Lightweight Directory Access Protocol

26th September 2018

IAM Workshop

LEARN
Overview

- Introduction to LDAP
- LDAP Protocol overview
- Directory structure
- Operations
  - Add
  - Bind
  - Delete
  - Search and compare
  - Modify
- Schema
Lightweight Directory Access Protocol

• Introduction

• Open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network

• directory services may provide any organized set of records, often with a hierarchical structure
  
  – *eg. telephone directory is a list of subscribers with an address and a phone number*

• LDAP is specified in a series of Internet Engineering Task Force (IETF) Standard Track publications called Request for Comments (RFCs), using the description language ASN.1. The latest specification is Version 3, published as RFC 4511.

• A common use of LDAP is to provide a central place to store usernames and passwords.

• allows many different applications and services to connect to the LDAP server to validate users.

• platform-independent protocol
Lightweight Directory Access Protocol

• Protocol Overview

• Directory System Agent (DSA)
  - A client starts an LDAP session by connecting to an LDAP server
  - by default on TCP and UDP port 389
  - The client then sends an operation request to the server, and the server sends responses in return
    • StartTLS — use the LDAPv3 Transport Layer Security (TLS) extension for a secure connection
    • Bind — authenticate and specify LDAP protocol version
    • Search — search for and/or retrieve directory entries
    • Compare — test if a named entry contains a given attribute value
    • Add a new entry
    • Delete an entry
    • Modify an entry
    • Modify Distinguished Name (DN) — move or rename an entry
    • Abandon — abort a previous request
    • Extended Operation — generic operation used to define other operations
    • Unbind — close the connection (not the inverse of Bind)
Lightweight Directory Access Protocol

• Directory Structure
  • usually structured hierarchically as a tree of nodes
  • the LDAP directory tree is sometimes referred to as the Directory Information Tree, or DIT
  • Each node represents a record, or “entry” in the LDAP database

• The Distinguished Name (DN)
  − An LDAP entry consists of numerous attribute-value pairs
  − uniquely identified by what is known as a “distinguished name” (DN)
  • eg.
    - dn: mail=joe@novell.com, dc=novell, dc=com
    - objectclass: inetOrgPerson
    - cn: Joe
    - sn: Somebody
    - mail: joe@novell.com
    - telephoneNumber: 1 234 567 8912
**Open LDAP**

- **What is Open LDAP**
  - free, open source implementation of the Lightweight Directory Access Protocol (LDAP)
  - BSD-style license called the OpenLDAP Public License
  - developed by the OpenLDAP Project
  - OpenLDAP has three main components
    - slapd – stand-alone LDAP daemon and associated modules and tools
    - libraries implementing the LDAP protocol and Basic Encoding Rules (BER)
    - client software: ldapsearch, ldapadd, ldapdelete, and others
Open LDAP

- Open LDAP using OLC (cn=config)
  - On-Line configuration for previous slapd.conf
  - Dynamic configuration of static configuration in slapd.conf where slapd restart needed
  - Configuration may be perform run time using a DIT cn=config
  - Zero down time configuration
  - Stored in /etc/ldap/slapd.d directory
  - Introduce in version 2.3
LDAP Schema

- **Schema**
  - A set of rules that define what can be stored as entries in an LDAP directory
  - Each LDAP directory has a default schema
  - The elements of a schema
    - Attributes, syntaxes, object classes

- **Attributes**
  - Defines a piece of information that directory entries contain
    - For example, some common attributes for entries related to people are cn (common name), telephoneNumber, and userPassword.

- **Syntaxes**
  - Defines the data format in which an attribute value is stored.
    - Directory String, Integer, and JPEG are examples of standard LDAP syntaxes.
LDAP Schema

- **Object Classes**
  - defines a set of attributes for a type of directory entry
  - two or more object classes in an object class hierarchy define the attributes for a type of entry
  - An object class inherits attributes from all parent object classes in the hierarchy and then adds attributes of its own
    - for example:
      Object class 1: adds attribute A
      Object class 2: inherits attribute A and adds attributes B, C, and D
      Object class 3: inherits attributes A, B, C, and D, and adds attributes E and F
  - There are three types of object classes: abstract, structural, and auxiliary
LDAP Schema Example

objectclass ( 2.5.6.6 NAME 'person' DESC 'RFC2256: a person' SUP top STRUCTURAL
    MUST ( sn $ cn )
    MAY ( userPassword $ telephoneNumber $ seeAlso $ description ) )

attributetype ( 2.5.4.4 NAME ( 'sn' 'surname' )
    DESC 'RFC2256: last (family) name(s) for which the entity is known by' SUP name )

attributetype ( 2.5.4.4 NAME ( 'cn' 'commonName' )
    DESC 'RFC4519: common name(s) for which the entity is known by' SUP name )

...