

# How to use LDAP for User management

Practical Introduction

Lukas Beeler <[lb-sucon04@projectdream.org](mailto:lb-sucon04@projectdream.org)>

Second Swiss Unix Conference 2004

# Overview

- **0900-0945: Introduction to LDAP (Theory)**
  - How does it work?
  - What can i do with it?
- **1000-1045: Setting up LDAP**
  - Configure OpenLDAP
  - Configure OS for LDAP Authentication
- **1100-1145: Other uses and questions**

# What is LDAP?

Lightweight Directory Access Protocol

- A directory service
- An object-oriented database

# What can LDAP do?

- Provide data to clients
- Search data with `filters`
- Access specific informations from an object

# Common usages

- Central data management
  - Authentication
  - E-Mail
  - Address books
  - Accounting
  - etc.

# Implementations

- OSS (OpenLDAP, tinyldap, etc.)
- Sun (Part of SunOne)
- Netscape (NDS)
- Microsoft (Part of Active Directory)
- Novell (Part of eDirectory)
- some more

# Object Orientation

- Each object has a distinguished name (dn)
- Each object has a set of attributes
- Each object belongs to one or more object classes
- Each attribute may hold an arbitrary amount of data

# Example object

```
dn: uid=lb,ou=People,dc=ds,dc=suug,dc=ch
objectClass: posixAccount
cn: Lukas Beeler
uid: lb
uidNumber: 10000
gidNumber: 10000
homeDirectory: /import/home/lb
```

# distinguished name

dn: uid=lb,ou=People,dc=ds,dc=suug,dc=ch

- Unique
- Specifies the object's location in the tree
- Consists of comma-separated attributes
- The last part is called 'base dn'

# base distinguished name

`dc=ds,dc=suug,dc=ch`

- The root of a ldap directory
- May consist of one or more attributes
- Multiple usual formats:

## Based on DNS:

`o=ds.suug.ch` (o as in Organization)

`dc=ds,dc=suug,dc=ch` (dc as in Domain Component)

## Based on X500:

`o="Swiss Unix User Group",c=CH` (c as in Country)

# object classes

`objectClass: posixAccount`

- Specifies mandatory/available attributes for an object
- Configured on the LDAP server with a schema
- Some objectclasses are standardized
- Some Applications come with their own

# Attributes

```
cn: Lukas Beeler  
uid: lb  
uidNumber: 10000
```

- Key:Value
- Value usually consists of plain text
- May contain encoded binary data
- Are assigned to objects through object classes

