

CS 448 Final Project

Kai Sleight

Client and Server Communication

As of now the client and server are meant to be run on the same machine, right now being binded to the IP of said host machine. If one desired however, changing the client code can allow the client to connect to the server remotely. With only one client being allowed to connect at a time.

The client and server messages are all encrypted through the use of a fernet key exchanged as soon as the connection is established for all exchanges through the server + client lifecycle. This key is not exchanged securely through any RSA measures, so this is a current vulnerability in the code.

Parser

Every command that can be run on the server can be displayed through typing "help". While I considered making it an option to display the usage tips whenever a user typed in anything that was not a command. For this current state there is not enough to do with commands to justify displaying help.

Commands are parsed in the main server.py file. Right now it just runs in a giant if else statement list. At the end of each command, the server will log the command that (if was logged in) ran, then send back the message to kill whatever operation was running, giving control back to the client.

User Database

The user database is implemented to contain the username + hashed password, using both the username and a randomly generated salt as a salts. Then include the salt itself as well.

An admin can remove or add users. Removing and adding users creates the user given the username and password, and removes or creates, the user's personal directory to store logs or personal files they may wish to create or remove.

A user can login through use of login <username>. Note a user does not have to logout first before logging into another user. That seemed redundant given the implementation.

A user can change their password once their logged in, by typing “change password”.

Data Storage

As of now, storage is a log file that each user automatically gets generated and appended to with everything they type in. They can display this through “display log.txt”. They must be explicit because each user is given their own personal directory, which is where they can eventually create and delete text files. (This may be expanded on Wednesday after my exams). This ontop of the ability for the admin to see other user’s log.

When a user is removed, the entire directory is removed as well.

NOTES:

Please run the myserver.py as admin. As it must create and remove directories. If run on linux, make sure the directory has sufficient permissions to write and read files.

```
$chmod <directory> 777 -R
```

If you want to be safe run the command above.

More might be added tomorrow (Wednesday).

If a directory exists for a user before they are added, the server may crash.