JNDI and LDAP – Part II

Noel J. Bergman
DevTech®
Session Overview

LDAP is an IETF-standardized core technology that you can use to integrate information across multiple enterprises, computing platforms, programming languages, and applications. We will discuss why and when to use a Directory Services model rather than a Relational Database model; look at published schema that standardize critical concepts; and examine how LDAP can be used to implement such things as centralized identity across operating systems, Web applications, mail servers, etc. We will also discuss fine-grained access control, persistent searches, and other advanced topics that highlight LDAP's advantages.

PLEASE ASK QUESTIONS! 😊
Session Prerequisites

- You will want a good understanding of Java.
- You will want an understanding of JNDI and some basic LDAP (see Part I).
- To run most of the sample code, you’ll need to install a suitable LDAP server.
## Directory vs. RDB

### Directory
- Relatively simple data model, optimized to be highly read intensive & searchable.
- Hierarchical with schema inheritance.
- Federated.
- Fine-grained, flexible, security.

### Relational Database
- Larger, more complex, relational data model, optimized for lots of transactions.
- Relational tables and queries.
- Centralized.
- Relational security.
Directory Standards Benefits

- Centralized information management of distributed, federated, multi-purposed data.
- Compared to application-specific databases, reduces:
  - Inconsistent data
  - Data redundancy
  - Administration
LDAP Benefits

- Common, simple, fast, implementation of X.500 concepts over TCP/IP stack.
- Well-defined language bindings for many languages, wire-level protocol, and most needed schema.
- Extensible, both in terms of new schema and new directory operations.
Common Uses for LDAP

- Identity
  - Authentication
  - Authorization
  - Profile

- Administration
  - Accounts
  - Servers
  - Services
LDAP Aspects

- Information
  - Named *entries* composed of *object classes*, which contain named *attributes*, both of whose presence and makeup is defined by *schema*.

- Structure
  - Hierarchical structure defined by a federated namespace.

- Operations
  - Authentication: bind (to server), unbind, abandon
  - Query: search and compare
  - Update: add, delete or modify entry; modify RDN
  - Controls (modify operations) and Extended (new) Operations

- Security
Schema Elements

- Attributes – named values. Must be defined by schema before they’re used in an object class.

Object Classes

- Structural – used to determine the entry’s location within the DIT, and group related attributes.
- Auxiliary – group related attributes. Can be attached to entries irrespective of their place in the DIT.
  - extensibleObject is a special AUXILIARY object class that permits *any* attribute defined by a currently loaded schema to be used.
- Abstract – cannot directly instantiate. See top.
Sample User

- **dn:**
  - uid=noel,ou=People,dc=apache,dc=org
- **objectClass:**
  - top
  - account
  -posixAccount
  -shadowAccount
  - qmailUser
- **uid:** noel
- **host:**
  - ajax
  - eris
  - hermes
  - loki
  - minotaur
  - nagoya
- **mailForwardingAddress:**
  - noel@devtech.com
- **cn:** Noel J. Bergman
- **uidNumber:** 711
- **gidNumber:** 711
- **homeDirectory:** /home/noel
- **loginShell:** /bin/bash
- **userPassword:** {CRYPT}...
- **mail:** noel@apache.org
Sample User Structure

- objectclass (2.5.6.0 NAME 'top' ABSTRACT MUST objectClass)
- objectclass (0.9.2342.19200300.100.4.5 NAME 'account' SUP top STRUCTURAL MUST userid
  MAY (description $ seeAlso $ localityName $ organizationalName $ organizationalUnitName $ host))
- objectclass (1.3.6.1.1.1.2.0 NAME 'posixAccount' SUP top AUXILIARY
  MUST (cn $ uid $ uidNumber $ gidNumber $ homeDirectory)
  MAY (userPassword $ loginShell $ gecos $ description))
- objectclass (1.3.6.1.1.1.2.1 NAME 'shadowAccount' SUP top AUXILIARY
  MUST uid
  MAY (userPassword $ shadowLastChange $ shadowMin $ shadowMax $ shadowWarning $ shadowInactive $ shadowExpire $ shadowFlag $ description))
- objectclass (1.3.6.1.4.1.7914.1.2.2.1 NAME 'qmailUser' SUP top AUXILIARY
  MUST (mail $ uid)
  MAY (mailMessageStore $ homeDirectory $ userPassword $ mailAlternateAddress $ qmailUID $ qmailGID $ mailQuota $ mailHost $ mailForwardingAddress $ deliveryProgramPath $ qmailDotMode $ deliveryMode $ mailReplyText $ accountStatus)
What Happened to userid?!