# organizational unit

dn: ou=People,dc=ds,dc=suug,dc=ch

ou: People

objectClass: top

objectClass: organizationalUnit

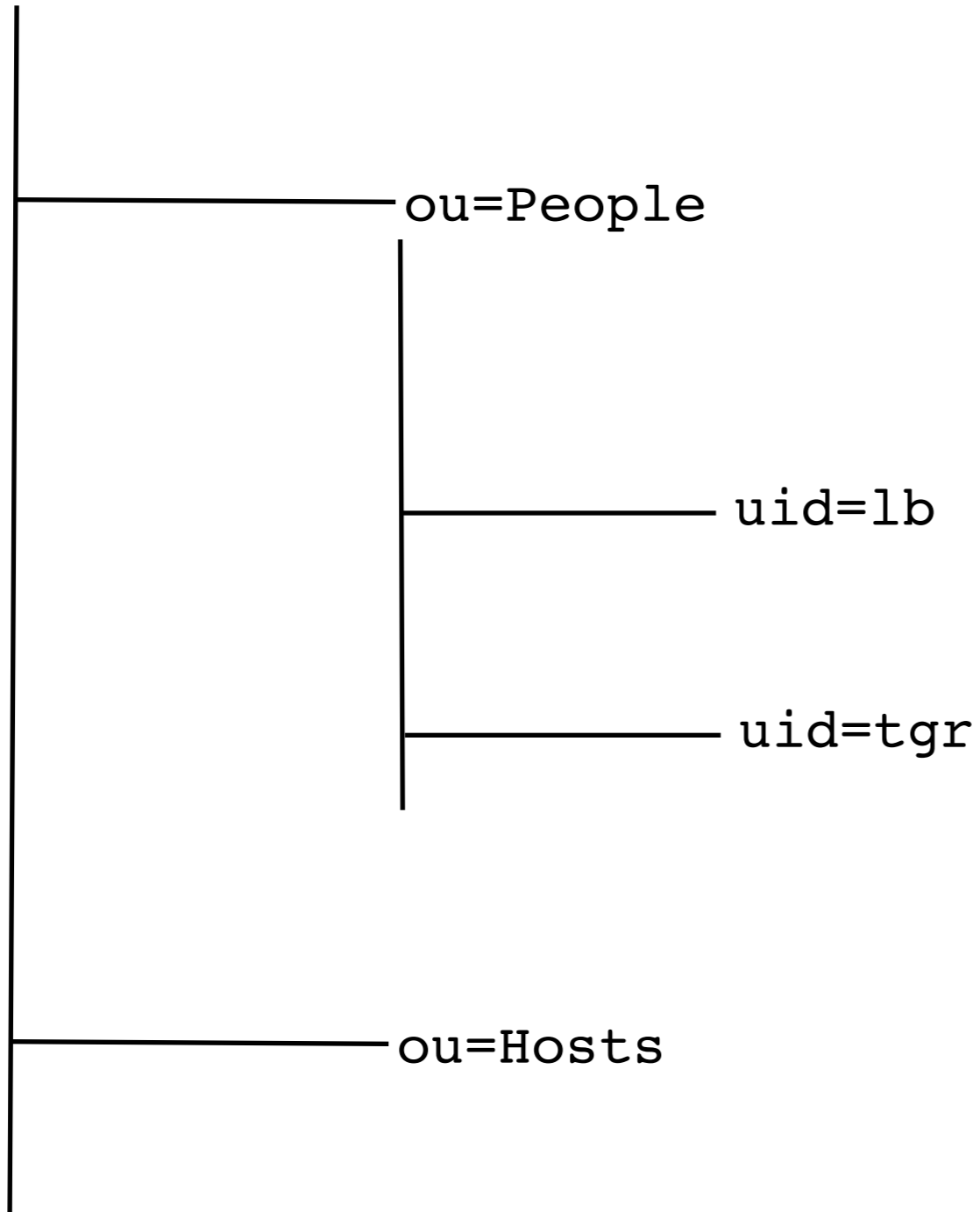
- Structural Objects
- Used to build the LDAP tree structure

# A tree-like Structure

- LDAP is organized as a tree
- The root of the tree is the base dn
- ou's are used as branches
- Objects are the leafs

# Example tree

dc=ds , dc=suug , dc=ch



# Schemas

```
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount' SUP top AUXILIARY
  DESC 'Abstraction of an account with POSIX attributes'
  MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
  MAY ( userPassword $ loginShell $ gecos $ description ) )
```

```
attributetype ( 1.3.6.1.1.1.1.0 NAME 'uidNumber'
  DESC 'An integer uniquely identifying a user in an administrative domain'
  EQUALITY integerMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
```

- Schemas define objectclasses and attributes
- Some attributes might be mandatory, some are not
- An attribute might be specified to only hold a certain type of data (e.g. an Integer)

# Searching with LDAP

- CLI utility `ldapsearch`
- Anonymous searches usually possible
- Administrative privileges necessary to see all attributes

# Simple Filters

<code>(&amp;(uid=lb)(objectClass=suugMember))</code>	<b>AND</b>
<code>( (uid=lb)(uid=tgr))</code>	<b>OR</b>
<code>-b ou=Hosts,dc=ds,dc=suug,dc=ch aRecord=195.134.158.23</code>	<b>Namespace Subset</b>
<code>( (uid=lb)(uid=tgr)) mailLocalAddress</code>	<b>Fetch single attributes</b>
<code>uid=lb</code>	<b>Simple Search</b>

