metaparadigm

Practical LDAP on Linux

A practical guide to integrating LDAP directory services on Linux

Michael Clark <michael@metaparadigm.com>
http://gort.metaparadigm.com/ldap/

Presentation Overview

- The need for LDAP
- LDAP Overview and Basics
- Setting up and tuning OpenLDAP
- Name services, authentication and authorisation
- Mail routing with sendmail and postfix
- Apache authentication
- Other LDAP tools and applications

The need for LDAP

- Multiple disparate sources of the same information
- Users need separate logins and passwords to login to different systems
- Complex to keep information in sync
- Similar data spread around many flat files or in database with different formats
- Inadequacies of NIS ie. Not very extensible
- X.500 is too complicated

LDAP Overview

- LDAP is a ‘Lightweight Directory Access Protocol’
- LDAP marries a lightweight DAP with the X.500 information model
- Uses an extensible hierarchical object data model
- An LDAP server may implement multiple ‘back-ends’: RDBMS, simple indexes (Berkeley DB), X.500 gateway
- Designed for frequent reads and infrequent writes

LDAP Benefits

- Standardised schemas exist for many purposes (well beyond that of NIS)
- Allows consolidation of many information sources
- Well defined API, support from many applications
- Easily replicated and distributed
- Multiple backends allow integration with existing data sources (RDBMS, etc)
- Much faster than RDBMS (using lightweight backend like Berkeley DB)

LDAP Basics

```
dc=com
   \-----  dc=metaparadigm
                    |-----  ou=people
                    |           \-----  uid=mclark
                    \-----  ou=groups
                                \-----  cn=users
```

- Data is organised into an hierarchical tree
- Each ‘entry’ (tree node) is identified by a DN (distinguished name) e.g. **`uid=mclark,ou=people,dc=metaparadigm,dc=com`**
- Each component of a DN is called an RDN (relative DN) and represents a branch in the tree
- The RDN must be unique within the nodes at the same level of the tree (is generally equivalent to one of the attributes ie. ‘uid’ or ‘cn’ in the case of a person)
- Each node has 1 or many attribute values associated with it. Each attribute can have 1 or many values

LDAP Basics (cont.)

- ‘objectClass’ is a mandatory attribute which specifies the schema (attribute constraints) for the given node
- Multiple ‘objectClass’ attributes can be combined together to achieve inheritance
- Example ‘objectClass’ (common schema) attributes: **`dcObject, organizationalUnit, person, organizationalPerson, inetOrgPerson, inetLocalMailRecipient`**
- CN (Canonical Name) is another common attribute used to provide a unique name for a directory object

LDAP Schemas

- Many standard schemas exist including:
  - People schemas - person, organisationalPerson, inetOrgPerson, posixAccount, mailLocalRecpient, strongAuthenticationUser
  - Group schemas – groupOfUniqueNames, posixGroup, organisationalRole, roleMember
  - Host / Network schemas – domain, ipHost, ipNetwork, ipProtocol, ipService, ieee802Device, bootableDevice
- An invaluable schema repository from Alan Knowles at the Hong Kong Linux Centre:
  - <http://ldap.akbkhome.com/>

LDIF File format

- LDIF (Lightweight Directory Interchange Format) is used to import/export from a LDAP directory server

```
dn: dc=metaparadigm,dc=com
objectclass: dcObject
objectclass: organization
o: Metaparadigm Pte Ltd
dc: metaparadigm

dn: ou=people,dc=metaparadigm,dc=com
objectclass: organisationalUnit
ou: people

dn: uid=mclark,ou=people,dc=metaparadigm,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: mclark
cn: Michael Clark
givenname: Michael
sn: Clark
o: Metaparadigm Pte Ltd
userPassword: {SSHA}D3DT4BJyKicf+PJ1+eqkWMNRG/B28xt+
mail: michael@metaparadigm.com
```

Custom schemas

- LDAP schemas uses SNMP style OIDs (Object Ids) for uniquely defining schema elements
- Apply for IANA enterprise number here:
  - <http://www.iana.org/cgi-bin/enterprise.pl>
- Private enterprise number OID prefix is 1.3.6.1.4.1 eg. Metaparadigm uses `1.3.6.1.4.1.11137`
- Information on custom schemas can be found here:
  - <http://www.openldap.org/doc/admin/schema.html>