- Wait! I thought that userid was a MUST for the account class!

- Yes, it is. But look at the attribute definition for userid:

```plaintext
attributetype ( 0.9.2342.19200300.100.1.1
    NAME ( 'uid' 'userid' )
    DESC 'RFC1274: user identifier'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256})
```

- The attribute has two names. 😊
POSIX Group Structure

- RFC 2307 NIS Schema class for a POSIX (UNIX) Group
  - objectclass ( 1.3.6.1.1.1.2.2 NAME 'posixGroup'
    SUP top STRUCTURAL
    MUST ( cn $ gidNumber )
    MAY ( userPassword $ memberUid $ description ))

- Sample Attributes:
  - attributetype ( 1.3.6.1.1.1.1.1 NAME 'gidNumber'
    EQUALITY integerMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
  - attributetype ( 1.3.6.1.1.1.1.12 NAME 'memberUid'
    EQUALITY caseExactIA5Match
    SUBSTR caseExactIA5SubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
Sample POSIX Group

- Sample Group:
  - dn: cn=mailAdmins,ou=Group,dc=example,dc=org
  - objectClass: top
    posixGroup
  - cn: mailAdmins
  - gidNumber: 117
  - memberUid: aaron
    berin
    erik
    justin
    noel
  - Description: Mail Administrators

- To find a user's groups, search with a filter (see RFC 2307 #5.2):
  \((&(objectClass=posixGroup)(memberUid=\%s)))\)
LDAP Security

- RFC 2820 specifies access control requirements for LDAP, but not a syntax.
- LDAP uses fine-grained access control.
- You could spend hours exploring OpenLDAP ACLs:
  - [http://sapiens.wustl.edu/~sysmain/info/openldap/openldap_configure_acl.html](http://sapiens.wustl.edu/~sysmain/info/openldap/openldap_configure_acl.html)

  Simple example: allow users or administrators to change their password, unauthenticated users to authenticate, and deny read except to the owner or administrators.
  
  - access to attr=userPassword
  - by self write
  - by group="cn=admins,ou=Groups,dc=apache,dc=org" write
  - by anonymous auth
  - by * none
LDAP Integration

- Two basic approaches can be taken to integrate with LDAP.
  
  - Direct – the operation to be integrated is tied directly into LDAP, relying upon it in real-time. This is the most common approach.
  
  - Indirect – the operation to be integrated uses its “native” data stores, which are populated from LDAP by helper tools.
    - The debian project’s “userdir-ldap” tools exemplify the indirect approach.

- These are not mutually exclusive approaches within an overall deployment.
LDAP Integration Compared

**Direct**
- LDAP is accessed directly in real-time.
- **Pros:**
  - Real-time consistency.
  - Leverages LDAP’s fine-grained access control.
- **Cons:**
  - Requires real-time LDAP access.
  - Requires code to be modified to use LDAP.
  - Operational reliability tied to quality of LDAP integration code and availability of LDAP.

**Indirect**
- Native data stores are accessed in real-time, and updated from LDAP.
  - Typically a periodic task.
  - Could use triggers (few LDAP servers support them).
- **Pros:**
  - Does not require change to functional code.
  - Works even when LDAP server is unavailable.
- **Cons:**
  - Native stores can lag LDAP.
  - Awkward, at best, to use LDAP access control when populating native stores.
**nss_ldap**

- Ties the Nameservice Switch directly to LDAP.
  - IETF RFC 2307
  - Operating systems that use NSS become LDAP enabled.

