

Unix System Administration

Chris Schenk

Lecture 25 – Tuesday Apr 22

CSCI 4113, Spring 2008

Administrivia

- Perl lab is out, due in a week
- LDAP lab in progress
 - Will be released as soon as it is done
 - Due at the time of the final exam
- Final exam released in one week
 - Due 10:00am Tuesday May 6th via email
- Extra Credit
 - If you've done some, email me to make sure I have it down!

Lightweight Directory Access Protocol (LDAP)

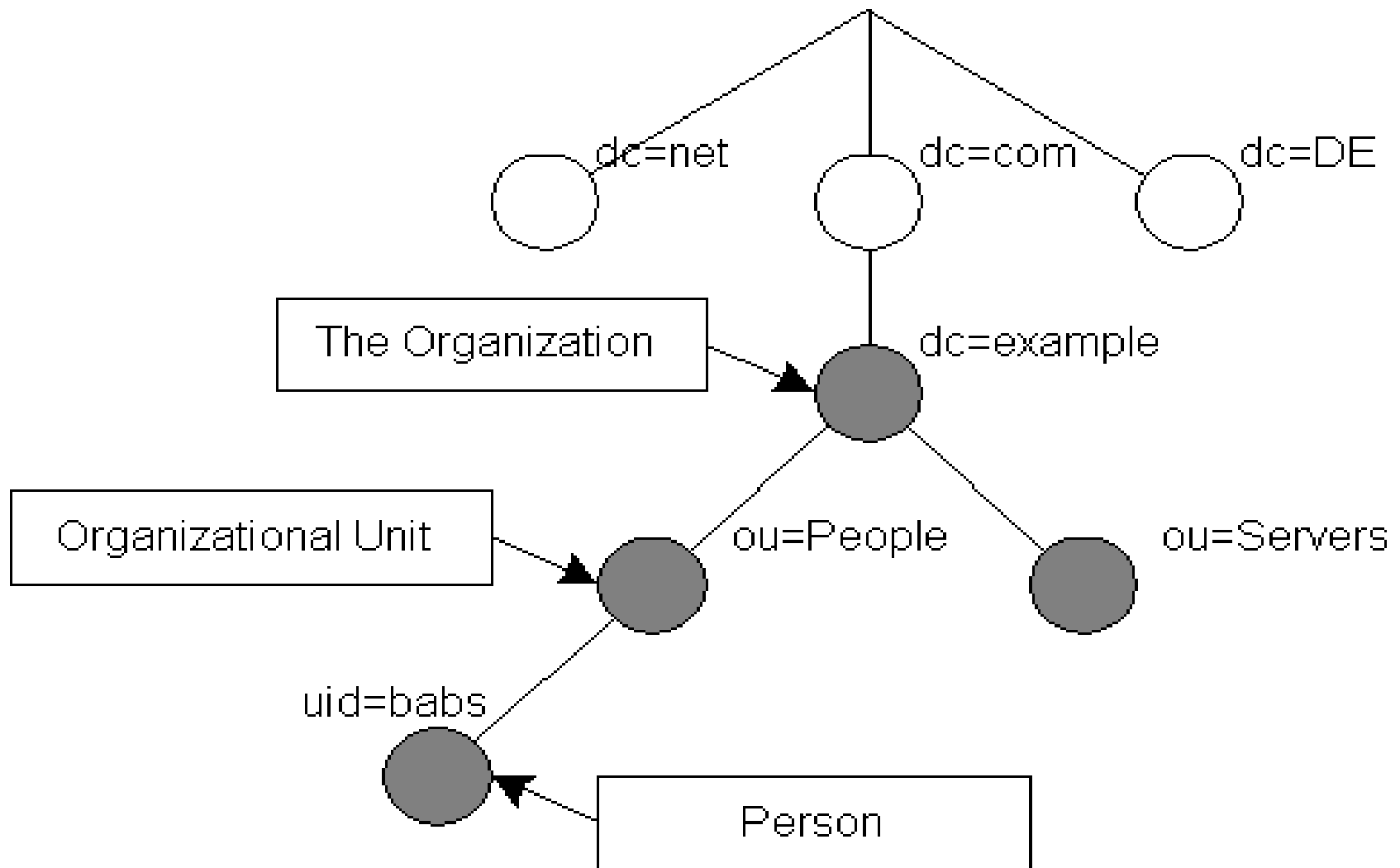
- Maintained at www.openldap.org
- <http://www.openldap.org/doc/admin23/>
 - All slide information pulled from here
- Replaces heavyweight X.500 directories
 - Never seen one myself, but that's what I read about
- A directory that contains searchable information
 - Usually used to store user data
 - Email addresses, contact information
 - Passwords for authentication

What is LDAP?

