

Homework

- 1) Ensure you've taken all required screenshots and put them where requested throughout the lab manual.

Should be 6 screenshots or information copy/pasted

- 2) [From lab] Note that the user id and group ids are quite high. Why would this be the case?

They need to be distinct from the system UID/GIDs

- 3) [From lab] Run the following command. In Q3, describe what it does.

Should return information regarding the schema of our domain

- 4) What is LDAP and how does it compare to Active Directory?

LDAP is an open protocol for querying/modifying items in a directory

AD is a database providing authentication, policies, etc. from Microsoft

AD supports LDAP, but provides much more

<http://stackoverflow.com/questions/663402/what-are-the-differences-between-ldap-and-active-directory>

- 5) What's the difference between a DC and an OU in LDAP? What is their purpose?

OU = organizational unit

DC = domain component

OU is basically higher up in the organizational structure

- 6) What is a Distinguished Name?

Fully qualified name of an object (cn=x,dc=y,dc=z,dc=a)

- 7) Assume we are running an LDAP server for the class. What changes would need to be made to each *client machine* in order for them to authenticate with the LDAP server over a local login? You don't need to provide exact details, but describe the technologies needed. This will require Googling, FYI.

Install client packages, specify where LDAP server is, configure client to use LDAP over local login (/etc/nsswitch.conf in Ubuntu)

<https://www.digitalocean.com/community/tutorials/how-to-authenticate-client-computers-using-ldap-on-an-ubuntu-12-04-vps>

- 8) Modify the script above to create a new user from the command line. Call it user_add.sh. It should accept a single user and password combination as follows (so, no need to loop anymore):

```
$ sudo bash user_add.sh username password
```

Run this and demonstrate that the user has been successfully added (either via phpLDAPAdmin or an ldapsearch query).

Should have the appropriate screenshot, script should look like this (make sure comment block is updated)

```
#!/bin/bash
# Erik Fredericks
# Lab description
```

```
user=$1
userpw=$2
```

```
ldif_file="/tmp/bulk_user.ldif"
dn="cn=admin,dc=csi3670,dc=local"
adminpw="temp12345"
```

```
touch $ldif_file
```

```
uid=2000 # these values should be >= 500
gid=2000
```

```
echo "Adding $user to LDAP directory with UID [$uid] and GID [$gid]"
```

```
echo "dn: uid=$user,ou=People,dc=csi3670,dc=local" >> $ldif_file
echo "objectClass: inetOrgPerson" >> $ldif_file
echo "objectClass: posixAccount" >> $ldif_file
echo "objectClass: shadowAccount" >> $ldif_file
echo "uid: $user" >> $ldif_file
echo "sn: 1" >> $ldif_file
echo "givenName: User" >> $ldif_file
echo "cn: User $i" >> $ldif_file
echo "displayName: User $i" >> $ldif_file
echo "uidNumber: $uid" >> $ldif_file
echo "gidNumber: $gid" >> $ldif_file
```

```
echo "userPassword: $userpw" >> $ldif_file
echo "gecos: User $i" >> $ldif_file
echo "loginShell: /bin/bash" >> $ldif_file
echo "homeDirectory: /home/$user" >> $ldif_file
```

```
cat $ldif_file
```

```
# Add user
```

```
ldapadd -x -D $dn -w temp12345 -a -f $ldif_file
```

```
# Clean up
```

```
rm $ldif_file
```

- a. *For extra credit, make your arguments intelligent and read any number of user/password combinations.*

Probably will use argparse module or some intelligent manner of handling arguments

- 9) Zip up this report, your script(s) and any other materials and submit to Moodle