- Originally developed by PADL Software. You may want to use current PADL code, but the same caveats apply as for pam_ldap.
Using `nss_ldap`

- **Edit** `/etc/nsswitch.conf`, *e.g.*:
  - passwd: files ldap
  - shadow: files ldap
  - group: files ldap

  - Local entries will be used first, *e.g.*, `root`, followed by entries in LDAP.

- **Edit** `/etc/ldap.conf`, *e.g.*:
  - host `ldap.apache.org`
  - base `dc=apache,dc=org`

- See RFC 2307 for LDAP details and examples.
pam_ldap

- Ties PAM (Pluggable Authentication Module) directly to LDAP.
  - All software that uses PAM becomes LDAP enabled.
- PAM is much more flexible than NSS, and allows us to add new authentication methods, such as OPIE.
  - See http://ldappubkey.gcu-squad.org/ to integrate OpenSSH Public Key authentication with LDAP.
- Usually derived from PADL Software’s code. Could use that code, or use the version that comes with your PAM distribution.
  - Frequently updated. Check their change logs. Test for memory leaks or other issues.
JAAS

- The Java Authentication and Authorization Service (JAAS) provides a PAM-like package to authenticate and authorize users in a Java environment.
- JAAS is intended to permit Java code to know who is executing it, regardless of whether the code is an application, applet, EJB, servlet, portlet, ... 
- JAAS integrates Principal-based access controls into the Java Security model.
- Can be backed by JNDI/LDAP by using the correct JAAS PAM plug-in module.
  - See:  http://www.theserverside.com/articles/article.tss?l=Pramati-JAAS
  - See also: “Security Foundations of Java” with Simon Roberts.
Debian manages many disparate systems.

- [http://db.debian.org/machines.cgi](http://db.debian.org/machines.cgi)
- Neither all co-located, nor always available.

Per-user attributes include:

- ssh authentication keys
- vacation notices
- xplanet coordinates

How to manage users, groups and other info?

Their purpose-built userdir-ldap tools use the indirect approach to provide centralized account management without code change.

Updates performed via LDAP, Web Forms and e-mail.

- Why won’t using passwd or changing a .forward file work?
**Triggers**

- LDAP Triggers are currently non-standard.
  - Bell Labs LTAP: [http://ltap.bell-labs.com](http://ltap.bell-labs.com)
  - The Apache Directory Server (Eve) will also have triggers.

- Pre- and post-operation interceptors.
- Can use with the indirect approach to update native stores.
- Can be used to enforce data validity and consistency.
- Can be used to integrate data across data models.
  - Operation on LDAP automatically results in an operation on another data store, such as a relational database.
- Extended operations can be used to implement triggers.
Extended Operations

- The LDAP wire protocol is extensible, allowing new LDAP operations to be encoded as specified by RFC 2251 §4.12.
- JNDI Supports extended operations using:
  - `javax.naming.ldap.ExtendedRequest`
    - Represents the extended operation to be encoded on the request.
  - `javax.naming.ldap.ExtendedResponse`
    - Represents the response from an extended operation.
- `javax.naming.ldap` provides extended request/response objects to implement the RFC 2830 STARTTLS operation.
- An unsolicited notification is also an extended response.
- Servers are not required to support an extended operation.
Unsolicited Notification

- RFC 2251 #4.4 defines unsolicited notification, which allows a server to send an LDAP message to a client that is not a response to a client request message.
- `javax.naming.ldap.UnsolicitedNotification` is the interface to an unsolicited notification. These will be processed as events.
- `javax.naming.ldap.UnsolicitedNotificationListener` is the contract used to listen for notification events.
- The `javax.naming.event` package is used here. In specific, we use the `javax.naming.event.EventDirContext`, which (conveniently) is implemented by the context returned by the LDAP Service Provider.
The LdapContext

- `javax.naming.ldap.LdapContext`
- Extends `javax.naming.Directory.DirContext`
- Part of the `javax.naming.ldap` package, which adds RFC 2251 specific features:
  - LDAP Controls
  - Extended Operations
  - Unsolicited Notifications
  - LDAP specific helper classes, such as `Rdn`.
- Used when necessary to access certain, more advanced, LDAP capabilities.
LDAP Controls

- The LDAP protocol defines parameterized operations. Operations may also have, often optional, modifiers. This extended data is known as a *control*, and defined by RFC 2251 #4.1.12.