- A database that contains entries
- Entries have a globally unique id
 - Called a Distinguished Name (DN)
- Attributes have types and one or more values
 - Typically strings
 - “cn” for common name, “mail” for email address, etc
 - Can be binary data
 - JPG image stored in the directory (photo of the person)
- Organized in a tree-structure (similar to DNS)
 - An entry in the tree is a Distinguished Name

LDAP Tree Structure

- DN: uid=babes,ou=People,dc=example,dc=com



LDAP Tree Structure (cont)

- Distinguished Names (RFC 2253)
- CN commonName (common)
- L localityName
- ST stateOrProvinceName
- O organizationName (common)
- OU organizationalUnitName (common)
- C countryName
- STREET streetAddress
- DC domainComponent (common)
- UID userid

What can LDAP do?

- Talk using SSL to encrypt auth functions
 - No sending passwords in the clear!
- Maintain access control
 - Only let people see certain attributes
 - Very similar to DNS views
- Talk to multiple back-end database types
 - Flat files, MySQL, Shell scrips, and others
- Failover support with replicated databases
 - Good to have backups

What Info can LDAP Contain?

- Anything!
 - Like DNS, it's simply a database
 - All key->value pairs for the data
- Schemas exist to define information
 - Specific format, define types and object classes
 - Defaults included under `/etc/ldap/schema`
 - Can write your own schemas (CU does this)
- Nodes in tree must be 'Structural'
 - Acts more like meta-data within the LDAP tree
 - All normal attributes are non-structural

Querying an LDAP database

- **Command-line style to CU ldap**
- `ldapsearch -x -H ldap://directory.colorado.edu -b dc=colorado,dc=edu uid=schenkc`
 - **Searches for my entry in the campus ldap**
- `ldapsearch -x -H ldap://directory.colorado.edu -b dc=colorado,dc=edu uid=schenk*`
 - **Wildcard searches supported (limited)**
- `ldapsearch -x -H ldap://directory.colorado.edu -b dc=colorado,dc=edu "cuEduPersonPrimaryMajor1=COMPUTER SCIENCE"`
 - **Search for all Computer Science majors**

More LDAP Queries (in CSEL)

- `ldapsearch -Zx cn=administrators`
 - Search for all entries that contain `cn=administrators`
- `ldapsearch -Zx '(&(cn=administrators)(objectClass=posixGroup))'`
 - Compound search using binary logic, search for all entries with `cn=administrators` and is a group
- `ldapsearch -Zx '(! (loginShell=/bin/bash))' uid`
 - Get all entries that do not have `loginShell` set to `/bin/bash` and only return the 'uid' attribute
- `ldapsearch -Zx '(&(! (loginShell=/bin/bash)) (! (ou:dn:=UQuotas)))' uid`
 - Same query, but this time omit all entries that belong to `ou=UQuotas`

LDAP Command Line Options

- `ldapsearch [options] [search filter] [attr list]`
 - `Z` – use SSL when talking to the database
 - `x` – Use a simple connection (non-SASL)
 - `D` – authentication DN when binding
 - `b` – search base in the tree
 - `H` – server to talk to
 - `W` – prompt for a password
 - `f` – load input from 'ldif' file
- `[search filter]` – Examples on previous slide
- `[attr list]` – which attributes to return from search

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=cs,dc=colorado,dc=edu"
rootdn "cn=manager,dc=cs,dc=colorado,dc=edu"
rootpw {SSHA}8apbHW80zBUC1CG9FZJBd2Dh2Y1MHrgb
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=cs,dc=colorado,dc=edu  
objectclass: dcObject  
objectclass: organization  
o: Coolname @ CU  
dc: cs  
  
dn: cn=manager,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 – Add User to Database

- Also done using an 'ldif' file

- ```
# Bob, coolname.cs.colorado.edu
dn: uid=bob,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/newuser.ldif
Enter LDAP Password:
adding new entry
"uid=bob,dc=coolname,dc=cs,dc=colorado,dc=edu"
```