

# How To Use

## Setup

1. Make sure you have Python 3.5.2 or above already installed on your computer.
2. Tested on GNU/Linux using Python 3.5.2 may work with earlier versions of python but not guaranteed to work.
3. Run client.py on the computer that you intend to send the data file from.
4. Run server.py on the computer that you intend to receive the data on.
5. All interaction with the exception of the dialog prompts will be done through the command line interface prompt.
6. A gui will be used so make sure you have an OS/OS Version that has a way to display a gui.

### client.py

- When you run this program it will prompt you for a host IP address, enter the IP address of the computer you intend to send the file to.
  - If you enter to IP address the host address will default to 127.0.0.1
- You will also be prompted for a port number, choose a number in the high thousands, anything above 1000 is pretty safe.
- After the connection is established you will be asked to enter a username and a password, if you lack the credentials look at the auth.txt doc located with this program.

The way that you add credentials to auth.txt is by following a specific format .

user:password,user2:PassW0rD,user3:NewPassW0dr,user4:ABC123 The username is to the left of the colons and the corresponding password for that user is to the right of the colon. Each username/password pair is separated by a comma. Don't use whitespace. Ex: (user4:ABC123) user4 is the username and ABC123 is the password.

- After successfully signing in the client will have to wait for the server to select whether they wish to apply Base64 Mime Encoding to the data being transmitted or not.
- After that the client will then be presented with two dialogs to select a source file they wish to send to the server and a key file they wish to use to encrypt their data with.
- It should be clear that the key that the client and server use will have to be the same file.
- If unsure what the dialog is asking look at the title of the dialog, I used a title that should make it clear.
  - Once the server has selected the same key file and a destination folder and a filename for the file to be transferred the transfer should begin. Depending on how big the file is the process might take a while. The data chunks are about 1MB in size so it is not the slowest process but the limiting factor on speed it the TCP protocol and how it happens to break up the chunks and the network itself.

### server.py

- When you run this program it will prompt you for a host IP address, enter the IP address of the computer you intend to send the file to.
  - If you enter to IP address the host address will default to 127.0.0.1
- You will also be prompted for a port number, choose a number in the high thousands, anything above 1000 is pretty safe.
- The server will then sit until the client connects to the same port number entered then a connection will be established.
  - The server will also wait for the client to check their credentials, if they are denied entry then the program terminates otherwise the server is then presented with a prompt.
- The server will be asked if they wish to apply Base64 Mime Encoding to the data it will be receiving. If y is selected then the encoding will be applied, any other entered text will be interpreted by the program is no.
- Then the server will be prompted with two dialogs, one to choose where to save the file to be transfered to along for a name for the file and one to choose a key.
  - The Keyfile needs to be the same file used by the client, this is crucial.
  - The file can be of any size as long as it not an empty file as that will cause an error.
- After the files have been chosen the transfer will begin. Depending on factors such as network congestion and how the TCP protocol breaks up the chunks of data into packets it can take a while.
- The server program will be looking for the auth.txt in the server.py local directory, this means you will get an error if you don't store the auth.txt file with the server program, it needs it to validate the user/client.