Rigoberto Quiroz

Assignment 1: Email Client

1. Sending Email through interacting with SMTP server from Terminal
   1. Make sure Telnet is enabled in your system.
   2. Establish a connection to the SMTP server, in this case UTEP’s SMTP server  
      hostname: smtp.utep.edu port: 25
      1. IMPORTANT: Make sure that either you are using UTEP’s Wi-Fi or VPN
   3. Once the handshake has been successful, enter ‘helo’ to introduce your system to the server.
      1. Helo utep.edu
   4. Enter the sender and recipient information.
      1. NOTE: Sender information should be your UTEP email address, recipient can be from any other email server (Gmail, yahoo, etc.) UTEP smtp server handles to transportation to recipient’s server

Mail from:<utep email>

Rctp to:<recipient email>

* 1. Once all that is done enter data. This will allow you to enter content of the email (Subject, body)
  2. When done enter ‘.’ This will indicate that the message is over, and the server can proceed to transport the message to recipient server.

A screenshot of a cell phone

Description automatically generated

Server Response

End User request

Email Content

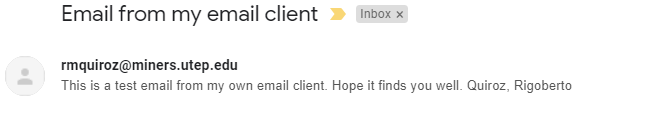
When you we check the recipient email, we should see the following:

Using a recipient email from UTEP (Outlook)

A close up of a logo

Description automatically generated

Using a recipient email from Gmail



1. Writing socket program to automate telnet commands
   1. The first step to create a socket program in python is having the socket library imported
   2. Establish a TCP connection with UTEP’s server

serverName: smtp.utep.edu

serverPortNumber: 25

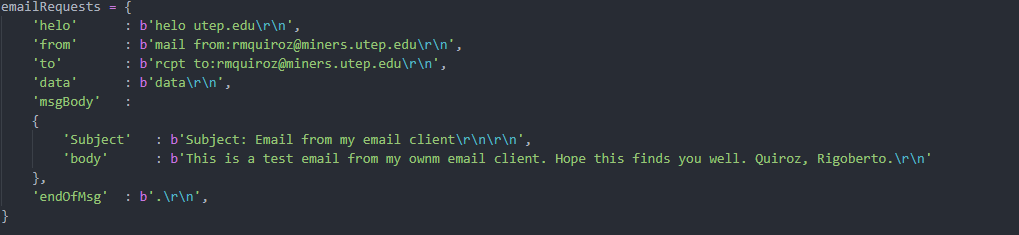


Note:

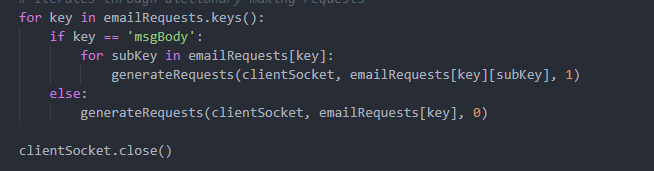
AF\_INET = Your machines IPV4

SOCK\_STREAM = TCP connection

* 1. Once the connection has been established make sure that your requests have ‘\r\n’ and are encoded in bytes.
     1. If the server receives a message without ‘\r\n’ then it will not recognize the message. This is mainly because of the socket in the server.