Linux LDAP servers

- OpenLDAP is the primary open-source LDAP implementation based on Univ. Michigan LDAP <http://www.openldap.org/>
- Sun provides the iPlanet Directory Server
- Oracle provides an LDAP server using an Oracle database backend
- Many others available (Innosoft)
- Linux can also integrate with LDAP servers running on other platforms such as Microsoft Active Directory or Novell eDirectory

Scalability and Fault Tolerance

- OpenLDAP supports real-time directory replication to provide load-balancing and high availibility
- OpenLDAP supports single master, multiple slaves
- Most LDAP aware applications can be configured to use multiple LDAP servers (providing fallback servers)
- Multiple master support is in the works (currently alpha)
- OpenLDAP can be integrated with ‘heartbeat’ and ‘mon’ to provide fault tolerance <http://www.linux-ha.org/>

Setting up OpenLDAP

- Configuration is located in: `/etc/openldap/slapd.conf`
- We need to include the schemas we are using

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/misc.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema
```

- Next we specify a database

```
database     ldbm
suffix       “dc=metaparadigm,dc=com"
rootdn       "cn=Manager,dc=metaparadigm,dc=com "
rootpw       {crypt}mvRCcD3ajNmf2
directory    /opt/openldap/var/openldap-ldbm
index        objectClass eq
```

Setting up OpenLDAP (cont.)

- We can now start slapd (Standalone LDAP daemon)

```
# /etc/init.d/ldap start
```

- Next step is to add data to the directory using the LDIF example presented earlier

```
# ldapadd -D cn=Manager,dc=metaparadigm,dc=com -W < init.ldif
Enter LDAP Password: xxxxx
adding new entry "dc=metaparadigm,dc=com"

adding new entry "ou=people,dc=metaparadigm,dc=com"

adding new entry "uid=mclark,ou=people,dc=metaparadigm,dc=com"
```

Tuning OpenLDAP

- We need to add additional indexes for performance

```
index uidNumber,gidNumber,mailLocalAddress pres,eq
index cn,sn,givenName,memberUid,uid,mail pres,eq,sub
```

- We need to add ACLs for security

```
access to attr=userPassword
            by self write
            by anonymous auth
            by * none

access to dn="" by * read

access to *
            by self write
            by users read
            by anonymous auth
```

Tuning OpenLDAP (cont.)

- Setup logging in syslog.conf (default is LOCAL4)

```
local4.*                                                /var/log/ldap.log
```

- Make sure ‘slapd’ runs as non privileged user
- Make ‘slapd’ bind to SSL port for security
  - need signed certificates with openSSL and modify slapd.conf

```
TLSCertificateFile /etc/openldap/ldap.metaparadigm.com.cer
TLSCertificateKeyFile /etc/openldap/ldap.metaparadigm.com.key
```

  - modify init script to bind to SSL port

```
/usr/libexec/slapd -h 'ldap://ldap.metaparadigm.com/ ldaps://ldap.metaparadigm.com/' \
                -l LOCAL4 -u ldap -g ldap
```

LDAP Search Filters

- LDAP uses a simple ‘search filters’ syntax (RFC2254)
- LDAP queries return all attributes of matching entries (or specifically selected attributes) which match the search filter
- LDAP query particles are enclosed within parenthesis in the form of (attribute <matching rule> value) ie. `(cn=Michael Clark)`
- Matching rules include (=, =~, >=, <=)
- * can be used as a wildcard within the value
- These can be combined together using the boolean operators: and, or and not (&, |, !) eg:
  - `(&(cn=Michael Clark)(objectClass=posixAccount))`
  - `(&(objectClass=inetOrgPerson)(!(o=Microsoft*)))`
  - `(|(cn=Michael*)(cn=Mike*))`

LDAP Search Filters

- The following example ldap search retrieves the names and email address of all users with a givenname of ‘Michael’ or ‘Mark’

