Final Assignment COOS291

Due April 18th, 2024

Your final assignment will involve a Perl script that will be responsible for managing LDAP users on remote systems. You will be required to have two components to your script, a client that access a client running on a remote machine and a server that is running on those remote machines that works on port 10912.

## Server’s Responsibilities

The client will access the remote systems and send one of 4 possible commands to the remote machine. The commands and the server’s response to them are detailed below.

* userlist: If the string “userlist” is sent to the server that it will respond to the client with a comma separated list of all the LDAP users on the server.
* grouplist: If the string “grouplist” is sent to the server, then the server will respond with a comma separated list of the LDAP groups on the server.
* setgroup: If the string “setgroup” is sent to the server, a second string is expected that will identify the name of the group to create any new users in. If this group does not exist, the server should create it. The server should respond to the client with “set” if the group existed, “created” if the group had to be created, or “error” if the group did not exist and could not be created.
* createuser: if the string “createuser” is sent to the server, a second string is expected that will identify the name of the user the client wants to create, and then a third string that will be the password for the user to create. This will create a new LDAP user on the server with the following specifications:
  + The user should use the username and password provided
  + The user should use the group previously set with setgroup, and if setgroup has not been called or was called and failed the user should not be created and the server should send an “error-nogroup” response to the client
  + The new user should have a uid above 5000 and not previously used by any other LDAP user on the server. You may assume uids and gids in the 5000-6000 range are not being used except by your script
  + The newly created user’s home directory should be in /home/[the name of the user’s group]/[the user’s name] and the server is responsible for appropriately creating the directory using /etc/skel and chowning the directory to the right uid and gid. If the directory cannot be created, the server should send an “error-nohome” response to the client
  + If the creation of the LDAP user fails for any other reason, the server should send the client an “error-other” response
  + How you create the LDAP users is up to you – you can create them as local users first and then convert them into ldif format before adding them, you can generate the ldifs from a template, or whatever you wish

In order to be able to fulfil the client’s requests, the server will need to be able to read the /etc/ldap,conf file to retrieve the top level domain components (ie: dc=cst2xx,dc=edu or whatever they may be). Your server will also have to be capable of using and parsing the ldapsearch and ldapadd commands, as well as constructing ldif files as necessary.

## Client Responsibilities

The client will be invoked with the IP address of the server that you want to access. If the remote server cannot be contacted, then the script will exit with a status of 1.

If the remote server is contacted than the client will present a menu from which the user can select the available options described above (note that some options will require additional input). The client will display the responses from the server in a reasonable fashion. The client should also allow for an “exit” option to terminate the connection.

Some restrictions/requirements for both client and server:

* Your script should be well-commented, including a purpose at the top of the script.
* Your script should be readable and follow good coding practices.
* Provide one script that includes both client and server functionality. If the script is invoked with an IP address as the argument, it will operate in client mode, otherwise it will operate as a server.