CSCI 3403 Project 3

Made by Parker Eischen

What I Did: I did everything of this project. My group did not contribute at all and I do not expect them to get any credit for this submission. I was able to implement the sudo SSL that was able to send client's AES key to the server encrypted with the public key. This was then decrypted and signaled back to the client. The client encrypted the username and password through the server's database and the verification string was sent back to the client. In the end this ended up working successfully as shown in the results section of the writeup. The libraries implemented include:

include socket(

from os import chmod

from Crypto.Cipher import AES (encrypted and decrypted AES)

from Crypto.PublicKey import RSA (used to create the public and private key of the server, encryption and decryption as well)

from Crypto import Random (to generate a the randomness of the RSA keys)

import hashlib (used to hash the password)

import uuid

What I Learned: Implementing a server client relationship is a pain. I didn't know all the steps needed to verify a user from the client to the server back to the client. I used to just use libraries that handed all the security for me. Before I cleaned up the code, I had dozens of print statements to verify exactly what each side was sending and receiving. After I got the hang of which variable types needed to be passed through and all the methods available in the RSA key objects, it wasn't too hard to get rolling. It was just the initial learning curve of how to send information between the two that was difficult to grasp. I appreciate client server interaction a lot more now after having to implement it.

Proof of Success: