Sub.py:

At first, run() function, the main function, is executed first. It calls the connect\_mqtt() function. An mqtt client is first created using the random client id by using the command client = mqtt\_client.Client(client

\_id). Then the user is authenticated by setting the username and password. .connect method tries to connect to the broker using the broker address and port number. The result of this connection is handled by the callback function on\_connect. If the connection is successful which means that rc=0, it will show a successful connection. Otherwise, it will show that the connection has failed. Then connect\_mqtt() will return the client and go back to the run() function.

Run() then calls subscribe function using the client received client from connect\_mqtt(). The client subscribes to the public and private topic using the client.subscribe function. Whenever a message is received on\_message call back function is triggered and it prints out the message and the topic name the message is received from.

Client.loop\_forever() maintains the connection to the network as long as possible.

Publish.py:

The process for establishing the connection is the same is the sub.py. After the connection has been established it starts the client.loop\_start() which listens for the messages and sends messages to the broker. Then the publish(client) function is called. It starts a loop which runs forever. Using time.sleep(1) it delays the program by 1 second. Then it generates a message for ssh request to a random client and assigns it to the msg variable. Then it publishes the msg to the topic2 which is the private topic using client.publish method. Client.publish returns a result. If the first element of the result is 0, then it will show the message which was sent along with its topic. Otherwise, it will show that the send has been failed. It will keep on performing these processes until interrupted.