LDAP

Getting Started with LDAP for small and large setups.

What is LDAP

- Lightweight Directory Access Protocol
  - Phonebook
- Based on X.500 DAP on OSI stack
- Developed in '93 @ UMich
- LDAPv3, Internet Standard RFC4510
  - RFC 4510-4521 and about 40 others for extensions
- Many implementations
  - OpenLDAP
  - Apache Directory Server
  - Sun Java System Directory (SunONE/iPlanet/Netscape)
  - Novell eDirectory
  - Microsoft Active Directory
  - many others...

What is a directory...

- Directory is a tree of entries
- Basic operations:
  - Search
  - Compare
  - Add
  - Modify
  - Delete
- Optimized for quick access, read performance
- LDAP server can serve multiple trees
- Schema's define and describe the contents of the directory
  - Collection of Attributes and Classes:
    - Syntax
    - Globally unique Object Identifiers (ASN.1)

What can be stored in LDAP

- Basically anything you can think of
- Mostly used for:
  - User accounts and group related data
  - Phone and address book
  - Mail accounts
  - Configuration data for various systems
- Other examples:
  - Sudo configuration
  - Evolution Addressbook
  - Bitlbee configuration and buddy-list
  - CUPS configuration
  - ...

Example LDAP Data

LDAP data is usually presented in LDIF (LDAP Data Interchange Format).
dn: o=Snow, c=nl
o: Snow
objectclass: organization dn: cn=Mark Janssen, o=Snow, c=nl
cn: Mark Janssen
sn: Janssen
mail: m.janssen@snow.nl
objectclass: person

A sample schema

```
...
attributetype ( 1.3.6.1.4.1.15953.9.1.5
    NAME 'sudoOption'
    DESC 'Options(s) followed by sudo'
    EQUALITY caseExactIA5Match
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 ) # IA5String (7-bit ascii)

objectclass ( 1.3.6.1.4.1.15953.9.2.1
    NAME 'sudoRole' SUP top STRUCTURAL
    DESC 'Sudoer Entries'
    MUST ( cn )
    MAY ( sudoUser $ ... $ sudoCommand $ ... $ sudoOption $ description )
    )
...
```

OpenLDAP

- Open source LDAP directory (slapd), interface library (libldap, liblber) and client utilities (ldapadd, ldapsearch, ldapmodify, etc)
- Open Source / Free Software, basically BSD-style license (OpenLDAP License)
- Front-end server (ldap interface) with various back-ends:
  - Storage backend (bdb,hdb, mysql, ldif)
  - Proxy backends (passwd, ldap, sql, ...)
  - Misc/Dynamic (config, monitor, perl, null, ...)
- Overlay support
  - Change presentation of data
  - Logging
  - Custom stuff
- Kernel-like versioning, current stable versions 2.2 and 2.4

Building OpenLDAP

- Requirements:
  - BerkeleyDB 4.7
  - OpenSSL
- Configure: ./configure --with-tls=openssl --enable-overlays --enable-crypt --enable-modules --enable-monitor --prefix=/opt/openldap --enable-syslog --enable-proctitle --without-subdir
- Installing
- Gathering additional schema's
  - http://web.singnet.com.sg/~garyttt/solaris.schema.txt
  - http://www.sudo.ws/cgi-bin/cvsweb/~checkout~/sudo/schema.OpenLDAP?rev=1.3
- Or use you distro-provided package if available

Server configuration: slapd.conf

- Include schema definitions
- SSL configuration
- ACL's
- Database definition
- Indexes
- Overlays
- Sizing/Tuning

slapd.conf

```
include schema/core.schema
include schema/cosine.schema
include schema/nis.schema
include schema/solaris.schema
include schema/ppolicy.schema
include schema/duaconf.schema
include schema/sudo.schema

# TLS Certificate
TLSCACertificateFile cacert.pem
TLSCertificateFile servercrt.pem
TLSCertificateKeyFile server.key
TLSVerifyClient never

# ACL's
access to *
  by self read
  by * read
```

```
database bdb
suffix "dc=company,dc=nl"
rootdn "cn=Manager,dc=company,dc=nl"

rootpw {SSHA}PassWordHash

# Indices to maintain

index objectClass,uid,uidNumber,index \
                                gidNumber,ou eq
index cn,mail,surname    eq,subinitial
index memberUid          eq
index nisDomain          eq
index uniqueMember       pres

# OVERLAY definitions:
overlay ppolicy
ppolicy_default "cn=default,
ou=policies,       dc=company,dc=nl"
password-hash {SSHA}
```

