Computer Security Assignment 6 Read Me

NAME: IMRAN SYED
NETID: IMS57
EMAIL: ims57@scarletmail.rutgers.edu

NAME: PEDRO CRUZ
NETID: PC605
EMAIL: pedro.cruztafoya@rutgers.edu

NAME: KRUNAL PATEL
NETID: KP609
EMAIL: kp609@scarletmail.rutgers.edu

NAME: JESSE JAMES
NETID: jcb295
EMAIL: jcb295@scarletmail.rutgers.edu

**Running**

To run the program, use python Server.py and python Client.py.

You will be prompted to sign in with the format USERNAME,PASSWORD.

You can also create your account here through the same format.

After this you can use the commands **GET, POST, END**

Get runs in the format **GET,GROUP** name this will display a list of messages from a group.

Post runs in the format **POST,MESSAGE,GROUP.** It will post your username, message and timestamp to the requested group.

**END** command runs in the format end and simply ends the client-side program.

**Design**

CLIENT:

--> Initiates the connection with the server **(Default address and port number is provided)**

--> Our CLIENT accepts a valid certificate sent from the SERVER. The certificate is verified through SSL library in python.

--> Provides USERNAME AND PASSWORD in the format specified ABOVE

--> If Account exists, it receives all group names.

--> Else it creates a new USERNAME AND PASSWORD in the format specified ABOVE

--> It then receives all group names

--> From this point, it can either use one of three methods (GET, POST, END)

--> GET takes a parameter of group name SAPERATED BY A COMMA and returns all the messages of the group.

--> POST takes two parameters (GROUP_NAME, MESSAGE) SAPERATED BY COMMAS and receives the confirmation of the message being added to the group.

--> END simple ends closes the client side socket and gets out

SERVER:

--> Makes a SLL socket and binds to the default port number **(12345)** and **localhost** address

--> Checks if the files (user.txt and Groups.txt) exist. If they do it retrieves the data else it creates them

--> It adds two Default Groups with no messages in it (FOOD, CLASS)

--> It then listens to connection with SSL SOCKET

--> upon connection request it sends the certificate

--> after certification it prompts user to enter their username and password in the correct format

--> if user exists it sends the group names else it allows user to create a new name and password

--> It then accepts three commands from user. GET,POST, END whose functionalities are specified in the Client section

**--> While creating an account the username and password is hashed through sha512 with a salt string at the end. (user.txt)**

**--> Groups and users are stored securely in separate text files. (Groups.txt)**

**--> !!! The server listens to 5 connections at once. You could change the number 5 if you wish to test with more simultaneous connections!!!**