# Complex Filters

```
( | (&(mailEnabled=1) (mailLocalAddress=foo@bar.com) ) (&(uid=1b) (objectClass=suugMember) ) )
```

Searches for Objects where:

mailEnabled=1, AND

mailLocalAddress=foo@bar.com

OR

uid=1b, AND

objectClass=suugMember

# Setting up LDAP

- Installing OpenLDAP on a Debian system
- Configuring the initial directory layout
- Configuring the system for ldap authentication

# Installing the software

- **run** `apt-get install slapd ldap-utils`
- Ignore debconf
- Add a system user (e.g. ldap)
- Create SSL/TLS Certs
- Clean `/var/lib/ldap/*` (debconf created)

# Initial Configuration

- Schemas to load
- SSL/TLS certs to use
- Database engine/base DN/etc.
- Indices/ACLs
- root `user` for db recovery
- Commented config file available at:  
<http://projectdream.org/publications/suug/ldap-setup.html>

# ACLs

```
access to attribute=loginShell,shadowLastChange,cn,title,  
    by dn="cn=admin,dc=ds,dc=suug,dc=ch" write  
    by self write  
    by self read  
    by * read
```

Allows the admin user to write to those attributes.

Allows the object to write to its own attributes (in case of successful authentication, of course).

Allows everyone to read those attributes.

# Indices

index	uidNumber	eq
index	gidNumber	eq
index	uid	eq
index	memberUid	eq

Sample indices. There are sample indices for all major schemas available on the internet.

Remember to shutdown `slapd` and run `slapindex` to rebuild the index database.

# OpenLDAP naming conventions

- slapd: LDAP daemon
- slap\*: Offline tools, unsafe to use when slapd is running, only fs privileges needed
- ldap\*: Online tools, ldap privileges needed

# Initial Directory Layout

```
# slapadd <<'EOF'  
dn: dc=ds,dc=suug,dc=ch  
objectClass: top  
objectClass: dcObject  
objectClass: organization  
o: Swiss Unix User Group  
dc: ds  
  
dn: cn=admin,dc=ds,dc=suug,dc=ch  
objectClass: simpleSecurityObject  
objectClass: organizationalRole  
cn: admin  
description: LDAP administrator  
userPassword: foo  
EOF
```

- Adds base DN to directory
- Adds admin user to directory

# Testing

- Files in `/var/lib/ldap` should belong to the 'ldap' user
- Start slapd

```
# ldapsearch -LLL -x cn=admin
dn: cn=admin,dc=ds,dc=suug,dc=ch
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
```

# More initial config

## Change the admin password:

```
# ldappasswd -D cn=admin,dc=ds,dc=suug,dc=ch -W -x -S
New password:
Re-enter new password:
Enter LDAP Password: foo
Result: Success (0)
```

## Add People OU

```
# ldapadd -D cn=admin,dc=ds,dc=suug,dc=ch -W -x <<'EOF'
dn: ou=People,dc=ds,dc=suug,dc=ch
ou: People
objectClass: top
objectClass: organizationalUnit
EOF
```

# The first user

```
# ldapadd -D cn=admin,dc=ds,dc=suug,dc=ch -W -x <<'EOF'  
dn: uid=lb,ou=People,dc=ds,dc=suug,dc=ch  
uid: lb  
uidNumber: 10000  
gidNumber: 10000  
homeDirectory: /home/lb  
cn: Lukas Beeler  
  
EOF
```

## Minimalistic user entry

# Setting a password

- This manual procedure can be used if PAM is not yet configured
- Requires LDAP privileges

```
# ldappasswd -D cn=admin,dc=ds,dc=suug,dc=ch -x \  
> -W -S uid=lb,ou=People,dc=ds,dc=suug,dc=ch
```

```
New password:
```

```
Re-enter new password:
```

```
Enter LDAP Password:
```

```
Result: Success (0)
```

# Authentication

NSS - Name Service Switch:

Resolves UIDs to names and vice versa.

PAM - Pluggable Authentication Modules:

Handles logon, password changes, etc.

# Configuring NSS

- Run `apt-get install libnss-ldap`
- Answer debconf questions or  
`edit /etc/libnss-ldap.conf`
- Modify `/etc/nsswitch.conf` like this:  
`passwd:                  compat ldap`

# /etc/libnss-ldap.conf

```
host 127.0.0.1  
base dc=ds,dc=suug,dc=ch  
ldap_version 3
```

Looks simple? It is.

# Testing NSS

- Run the `id` command
- Our GID isn't resolved to a group name

```
# id lb  
uid=10000(lb) gid=10000 groups=10000
```

# Configuring PAM

- Run `apt-get install libpam-ldap`
- Answer debconf questions or edit `/etc/pam_ldap.conf`
- Write LDAP admin password to `/etc/ldap.secret`
- Adjust PAM configuration in `/etc/pam.d/`

# /etc/pam\_ldap.conf

```
host 127.0.0.1  
base dc=ds,dc=suug,dc=ch  
rootbinddn cn=admin,dc=ds,dc=suug,dc=ch  
pam_password md5
```

Still pretty simple.

# /etc/pam.d/

Debian Sarge has /etc/pam.d/common-\*

```
# grep -v ^# common-auth
auth    [success=done new_authtok_reqd=done default=ignore auth_err=bad] pam_ldap.so
auth    required          pam_unix.so try_first_pass
# grep -v ^# common-account
account sufficient      pam_ldap.so
account required       pam_unix.so
# grep -v ^# common-password
password sufficient    pam_ldap.so
password required     pam_unix.so md5
```

Some services might have their own /etc/pam.d/  
foo file

# Testing PAM

- Restart the service you want to use for testing (e.g. sshd)
- Try logging in with username and password

```
# ssh lb@localhost
lb@localhost's password:
Could not chdir to home directory /import/home/lb: No such file or directory
celestas% id
uid=10000(lb) gid=10000 groups=10000
```

# Adding Groups

```
# ldapadd -D cn=admin,dc=ds,dc=suug,dc=ch -W -x <<'EOF'
```

```
dn: ou=Group,dc=ds,dc=suug,dc=ch
```

```
ou: Group
```

```
objectClass: top
```

```
objectClass: organizationalUnit
```

```
dn: cn=lb,ou=Group,dc=ds,dc=suug,dc=ch
```

```
cn: lb
```

```
gidNumber: 10000
```

```
objectClass: top
```

```
objectClass: posixGroup
```

```
EOF
```

```
# id lb
```

```
uid=10000(lb) gid=10000(lb) groups=10000(lb)
```

# Review

- Initial directory layout using `slapadd`
- Started `slapd`
- Added OU and User using `ldapadd`
- Configured NSS and PAM
- Added OU and Group

# Troubleshooting

```
% ldapsearch  
ldap_sasl_interactive_bind_s: No such attribute (16)
```

You forgot the -x switch. LDAP defaults to SASL authentication.

# Troubleshooting

```
% slapcat  
zsh: segmentation fault  slapcat
```

slapcat must have read and write permission on /var/lib/ldap. This is an unhandled error in slapcat. The open fails, but the fd is accessed anyway.

```
% strace -eopen slapcat 2>&1  
open("/var/lib/ldap/__db.001", O_RDWR|O_LARGEFILE) =  
-1 EACCES (Permission denied)  
--- SIGSEGV (Segmentation fault) @ 0 (0) ---  
+++ killed by SIGSEGV +++
```

# Troubleshooting

```
% ldapsearch -ZZ  
ldap_start_tls: Connect error (91)  
    additional info: Error in the certificate.
```

The TLS libraries were unable to check the certificate or the hostname you are connecting to does not equal the cn in the SSL cert.

