

# Unix System Administration

Chris Schenk

Lecture 26 – Thursday Apr 24

CSCI 4113, Spring 2008

# LDAP Object Classes

- Are the backbone of any LDAP entry
- Are defined within LDAP schema files
  - Under Ubuntu, they live in `/etc/ldap/schema`
- Contain required (must) & optional (may) fields
- `posixAccount`, `shadowAccount`, `posixGroup`
  - In `/etc/ldap/schema/nis.schema`
  - Common object classes for unix account info
  - Attributes belong to an object class
- Many other schemas available
  - `person`, `InetOrgPerson`, `ipHost`, `ipNetwork`, etc

# Building an LDAP Database

- By far the hardest thing to do
  - Followed by authenticating to one
- High-level steps:
  - Define a database (suffix) and where the files live
  - Set default hashing function for passwords
  - Set access control for that database
  - Set which attributes to index (make searchable)
  - Set 'root' (fully-privileged) user for database
  - Enable security using SSL (optional, but highly recommended)

# LDAP Files under Ubuntu

- ldap-utils – Client command line utilities
  - ldapsearch, ldapmodify, ldapdelete
  - Config file /etc/ldap/ldap.conf
- slapd – Actual server database
  - Lives under /etc/ldap as well
  - Config file /etc/ldap/slapd.conf
- libnss-ldap, libpam-ldap
  - Allows PAM to authenticate to LDAP
  - Config files /etc/pam\_ldap.conf, /etc/nsswitch.conf
  - /etc/libnss\_ldap.secret – For root to work properly

# LDAP - Define a Database

- `/etc/ldap/slapd.conf`:

- ```
database          bdb
suffix            "dc=coolname,dc=cs,dc=colorado,dc=edu"
rootdn            "cn=manager,dc=coolname,dc=cs,dc=colorado,dc=edu"
rootpw            {SSHA}8apbHW8OzBUC1CG9FZJBd2Dh2YlMHrgb
password-hash     {SSHA}
directory         /var/lib/ldap
index uidNumber,gidNumber,loginShell      eq,pres
index uid,memberUid                        eq,pres,sub
```

- Many databases may be defined in one server
  - A new database starts with the 'database' keyword
  - Differentiated by the suffix
- Notice root password is defined, but hashed
  - `slapd.conf` is only readable by root

# Keywords 'database' and 'suffix'

- 'database' defines the back-end to be used
  - Can be one of many things: bdb, perl, shell, ldbm, passwd
  - 'bdb' is Berkeley DB, common db back-end
  - 'shell' and 'perl' are exactly what they sound like, a shell or perl script back-end
  - 'ldb' is a flat-file back-end
  - 'passwd' is a simple wrapper to /etc/passwd
- 'suffix' defines the specific database tree
  - This is used to differentiate queries to the server
  - Very similar to Apache's 'ServerName' directive

# Keywords 'rootdn' and 'rootpw'

- The root distinguished name defines the user that has full access to the database
  - Needed to add/remove other users/entries
  - Protect just like you would a root account!
- Root password is set here
  - We set it using a hash, but can also do plain text
    - Plain text is bad, don't do it!
  - Root password is set using a salted SHA1 hash
  - <http://www.openldap.org/faq/data/cache/347.html>
  - `slappasswd -h {SSHA} -s mycoolpassword`

# Keyword 'index'

- Define searchable attributes
  - All attributes are technically searchable, but this option will make certain ones more quickly searchable
- We set index options on our most common attributes
  - uid, uidNumber, gidNumber, memberUid, loginShell
- We also add in index options 'eq' and 'sub'
  - I *\*think\** they mean 'equality' and 'substring'

# LDAP – Database Access Control

- /etc/ldap/slapd.conf
  - Underneath our previous database definition
  - First-match-wins behavior
- access to attrs=userPassword
  - by dn="cn=manager,dc=cs,dc=colorado,dc=com" write
  - by anonymous auth
  - by self write
  - by \* none
  - Allow people to auth to password, but not read it (except to write), and admin has full access
- access to \*
  - by dn="cn=manager,dc=cs,dc=colorado,dc=com" write
  - by \* read
  - Allow everyone to see everything else

# LDAP – Add Entries to Database

- We must create a structure for our DB
  - Using structural object classes
- Create an 'ldif' file
  - LDAP Data Interchange Format
- All starts with 'dcObject'
  - Defines the top of the tree
- Also must contain the manager entry
  - Gotta have a root user defined somehow!