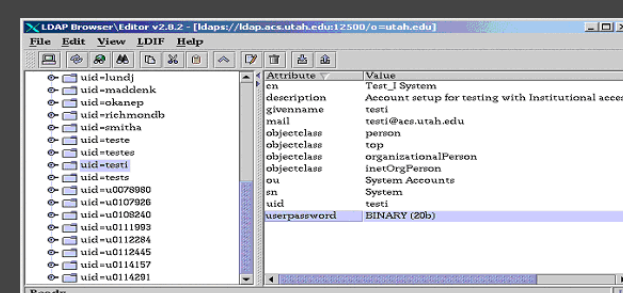
Loading initial content

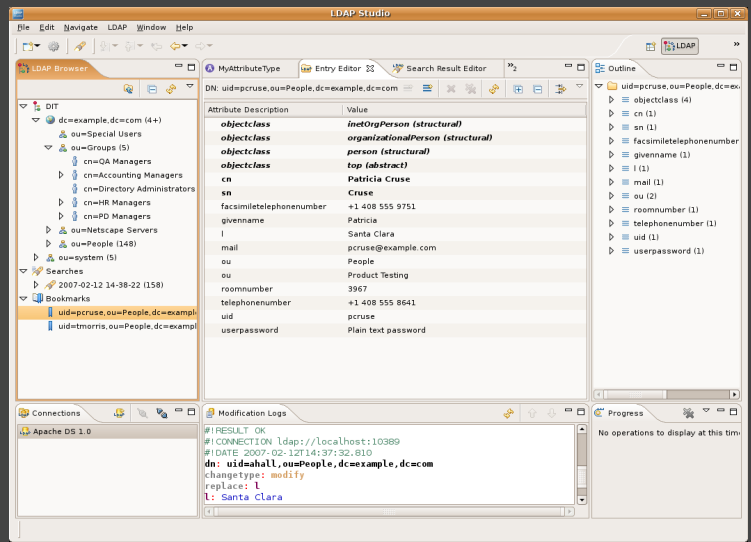
```
dn: dc=company,dc=nl
associatedDomain: company.nl
dc: company
objectClass: top
objectClass: dcObject
objectClass: domain
objectClass: domainRelatedObject
objectClass: nisDomainObject
nisDomain: company.nl
o: Your Company Namedn:
ou=People,dc=company,dc=nl
ou: People
objectClass: top
objectClass: organizationalUnit
dn: cn=Users,ou=Group,dc=com...
gidNumber: 1000
objectClass: top
objectClass: posixGroup
cn: Users
```

```
dn: cn=proxyagent,ou=People,...
userPassword:: PASSWORDHASH
objectClass: top
objectClass: person
sn: proxyagent
cn: proxyagentdn: cn=Manager, dc=company,dc=nl
userPassword:: PASSWORDHASH
objectClass: person
objectClass: top
sn: Manager
cn: Manager
$ ldapadd -D binddn -w secret \
    -b dc=company,dc=nl -f initial.ldif
```

Interacting with your directory

- Command-line tools (ldapsearch, ldapadd, ldapmodify)
- LBE: Ldap Browser and Editor (missing in action)
- Apache Directory Studio: http://directory.apache.org/studio/





Apache Directory Studio

Access control considerations

- Based on first match
- Specify subtree and/or attributes
- Rights: None/Auth/Read/Write
- User specifier: Wildcard, Anonymous, Self or specified.

- Allow access to public data
- Limit access to sensitive data
- Disallow access to private data
- Allow users to modify some fields (contact info)
- Allow system-tools access to posix account fields

ACL Examples

```
access to dn.subtree="ou=People,dc=domain,dc=tld" \
attrs=userPassword,shadowLastChange
  by dn="cn=proxyagent,ou=profile,dc=domain,dc=tld" write
  by dn="cn=webagent,ou=profile,dc=domain,dc=tld" auth
  by self write
  by anonymous auth
  by * read

access to attrs=uid,uidNumber,gidNumber,memberUid
       by * read

access to dn.subtree="ou=SUDOers,dc=domain,dc=tld"
  by dn="cn=sudoagent,ou=profile,dc=domain,dc=tld" read
  by * none

access to *
       by * read
```

Limit access to fields userPassword and shadowLastChange

Prevent users from changing their unix account rights

Limit a tree to a specific user or authorization.

End-all passthrough rule.