```
# ldapsearch -LLL -h ldap1-prd -b dc=ofs,dc=edu,dc=sg \
    '(&(|(givenname=Michael)(givenname=Mark))(objectClass=inetOrgPerson))' cn mail

dn: uid=mark_bergeron,ou=people,dc=ofs,dc=edu,dc=sg
mail: mark_bergeron@ofs.edu.sg
cn: Mark Bergeron

dn: uid=michael,ou=people,dc=ofs,dc=edu,dc=sg
mail: michael_chen@ofs.edu.sg
cn: Michael Chen

dn: uid=mclark,ou=people,dc=ofs,dc=edu,dc=sg
mail: michael_clark@ofs.edu.sg
cn: Michael Clark
…
```

- Very easy to incorporate this into shell scripts with awk or sed

Unix Name service

- LDAP integrates with NSS (Name Service Switch) using the nss_ldap module <http://www.padl.com/OSS/nss_ldap.html/>
  - Requires configuration of `/etc/ldap.conf`

```
host ldap.metaparadigm.com
base dc=metaparadigm,dc=com
ldap_version 3
binddn cn=Manager,dc=metaparadigm,dc=com
bindpw secret
pam_filter objectclass=posixAccount
pam_login_attribute uid
pam_member_attribute memberUid
nss_base_passwd ou=people,dc=metaparadigm,dc=com?one
nss_base_group dc=metaparadigm,dc=com?sub
```

  - Unix lookups are redirected in the same way as NIS: `/etc/nsswitch.conf`

```
passwd:     files nisplus ldap
shadow:     files nisplus
group:      files nisplus ldap
hosts:      files nisplus dns ldap
```

LDAP authentication

- LDAP integrates with PAM (Pluggable Authentication Modules) using pam_ldap <http://www.padl.com/OSS/pam_ldap.html>
  - pam_ldap shares /etc/ldap.conf with nss_ldap.conf
  - We create a pam definition file: **`/etc/pam.d/ldap-auth`**

```
#%PAM-1.0
auth         required      /lib/security/pam_env.so
auth         sufficient    /lib/security/pam_unix.so likeauth nullok
auth         sufficient    /lib/security/pam_ldap.so
auth         required      /lib/security/pam_deny.so
account      required      /lib/security/pam_unix.so
session      required      /lib/security/pam_limits.so
session      required      /lib/security/pam_unix.so
```

  - We point a services auth at LDAP eg. **`/etc/pam.d/imap`**

```
auth         required    /lib/security/pam_stack.so service=ldap-auth
account      required    /lib/security/pam_stack.so service=ldap-auth
session      required    /lib/security/pam_stack.so service=ldap-auth
```

Mail routing - sendmail

- We use the ‘inetLocalMailRecipient’ schema which extends ‘inetOrgPerson’
- Additional attributes ‘mailLocalAddress’, ‘mailHost’ and ‘mailRoutingAddress’. Users can have multiple ‘mailLocalAddress’ attributes.
- Allows for easily distributed multiple back-end mail stores
- Below are changes to **`sendmail.mc`** (tested in 8.11.x)

```
define(`confLDAP_DEFAULT_SPEC',`-h ldap.metaparadigm.com -b dc=metaparadigm,dc=com')dnl
FEATURE(ldap_routing)dnl
LDAPROUTE_DOMAIN(metaparadigm.com)
```

Mail routing – sendmail (cont.)

- Any sendmail map can be defined using LDAP.
- Example of custom alias map using 8.11.x (should also work on 8.12.x only official map schema support is available)

```
LOCAL_CONFIG
undefine(`ALIAS_FILE')
Kldapaliases ldap -z, -v mailForwardingAddress -k (&(objectClass=mailForwardingAlias)(mailAlias=%0))
O AliasFile=sequence:ldapaliases
```

```
attributetype ( 1.3.6.1.4.1.11137.3.1.48
        NAME 'mailAlias' DESC 'alias part of address'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.11137.3.1.47
        NAME 'mailForwardingAddress' DESC 'RFC822 address to use'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} )

objectclass ( 1.3.6.1.4.1.11137.3.2.49
        NAME 'mailForwardingAlias' DESC 'Internet local mail recipient'
        SUP top AUXILIARY
        MAY  ( cn $ o $ mail $ mailAlias $ mailForwardingAddress ) )
```

Mail routing - postfix

- Postfix documentation specifies a non-standard schema. Below example is implemented with same schema as sendmail. (change in `/etc/postfix/main.cf`)