# LDAP – Create Tree

- I create a temp file /tmp/root.ldif
  - dn: dc=coolname,dc=cs,dc=colorado,dc=edu  
objectclass: dcObject  
objectclass: organization  
o: Coolname @ CU  
dc: coolname  
  
dn: cn=manager,dc=coolname,dc=cs,dc=colorado,dc=edu  
objectclass: organizationalRole  
cn: manager
- I now load this file using the user I configured in slapd.conf
  - First stop slapd: /etc/init.d/slapd stop
- sudo slapadd -v -l /tmp/root.ldif
  - Slapadd modifies files directly

# LDAP – Create Org. Units

- Good to separate people from other things
  - ou=people,dc=coolname,dc=cs,dc=colorado,dc=edu
  - ou=groups,dc=coolname,dc=cs,dc=colorado,dc=edu
- Objectclass 'organizationalUnit', /tmp/groups.ldif
- # Groups, coolname.cs.colorado.edu  
dn: ou=groups,dc=coolname,dc=cs,dc=colorado,dc=edu  
ou: Groups  
objectClass: organizationalUnit
- Add 'groups' ou just like a user:
  - chris@coolname:/tmp\$ **ldapmodify -a -x -D**  
**cn=manager,dc=coolname,dc=cs,dc=colorado,dc=edu -W -f**  
**/tmp/groups.ldif**  
Enter LDAP Password:  
adding new entry  
"ou=groups,dc=coolname,dc=cs,dc=colorado,dc=edu"

# LDAP – Add User to Database

- Also done using an 'ldif' file, /tmp/bob.user.ldif

```
• # Bob, coolname.cs.colorado.edu
dn: uid=bob,ou=people,dc=coolname,dc=cs,dc=colorado,dc=edu
objectClass: top
objectClass: person
objectClass: shadowAccount
objectClass: posixAccount
cn: Bob
sn: Monroe
uid: bob
uidNumber: 1337
gidNumber: 31337
loginShell: /bin/bash
homeDirectory: /home/bob
gecos: Bob 'The Yellow Dart' Monroe
```

## – Now to add:

- ```
chris@coolname:/etc/ldap$ ldapmodify -a -x -D
cn=manager,dc=coolname,dc=cs,dc=colorado,dc=edu -W -f
/tmp/bob.user.ldif
Enter LDAP Password:
adding new entry
"uid=bob,ou=people,dc=coolname,dc=cs,dc=colorado,dc=edu"
```

# LDAP – Create a Group

- Easy to do once the OU is setup
- Create another LDIF file, /tmp/admins.group.ldif
- ```
dn: cn=admin,ou=groups,dc=coolname,dc=cs,dc=colorado,dc=edu
objectClass: posixGroup
cn: admins
gidNumber: 31337
```
- posixGroup requires 'cn' and 'gidNumber'
- Optional 'memberUid' used to add people who have a different default group
- ```
dn: cn=admin,ou=groups,dc=coolname,dc=cs,dc=colorado,dc=edu
objectClass: posixGroup
cn: admins
gidNumber: 31337
memberUid: schenkc
memberUid: cmorebutts
```

# LDAP – Search on New Entry

- Search for our new user bob:
  - ```
chris@coolname:/etc/ldap/schema$ ldapsearch -LLL -x uid=bob
dn: uid=bob,ou=people,dc=coolname,dc=cs,dc=colorado,dc=edu
objectClass: top
objectClass: person
objectClass: shadowAccount
objectClass: posixAccount
cn: Bob
sn: Monroe
uid: bob
uidNumber: 1337
gidNumber: 31337
loginShell: /bin/bash
homeDirectory: /home/bob
gecos: Bob 'The Yellow Dart' Monroe
```
- We have information set, but does it all exist?
  - We're missing a real home directory

# LDAP – Modifying Attributes

- LDIF changes to accommodate edits
- To change bob's information:
  - ```
dn: uid=bob,ou=people,dc=coolname,dc=cs,dc=colorado,dc=edu
changetype: modify
replace: homeDirectory
homeDirectory: /home/administrators/bob
-
replace: gidNumber
gidNumber: 1600
-
replace: loginShell
loginShell: /bin/tcsh
-
```
- 'changetype: modify' is the important part
  - Multiple attributes are separated by hyphens
  - Different DNs can be specified in one LDIF file

# LDAP – Modifying Attributes (cont)

- More than one entry of an attribute can exist
  - memberUid in a group is a great example
- If I want to add or remove a member in a group, I must re-add all existing members:
- ```
dn: cn=admins,ou=groups,dc=coolname,dc=cs,dc=colorado,dc=edu
objectClass: posixGroup
cn: admin
gidNumber: 31337
memberUid: schenkc
memberUid: cmorebutts
memberUid: smithj
```
- If you forget to add existing attributes, they all disappear
  - And you remove people from the group!