Client configuration - generic/linux

- Generic
  - Install pam-ldap and nss-ldap
  - place your cacert.pem file and certificates in /etc/ldap/
  - edit pam.conf, nsswitch.conf, /etc/ldap/ldap.conf
- Red Hat Enterprise 4 or 5
  - pre-populate /etc/ldap.conf with binddn and bindpw values (can't specify these in config-tool yet)
  - authconfig or system-config-authentication
  - Check 'Use LDAP', 'Use TLS', specify server/basedn
  - Check 'Cache Information' (enable nscd)
  - Check 'Use LDAP Authentication' and 'Local autentication is sufficient'
  - Further ldap.conf, pam.conf and nsswitch.conf configuration is done for you by authconfig.

Client configuration - Unix

- Solaris 10
  - Create or update certificate store
    - certutil -N -d /var/ldap
    - certutil -A -d /var/ldap -n 'CA Name' -i /path/to/cacert.pem -a -t CT
  - Edit /etc/nsswitch.ldap, making sure to change the entries for hosts and ipnodes to ‘files dns’
  - ldapclient init -v -a proxyDN=cn=proxyagent,dc=domain,dc=tld -a proxyPassword=secret -a domainName=domain.tld -a profileName=tls_profile ldapserver.domain.tld ldapserver2.domain.tld
  - Modify pam.conf to support ldap
- AIX 5.3 / 6.1
  - Use gsk7ikm to convert cacert.pem to a keydb
  - Install client binaries (idsldap.clt32bit61.rte, idsldap.clt64bit61.rte, idsldap.cltbase61.adt, idsldap.cltbase61.rte)
  - mkseccldap -c -h ldapserver1.domain.tld,ldapserver2 -a cn=proxyagent, dc=domain,dc=tld -p secret -k /path/to/your-keydb.kdb -w keydbpassword -A ldap_auth

Config files: /etc/(ldap)/ldap.conf

```
binddn cn=proxyagent,dc=domain,dc=tld
bindpw secret
base dc=domain,dc=tld
timelimit 120
bind_timelimit 120
idle_timelimit 3600
nss_initgroups_ignoreusers root,ldap,named,avahi,haldaemon,
dbus,nscd,gdm
uri ldap://ldapserver1.domain.tld ldap://ldapserver2
ssl start_tls
tls_cacertfile /path/to/cacert.pem
pam_password md5
```

Config files: /etc/nsswitch.conf

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap
hosts:      files dns
netgroup:   files ldap
automount: files ldap
sudoers:    files ldap
```

Config files: Sample pam config

```
# Sufficient samples are included with pam_ldap and pam is hightly
# OS/System dependant, this is just an example, don't just start using this.
# /etc/pam.d/login

#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_nologin.so
auth       sufficient   /lib/security/pam_ldap.so
auth       required     /lib/security/pam_unix_auth.so try_first_pass
account    sufficient   /lib/security/pam_ldap.so
account    required     /lib/security/pam_unix_acct.so
password   required     /lib/security/pam_cracklib.so
password   required     /lib/security/pam_ldap.so
password   required     /lib/security/pam_pwdb.so use_first_pass
session    required     /lib/security/pam_unix_session.so
```

Enhancing your LDAP config

- Password management (expiration, quality)
  - overlay ppolicy
    ppolicy_default "cn=default,ou=policies,dc=..."
    ppolicy_hash_cleartext on
    ppolicy_use_lockout
- Unique attributes
  - overlay unique
    unique_uri ldap:///ou=People,dc=dom,dc=tld?uidNumber,uid?sub
    unique_uri ldap:///ou=Group,dc=dom.dc=tld?gidNumber,cn?sub
- Replication (provider part)
  - overlay syncprov
  - syncprov-checkpoint 100 10
  - syncprov-sessionlog 100

Replication -- Receiver

```
syncrepl rid=001
   provider=ldap://ldap2.domain.tld
   bindmethod=simple
   starttls=critical
   binddn="cn=proxyagent,ou=profile,dc=domain,dc=tld"
   credentials=secret
   searchbase="dc=domain,dc=tld"
   schemachecking=on
   type=refreshAndPersist
   retry="60 +"

# 2-Master mode
mirrormode on
```

Adding LDAP support to tools

- SUDO
  - Use version 1.7 (or 1.6 with patch)
  - Usage: Highly recommended
- Secure Shell
  - Store users 'authorized_keys' in LDAP
  - Use 'openssh-lpk' patch
  - Usage: Handy for large installations/userbases
  - Still needs some work, not standard

Thank you... any questions ?

References:

- http://en.wikipedia.org/wiki/Ldap
- http://blog.maniac.nl/tag/ldap/
- ...

Mark Janssen
E-Mail: m.janssen@snow.nl
IRC: Foo-Bar (IRCnet), FooBar (OFTC)
Jabber: maniac.nl@gmail.com