Client Server Project: Final Iteration

Purpose :

The purpose of this code is to create a server/client socket connection to allow for a password check. The client sends a user input password to the server which checks it against a stored server side password and returns valid or invalid.

Functionality :

- Server
  - Sets up socket communication to allow for a client to connect
  - Uses threading to allow for multiple clients at a time (currently allows 5)
  - Reads a password from a file and stores the information for future reference
  - Receives input from the client:
    - Splits the encrypted password and the hash of the password into two separate variables
    - decrypts the password
    - creates a hash of the decryption to check validity
    - Compares the received hash to the newly created hash and continues if the hashes match
    - Compares the decrypted password to the stored password
  - Writes a valid or invalid response to the clients terminal given whether or not the clients input matches the password
- Client
  - Sets up socket communication to connect to the server
  - Takes user input:
    - encrypts using AES
    - hashes the password using SHA2
    - appends the hash to the encryption
    - sends the total data to the server
  - When it receives the valid or invalid reply from the server, it closes

Security:

There are two security policies on either side of the code. For the client these policies protect against brute force attacks and implements a password length policy. On the server side, it has a similar password policy as well as a time out for the server after 5 clients have disconnected. There is also CUnit testing done on all of these security policies and the documentation for both can be found in their respective directories within the code structure.

Documentation :

As well as this document, there is more documentation provided within the structure of this code. In both the client and server files, there are the results of Splint and Flaw-finder when run on the .c files within the code. These two commands, are used to find issues, security flaws, and possible exploitable or breakable points in code. For the purpose of this project, any results that are a result of a library or common function call are being excused, and only results specifically pertaining to the newly written code are being evaluated. Also included, is a License file in both directories, applying copyright protection to the code as well as a documents that evaluates this codes compliance to PCI DSS standards.