# LDAP – Authenticating to LDAP

- Now that we have user information, how do we get our linux machines to look at LDAP?
- Four high-level steps:
  - Install the nss and pam libraries
    - Requires editing of your apt-get sources.list file
  - Edit your PAM config files to also look at ldap
  - Edit your nsswitch.conf file to look at ldap
  - Edit your /etc/libnss\_ldap.conf
- You must be careful, any misconfiguration or typo can severely impact your machine

# LDAP Auth – Install Libraries

- You must uncomment four lines in your `/etc/apt/sources.list` file to enable 'universe' repositories
  - `deb http://us.archive.ubuntu.com/ubuntu/ edgy universe`
  - `deb-src http://us.archive.ubuntu.com/ubuntu/ edgy universe`
  - `deb http://security.ubuntu.com/ubuntu edgy-security universe`
  - `deb-src http://security.ubuntu.com/ubuntu edgy-security universe`
- Once uncommented, run 'apt-get update' to have your machine download list of packages
- Install the packages 'libnss-ldap', 'libpam-ldap'
  - Asks you a number of questions for install
  - <http://mcwhirter.com.au/node/25>

# LDAP Auth – Edit PAM Files

- Always back up your files first before editing!

```
- for i in /etc/pam.d/common-*; do sudo cp $i $i.bak; done
```

- **/etc/pam.d/common-account**

- `account sufficient pam_ldap.so`  
`account required pam_unix.so`

- **/etc/pam.d/common-auth**

- `auth sufficient pam_ldap.so`  
`auth required pam_unix.so nullok_secure use_first_pass`

# LDAP Auth – Edit PAM Files (cont)

- **/etc/pam.d/common-password**

- These are on one line each (no newlines)

- `password required pam_cracklib.so retry=3 minlen=8  
dcredit=0 ucredit=0 lcredit=0 ocredit=0 difok=3`

```
password sufficient pam_unix.so use_authtok remember=2  
nullok md5
```

```
password required pam_ldap.so use_authtok
```

- **/etc/pam.d/common-session**

- `session sufficient pam_ldap.so  
session required pam_unix.so  
session optional pam_foreground.so`

# LDAP Auth – Edit NSS Config

- Name-Service Switch, /etc/nsswitch.conf

- Replace three lines:

```
- passwd:          compat
  group:           compat
  shadow:          compat
```

- With:

```
- passwd:          files ldap
  group:           files ldap
  shadow:          files ldap
```

# LDAP Auth – Edit LDAP Client Conf

- **Identical entries in `/etc/libnss-ldap.conf` and `/etc/pam_ldap.conf`**
- ```
host 127.0.0.1
base dc=coolname,dc=cs,dc=colorado,dc=edu
ldap_version 3
rootbinddn cn=manager,dc=coolname,dc=cs,dc=colorado,dc=edu
pam_password crypt
nss_base_passwd ou=people,dc=coolname,dc=cs,dc=colorado,dc=edu?one
nss_base_shadow ou=people,dc=coolname,dc=cs,dc=colorado,dc=edu?one
nss_base_group ou=group,dc=coolname,dc=cs,dc=colorado,dc=edu?one
```
- **Be sure both files have the same lines**
- **Allow 'root' to edit ldap entries**
  - Add in 'manager' password to `/etc/pam_ldap.secret`

# LDAP Auth – Bug in Ubuntu

- <https://launchpad.net/ubuntu/+source/libnss-ldap/+bug/51315>
- A service called 'udevd' attempts to look up the group name 'nvram' which doesn't exist
  - Simply adding this group to the /etc/group file will do the trick
- You have to do this in Ubuntu 6.10 to do LDAP authentication
  - `% sudo groupadd nvram`
  - If you don't, your machine will hang upon bootup

# LDAP Auth – Extra Notes

- How to test if it works?
  - % id bob
  - % getent passwd bob
  - % getent group admins
- This entire setup is still insecure
  - Extra work needs to be done to SSL-enable queries
  - SSL is essential to remain secure
- Lots of places where things can go wrong
  - Do NOT make typos, double-check everything