Deploying OpenLDAP

John H Terpstra, CTO
PrimaStasys Inc.

jht@primastasys.com & jht@samba.org
Agenda

- Understand basics of OpenLDAP
- Discover how to deploy it quickly
- Look at ways LDAP can be used
- Overview LDAP utilities
- Know what LDAP servers are in common use
Assumptions

- Use commercial Linux distribution that includes OpenLDAP
  - We will NOT compile and build OpenLDAP
- You want to get OpenLDAP running quickly
  - Before understanding all the technical details
  - Learn by experimentation
    - Modify working system
    - See what breaks
- When you have learned enough
  - Start with a fresh, new installation
- Expect to start from scratch for final solution
BASICS
Basics

- What is LDAP?
- What is OpenLDAP?
- How can one get started quickly?

Assumptions:
- It is easier to comprehend what is happening if you can see an example in operation
- Experience accelerated learning through observation
- It is easier to modify an working system than to build it from a cold start
LDAP Defined

- Lightweight Directory Access Protocol
  - Lightweight protocol for accessing directory service
    - X.500-based
    - Runs over connection oriented network protocols
      - TCP/IP
      - Defined in RFC2251
      - Technical specification RFC3377
Directory Definition

- What is the difference between a directory and a database?
  - A directory is a specialized database optimized for reading, browsing and searching
  - A directory has database and index files
    - Optimized for rapid information retrieval
    - Is not optimized for transaction oriented work, has no roll-back ability
    - Information in a directory is generally of a descriptive nature
- The Internet Domain Name System is an example of a directory
Use of directories

- Identity management:
  Note: Single-Sign-On requires additional services / utilities
  - Unified back-end for authentication data
    - Mailing system user and list management
    - FTP Server user access control
    - Samba password backend
    - Replacement for NIS/YP database
    - Web access control backend
- Back-end for DNS and DHCP information
- Much more ...
Directory Terms

- **Distinguished Name (DN)**
  
  jht@example.com becomes:
  
  $$dn = uid=jht,dc=example,dc=com$$

- **Relative Distinguished Name (RDN)**
  
  $$uid=jht$$ is an example of an RDN

- **Common Name (CN)**
  
  jht@example.com can also be expressed as:
  
  $$dn = cn=jht,dc=example,dc=com$$

  Could also just contain the User's common name, eg:
  
  $$cn = John H Terpstra$$

- **Many more - discussed later**
Directory Arrangement

- A directory should be arranged to suit the purpose for which information is required
  - Consider:
    - What type of information will be retrieved most often
    - Performance requirements
    - Future use
    - Expansion / contraction possibilities
    - Filters that may be needed to locate information
  - Must be optimised
  - Must be appropriately indexed
Example: Organizational Directory
Example: DNS Type Directory

The Organization

Organizational Unit

Person

uid=babs

ou=People

dc=example

dc=com

dc=net

dc=DE
Directory Organization

- **Singel Master**
  - Local Directory

- **Multi-Master**
  - Local Directory with Referrals
  - Replicated Directory Services
  - Distributed Directory Service

- **Multiple Directories may involve:**
  - Superior directories
  - Subordinate directories
Local Directory
Local Directory with Referrals

1. Request

2. Referral

3. New Request

Client

Superior Server

Server
Replicated Directory

1. Update Request
2. Referral
3. New Request
4. Response
5. Replication Log
6. slurpd
7.
Distributed Directory

- Uses a mixture of *superior* and *subordinate* directories
  - With a mixture of referrals and replication
- Proprietary vendors invent own terms to describe complex directory structures
  - Microsoft
    - *ADS Forests* - Contain *Multiple Domains*
    - Domains within a forest can be *trusted*
    - *Foreign Domains* can be *trusted*
    - *Foreign forests* can be *trusted*
- Representation schematics of directories varies
Description of Directory Data

dc=berlin,dc=de

ou=people

cn=Peter Jutz

ObjectClass: Person
cn: Peter Jutz
telephoneNumber: 0162 323 6123
address: Ausfaht 10
...
Information Storage

- The full DN format is described in RFC2253

- Common Terms:
  dn, cn, dc, etc. are defined in the directory schema in ObjectClasses and are subject to rules that specify the type of data that may be stored

- There is a selection of common schemas

- Part of the configuration involves design of the directory
  - schemas, access controls, action controls, etc.
## Common Schema Files

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Schema File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corba RFC2714</td>
<td>corba.schema</td>
</tr>
<tr>
<td>Basic RFC2251-2256</td>
<td>core.schema</td>
</tr>
<tr>
<td>X.400 RFC1274</td>
<td>cosine.schema</td>
</tr>
<tr>
<td>DHCP Server Schema</td>
<td>dhcpp.schema</td>
</tr>
<tr>
<td>DNS Zone Schema</td>
<td>dnszone.schema</td>
</tr>
<tr>
<td>?</td>
<td>dyngroup.schema</td>
</tr>
<tr>
<td>Ximian Evolution Schema</td>
<td>evolutionperson.schema</td>
</tr>
<tr>
<td>InetOrgPerson RFC2798</td>
<td>inetorgperson.schema</td>
</tr>
<tr>
<td>Java Objects RFC2713</td>
<td>java.schema</td>
</tr>
<tr>
<td>Miscellaneous Objects</td>
<td>misc.schema</td>
</tr>
<tr>
<td>NIS RFC2307</td>
<td>nis.schema</td>
</tr>
<tr>
<td>Experimental OpenLDAP Schema</td>
<td>openldap.schema</td>
</tr>
<tr>
<td>NIS RFC2703bis</td>
<td>rfc2307bis.schema</td>
</tr>
<tr>
<td>Samba-3 - Caution: Still changing</td>
<td>samba3.schema</td>
</tr>
<tr>
<td>SuSE OpenExchange Schema</td>
<td>suse-mailserver.schema</td>
</tr>
<tr>
<td>SuSE YaST2 Schema</td>
<td>yast.schema</td>
</tr>
</tbody>
</table>
Log onto your computer:
`cd /etc/openldap/schema`

Now examine the contents of each file
Getting Started: Configuration & Initiation
Getting Started

- Edit `/etc/openldap/slapd.conf`
- Pre-requirements
  - Decide what type of directory you want
    - ie: Organizational, Domain Class, etc.
  - What type of information must be stored
    - ie: Schema components needed
  - Security requirements
  - Indexing requirements
- Start `slapd`
- Initialize the Directory
Telephone Directory

- Attributes
  - Common Name (cn)
  - Last Name (sn)
  - Address (Town / City)
  - Post Code
  - Telephone Number
  - email Address
Use of the OID

- core.schema:

```plaintext
attributetype ( 2.5.4.2 NAME 'knowledgeInformation' 
DESC 'RFC2256: knowledge information'
EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )

OID = Object Identifier
```