Network programming – Client/Server message relay over TCP protocol

Current Security issues

Currently the client/server model for relaying messages over TCP protocol using Python sockets are not secure.

- There are no encryption/decryption algorithms used to securely transmit messages between client and server.
- There is currently no authentication mechanisms setup to identify a user for the system only by logging into the system by any name.

Mitigation of Security Issues

- Proposal of using MD5 hash algorithm submitted along with each message as a handshake attempt to transmit messages.
- Proposal of un and pw transmission to system prior to using it or an MD5 hash transmission (as above) suffice as the number to which the MD5 hash always reverts to remains the same, so long as the client has the same number as the server and transmits this number as a connection parameter.
- Proposal of an encryption/decryption algorithm.

Implementation of Security Protocols

- Adding the hashlib with md5 python libraries.
- Implementing a function to generate an MD5 hash from the given number.
- Implementing a function to validate the number through MD5 hash generation and perform a match from what client sent and what server has, to allow the message transmission.
- Logging into server with un and pw on client side, client side performs an MD5 hash to pw prior to submission to server, server matches un and md5 hash pw against JSON DB.
- A Fernet encryption/decryption algorithm to put into place to encrypt all messages sent from client or server and decrypt all messages received from client or server.