- Controls are represented by:
  - Control Type (unique ID)
  - Criticality (must the server support it)
  - Control Value

- LDAP Persistent Searching is implemented using a control.
Controls and HasControls

- `javax.naming.ldap.Control`
  - Specifies the basic contract for an LDAP Control.
  - Implemented by a number of `Control` subclasses in `javax.naming.ldap`.

- `javax.naming.ldap.HasControls`
  - Used to test to see if a returned object has controls, and to access them.
  - `Controls[] getControls()`
Persistent Search

- LDAP Persistent Searching is done using a control.
  - [http://www.ietf.org/proceedings/01aug/I-D/draft-ietf-ldapext-psearch-03.txt](http://www.ietf.org/proceedings/01aug/I-D/draft-ietf-ldapext-psearch-03.txt)

- A PersistentSearch control is attached to the SearchRequest message.

- The LDAP server will send SearchResultEntry messages until the client abandons the request or unbinds from the server. The server will not send a SearchResultDone message.

- The SearchResultDone message(s) may have attached EntryChangeNotification controls.
IETF LDAP Java API

- An IETF effort to produce an RFC standardized JAVA binding to LDAP.
- Sun, Novell and Netscape are the joint authors of the draft RFC.
- Sun’s LDAP Triggers paper uses this API, rather than JNDI.
- Unlike JNDI, which we used to access LDAP in part I of this presentation, this API is not a generic interface, like JNDI. It is LDAP specific, providing a much-richer, but also more complex, dialect for working with an LDAP directory.
- Novell’s implementation is available through the OpenLDAP Project: [http://www.openldap.org/jldap/](http://www.openldap.org/jldap/)
WebSphere Member Manager

- The WebSphere Member Manager (WMM) stores User (Member) information in LDAP and SQL stores.
- Enterprise-wide LDAP DIT might contain the most common user information.
- WMM database can store additional, WebSphere-specific, user information.
- WebSphere Member information can be extended without changing the corporate LDAP schema.
- WMM metadata defines what to store where.
Wrap-up

- Are there any questions?
- Please remember to turn in your speaker evaluation forms.
- Thank you for coming. I hope that you’ve enjoyed the session.
Links – JNDI

- The JNDI Specification
  http://java.sun.com/j2se/1.4.2/docs/guide/jndi/spec/jndi/

- Sun’s generally excellent JNDI Tutorial
  http://java.sun.com/products/jndi/tutorial

- JavaWorld articles on JNDI

- ONJava article on JNDI and LDAP
  http://www.onjava.com/pub/a/onjava/2001/05/21/jndi.html

- JNDI/LDAP: Guidelines for LDAP Service Providers
  http://java.sun.com/j2se/1.4.2/docs/guide/jndi/jndi-ldap-gl.html
## Links – LDAP

- **LDAP RFC Documents**
  [http://www.rfc-editor.org/cgi-bin/rfcsearch.pl?searchwords=ldap&num=100](http://www.rfc-editor.org/cgi-bin/rfcsearch.pl?searchwords=ldap&num=100)

- **Understanding X.500**

- **IBM Redbook: Understanding LDAP – Design and Implementation**

- **Demystifying the LDAP DIT**

- **Linux Magazine Series on LDAP**

- **OpenLDAP**
  [http://www.openldap.org](http://www.openldap.org)

- **Apache Directory Project**
  [http://incubator.apache.org/directory](http://incubator.apache.org/directory)
Links – LDAP Schema

