**CSI3670**

**Winter 2019**

**Lab 4: LDAP**

**Part 4**

**Parts 1 and 2 due Feb 28th at 11:55pm**

**Synopsis:**

In this lab, you will be installing LDAP and playing around with Bash. A nice complement to Part 1.

**LDAP Installation**

1) Login to your GCP Ubuntu partition

LDAP:

$ sudo apt-get install slapdldap-utils

It will ask you for an administrator password. This will be the root credentials of your rootDN of the database (root distinguished name).

The installation set up two directory information trees (DIT). One for slapd-config (config utility for LDAP) and your user-defined information.

Now let’s configure LDAP.

$ sudodpkg-reconfigure slapd

\* Select **No** for the first option (Omit OpenLDAP server configuration)

\* DNS domain name should be csi3670.local

\* Organization name should be your team name

\* Give it the local Administrator password you created when you first installed (probably temp12345)

\* Select MDB database

\* Select Yes for database to be removed when it is purged

\* Yes to Move Old Database

\* No to LDAPv2 if asked

Run the following command to make sure we have the proper config:

$ ldapsearch -x

**Copy and paste the output in Q1**

Now let’s create some entries, a node for our users (People), a node for our groups (Groups), and a node for your project team (<your team name>). Create the following file: **csi3670\_content.ldif**

dn: ou=People,dc=csi3670,dc=local

objectClass: organizationalUnit

ou: People

dn: ou=Groups,dc=csi3670,dc=local

objectClass: organizationalUnit

ou: Groups

dn: cn=<teamname>,ou=Groups,dc=csi3670,dc=local

objectClass: posixGroup

cn: <teamname>

gidNumber: 5000

dn: uid=user1,ou=People,dc=csi3670,dc=local

objectClass: inetOrgPerson

objectClass: posixAccount

objectClass: shadowAccount

uid: user1

sn: 1

givenName: User

cn: User 1

displayName: User 1

uidNumber: 10000

gidNumber: 5000

userPassword: Temp12345

gecos: User 1

loginShell: /bin/bash

homeDirectory: /home/user1

**Note that the user id and group ids are quite high. Why would this be the case? (Q2)**

Add the file to your LDAP directory:

$ ldapadd -x -D cn=admin,dc=csi3670,dc=local -W -f csi3670\_content.ldif

Check to make sure that the data was appropriately added:

$ ldapsearch -x -b dc=csi3670,dc=local 'uid=user1'

**Copy and paste the output in Q1. You should be seeing information about the user you created.**

**Run the following command. In Q3, describe what it does.**

$ ldapsearch -x -b dc=csi3670,dc=local subschemaSubentry

Now let’s install a web interface. We're going to install phpLDAPadmin. You may or may not be aware, but official support for this interface has stopped and doesn't work with PHP7.

Fortunately, somebody out there forked it and cleaned it up.

Go to your /var/www/html directory and clone the repository:

$ sudo git clone https://github.com/breisig/phpLDAPadmin

Navigate into the config directory inside the phpLDAPadmin folder and rename the config.php.example to be config.php.

You **may** need to do the following to open up traffic:

$ sudo service apache2 restart

$ sudoufw allow 80

Your server is at <your ip address>/phpldapadmin.

**Take a screenshot of phpLDAPadmin in your browser and paste it into Q1.**

You can login with your admin credentials. The username is cn=admin,dc=csi3670,dc=local.

**Create a new user account using the web interface. Take a screenshot of your webpage after you’ve done so and put it in Q1.**

**2) Bash fun with LDAP!**

Create a bash script. Call it sample\_user\_add.sh

#!/bin/bash

# Comment block similar to the script from Part 1

users=( "tempuser1" "tempuser2" "tempuser3" )

userpw="temp12345"

ldif\_file="/tmp/bulk\_user.ldif"

dn="cn=admin,dc=csi3670,dc=local"

adminpw="temp12345"

i=0

for user in "${users[@]}"; do

touch $ldif\_file

uid=$(( $i + 1000 ))

gid=$(( $i + 1000 ))

echo $uid $gid

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

i=$(( $i + 1 ))

done

Run it with bash (don’t run it with sh):

$ sudo bash sample\_user\_add.sh

**Take a screenshot or copy/paste the results here:**

**Go back to phpLDAPadmin. Refresh the tree on the left. Take a screenshot showing that tempuser1,tempuser2, and tempuser3 now exist.**

**Homework**

1. Ensure you’ve taken all required screenshots and put them where requested throughout the lab manual.
2. [From lab] Note that the user id and group ids are quite high. Why would this be the case?
3. [From lab] Run the following command. In Q3, describe what it does.
4. What is LDAP and how does it compare to Active Directory?
5. What’s the difference between a DC and an OU in LDAP? What is their purpose?
6. What is a Distinguished Name?
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.
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 anldapsearch query).

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

1. Zip up this report, your script(s) and any other materialsand submit to Moodle