Add a TLS\_CACERT option to `/etc/ldap/ldap.conf`, and point it to the CA certificate you are using.

Correct the URI option in `/etc/ldap/ldap.conf` to point to the cn of the certificate you are using.

# Troubleshooting

```
slapadd: could not add entry dn="dc=ds,dc=suug,dc=ch"  
(line=7): txn_begin failed: Invalid argument (22)
```

The directory `/var/lib/ldap` does not exist.

# Troubleshooting

```
# id lb  
uid=10000(lb) gid=10000 groups=10000
```

Groups don't get resolved, even after they were added.

Most probably, the value for 'base' in `/etc/libnss-ldap.conf` has been set to a specific OU. Instead it should list the base DN.

# Troubleshooting

```
# ssh lb@localhost
```

```
lb@localhost's password:
```

```
Permission denied, please try again.
```

There is no password set for the user lb. Use `ldappasswd` to set one.

You did not restart the service after changing the PAM configuration.

# Replication

- Backups without downtime
- Reliability
- Performance

# Multimaster Replication

- All Servers are masters
- Editing possible when link is down
- Changes get merged after link up
- Currently 'EXPERIMENTAL' in OpenLDAP

# Master Slave Replication

- Read-only redundancy
- Read performance improvement
- Write access only on masters
- Stable and Well-tested in OpenLDAP

# Replication Setup

## On the master:

```
replica uri=ldap://naru.suug.ch  
        binddn="cn=Replicator,dc=ds,dc=suug,dc=ch"  
        bindmethod=simple credentials=foobar
```

## On the slave:

```
rootdn cn=Replicator,dc=ds,dc=suug,dc=ch  
# create password using slappasswd(1)  
rootpw {SSHA}SaKMI6SDIBxbG/GP6epR/aM/VRC+vhKI  
updatedn cn=Replicator,dc=ds,dc=suug,dc=ch  
updateref ldap://mahoro.suug.ch
```

# Replication Setup

On the master:

- Shutdown `slapd`
- `slapcat > db`
- `scp db naru:~`

On the slave:

- `slapadd < db`
- Start `slapd`

# Replication Setup

On the master:

- Start `slurpd`
- Start `slapd`

# Troubleshooting

DNS / Names need to be correct

After debugging, make sure to flush `slurpd`'s  
spool

# Mailrouting with LDAP

- Alias database
- Full virtual users possible
- SMTP Authentication

We will use Postfix 2.1 for our examples

# Aliases

- Configure postfix for another alias map
- Edit `/etc/postfix/main.cf`:

```
alias_maps = hash:/etc/aliases, ldap:/etc/postfix/ldap-aliases.cf
```

# Aliases

- Create `/etc/postfix/ldap-aliases.cf`

```
server_host = 127.0.0.1
search_base = ou=People,dc=ds,dc=suug,dc=ch
query_filter = (&(objectClass=inetLocalMailRecipient)(mailLocalAddress=%s@suug.ch))
result_attribute = mailRoutingAddress,mailForwardAddress
bind = no
version = 3
```

# SMTP Auth

- Edit `/etc/postfix/main.cf`
- Run `saslauthd` like this:  
`saslauthd -d -a pam`
- SASL supports native LDAP authentication  
for more complicated setups

```
smtpd_sasl_auth_enable = yes
```

```
smtpd_tls_auth_only = yes
```

```
smtpd_recipient_restrictions = permit_mynetworks, permit_sasl_authenticated,  
reject_unauth_destination
```

# DNS with LDAP

- PowerDNS supports a live LDAP Backend
- Unofficial Bind Patches are available
- LDAP to tinydns data dumpers are available

# Adress Books

- Many MUA's require LDAPv2, make sure to enable it in `/etc/ldap/slapd.conf`
- Reading entries is usually simple, but writing isn't always that easy

# End

- Questions?
- HOWTO:  
<http://projectdream.org/publications/suug/ldap-setup.html>
- Slides:  
<http://projectdream.org/publications/suug/ldap.pdf>