```
virtual_maps = ldap:metaroute

metaroute_server_host = ldap.metaparadigm.com
metaroute_search_base = dc=metaparadigm,dc=com
metaroute_query_filter = (mailLocalAddress=%s)
metaroute_result_attribute = mailRoutingAddress
metaroute_domain = metaparadigm.com
metaroute_bind = no
```

- Transport map can also be LDAP routed to provide mutliple backend support

Shared Address book

- LDAP provides a convenient Corporate style shared address book similar to that of Exchange which is not otherwise available with standard Internet e-mail
- Support in almost all email clients:
  - Evolution
  - Mozilla
  - Outlook
  - Eudora
  - Various web mail clients
  - …

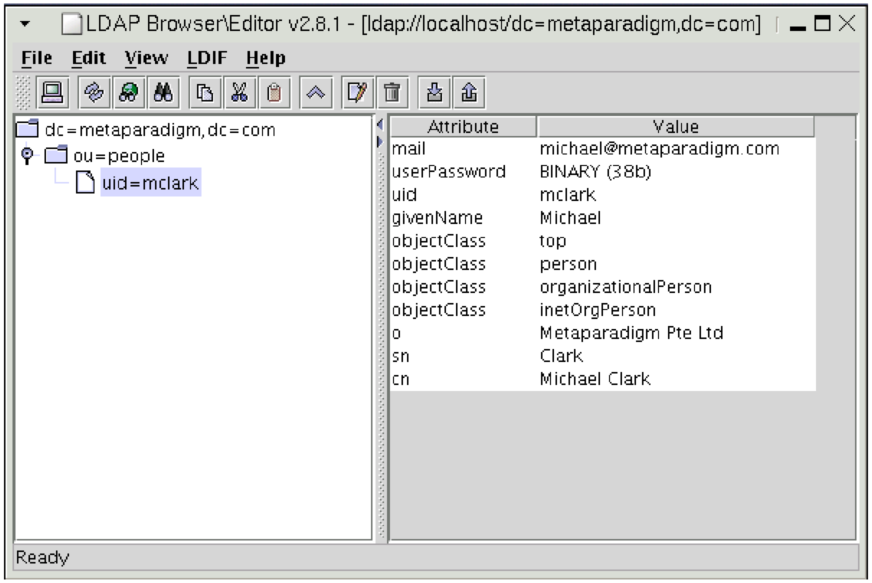
Apache Authentication

- 2 Apache modules available
  - ‘mod_auth_ldap’ apache module <http://nona.net/software/ldap/>
  - ‘auth_ldap’ apache module <http://www.rudedog.org/auth_ldap/>
- Example httpd.conf using mod_auth_ldap

```
<Location /secret>
  AuthType Basic
  AuthName "Secret"
  AuthLDAPURL ldap://ldap.metaparadigm.com:389/ou=people,dc=metaparadigm,dc=com?uid
  require valid-user
</Location>
```

LDAP Browsers and Editors

- Java LDAP Browser <http://www.iit.edu/~gawojar/ldap/>
  - Allows easy updating and editing of directory information.
  - Can create templates for commonly used directory objects.
- Huge number of other tools (web, GTK, …)


LDAP Browser\Editor v2.8.1 - [ldap://localhost/dc=metaparadigm,dc=com]
File
Edit
View
LDIF
Help
dc=metaparadigm,dc=com
ou=people
uid=mclark
Ready

Attribute	Value
mail	michael@metaparadigm.com
userPassword	BINARY (38b)
uid	mclark
givenName	Michael
objectClass	top
objectClass	person
objectClass	organizationalPerson
objectClass	inetOrgPerson
o	Metaparadigm Pte Ltd
sn	Clark
cn	Michael Clark

Migration to LDAP

- Padl migration tools
  - <http://www.padl.com/OSS/MigrationTools.html>
- Migrates existing flat files or NIS databases
  - passwd, group, hosts, networks, services, etc…

Other Application support

- Samba LDAP-PDC
  - <http://www.unav.es/cti/ldap-smb-howto.html>
- LDAP DNS (no more HUPing named)
  - http://www.nimh.org/code/ldapdns/
- RADIUS (various patches floating around)
- Any application that supports PAM
  - ssh, netatalk, many others…