- LDAP Schema RFC Documents *(nota bene: search is a bit broad)*
  [http://www.rfc-editor.org/cgi-bin/rfcsearch.pl?searchwords=schema&num=100](http://www.rfc-editor.org/cgi-bin/rfcsearch.pl?searchwords=schema&num=100)

- IBM SecureWay Schema (w/ IETF RFC defined schema)

- Debian schema
  [http://db.debian.org/userdir-ldap.schema](http://db.debian.org/userdir-ldap.schema)

- qmail LDAP Schema

- LDAP Schema for UDDI
Links – LDAP Integration

- NIS and PAM LDAP Tools
  http://www.padl.com/
- Using LDAP to Manage Unix Accounts
- Securing J2EE Applications using LDAP
  http://docs.sun.com/source/817-6087/dgsecure.html#wp23244
- Security with LDAP
  http://www.skills-1st.co.uk/papers/security-with-ldap-jan-2002/
- Kerberos and LDAP
  http://www.ofb.net/~jheiss/krbldap/kerberos_and_ldap.html
- Debian userdir-ldap tools
  http://people.debian.org/~troup/userdir-ldap.tar.gz
- OpenSSH Public Key Authentication with LDAP
  http://ldappubkey.gcu-squad.org/ or http://www.b0l.org/
- mod_user_ldap
  http://horde.net/~jwm/software/mod_ldap_userdir/
- qmail and ezmlm with LDAP
  http://www.lifewithqmail.org/ldap/
LDAP RFC List (3Q2004)

- RFC 1274 The COSINE and Internet X.500 Schema
- RFC 1804 Schema Publishing in X.500 Directory
- RFC 2079 Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)
  ... plus many more related to X.500
- RFC 2247 Using Domains in LDAP/X.500 Distinguished Names
- RFC 2254 The String Representation of LDAP Search Filters
LDAP RFC List (3Q2004)

- RFC 2255 The LDAP URL Format
- RFC 2256 A Summary of the X.500(96) User Schema for use with LDAPv3
- RFC 2293 Representing Tables and Subtrees in the X.500 Directory
- RFC 2294 Representing the O/R Address hierarchy in the X.500 Directory Information Tree
- RFC 2307 An Approach for Using LDAP as a Network Information Service
- RFC 2377 Naming Plan for Internet Directory-Enabled Applications
- RFC 2587 Internet X.509 Public Key Infrastructure LDAPv2 Schema
- RFC 2596 Use of Language Codes in LDAP
LDAP RFC List (3Q2004)

- RFC 2649 An LDAP Control and Schema for Holding Operation Signatures
- RFC 2696 LDAP Control Extension for Simple Paged Results Manipulation
- RFC 2713 Schema for Representing Java(tm) Objects in an LDAP Directory
- RFC 2714 Schema for Representing CORBA Object References in an LDAP Directory
- RFC 2798 Definition of the inetOrgPerson LDAP Object Class
- RFC 2829 Authentication Methods for LDAP
- RFC 2849 The LDAP Data Interchange Format (LDIF) - Technical Specification
LDAP RFC List (3Q2004)

- RFC 2891 LDAP Control Extension for Server Side Sorting of Search Results
- RFC 3045 Storing Vendor Information in the LDAP root DSE
- RFC 3062 LDAP Password Modify Extended Operation
- RFC 3088 OpenLDAP Root Service An experimental LDAP referral service
- RFC 3112 LDAP Authentication Password Schema
- RFC 3377 LDAP(v3): Technical Specification
- RFC 3383 Internet Assigned Numbers Authority (IANA) considerations for the Lightweight Directory Access
- RFC 3384 LDAP v3 Replication Requirements
Related Sessions

- “Federated Identity Management, the Real Story” (Anthony “Dr. Secure” Nadalin)
- “Security Foundations of Java” (Simon Roberts)
- JNDI can be used as an interface to UDDI, as well, and has some overlap with JAXR (Java API for XML Registries).

➤ See the IETF Draft Schema for UDDI in LDAP:
  http://ietfreport.isoc.org/idref/draft-bergeson-uddi-ldap-schema/