you will create to house objects is called an organizational unit (OU). Another type of container, which is actually called a container, can also be used to store a hierarchy of objects and containers. Although both can contain huge hierarchies of containers and objects, an organizational unit can have group policies applied to it.

Well, those are the words that you need to know to learn deep LDAP related information. I'll conclude this introductory post by comparing LDAP's to databases.

How does LDAP differ from a database?

In fact, LDAP is just a repository of different entities, which you can define in a database as well. In this point of view, LDAP is nothing more than a special type of database that organizes data into tree structures, like a file system, instead of a table structure as in an RDBMS. But one attribute that distinguishes an LDAP from databases is that LDAP's are more read optimized than DBs. DBs are supposed to be very good at reading as well as writes. Features like transaction write locks etc. are not needed for an LDAP. One thing to keep in mind is where LDAP stores the data. For your information, it could be an RDBMS database. Note, one part of LDAP is the protocol. And it doesn't strictly impose any restriction on the data store. LDAP implementations are free to choose whatever deem fit for them.

Well. I think I just shared few pieces of information here. I'll meet you with another post that talks about configuring a simple LDAP hierarchy up and running in your local Linux machine.