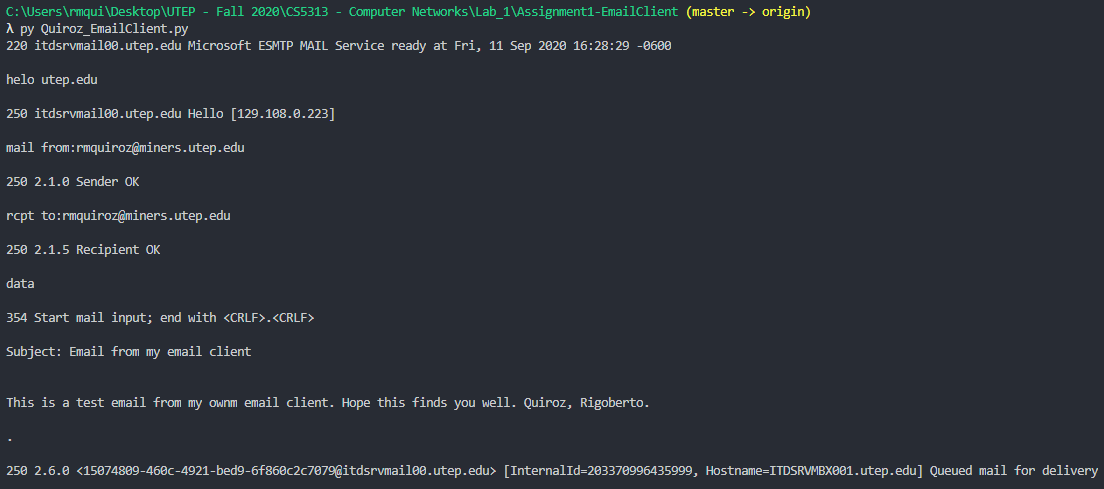
* 1. Once you have sent all the requests to the server and email content it will send that email to its recipient.



Once the loop is done it will close the TCP connection to the server.



Sample output from terminal:



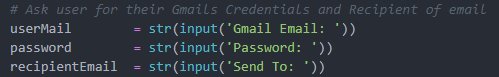
Extra Credit:

Establish a secure connection with Google’s SMTP server and sending an email with one’s email address.

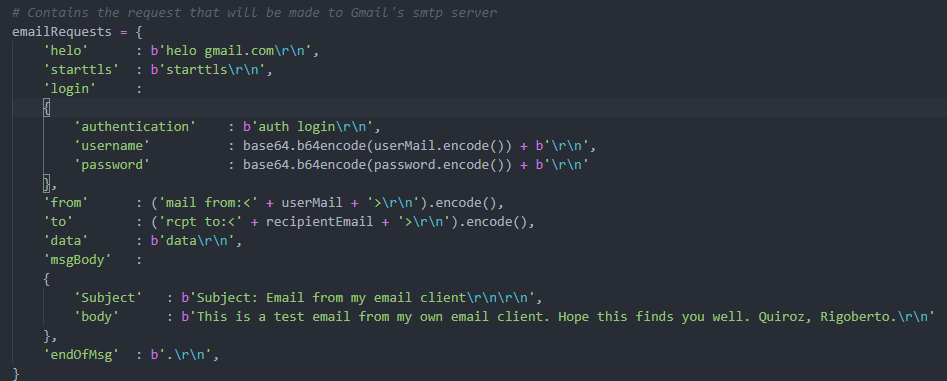
1. We will first need to import some libraries. These libraries will help us communicate with Google’s SMPT server and encode our username and password.



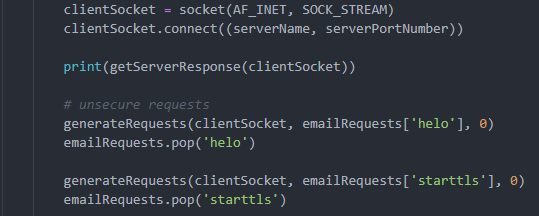
1. Since we will be using your Gmail account to send an email, we will need your credentials, and the person we will send an email to.



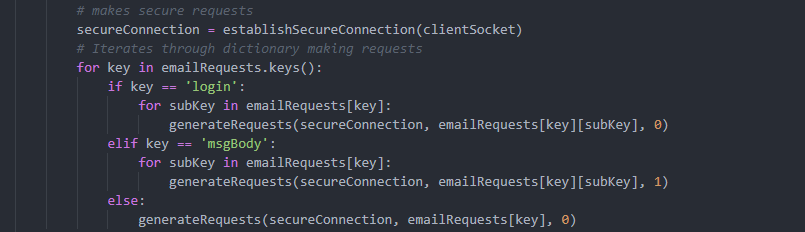
1. Then we can encode and store all our requests in a data structure. This will make it easier to locate and change requests in the future.

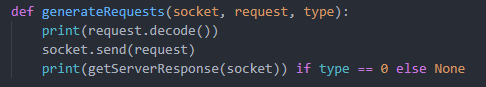


1. Establish a TCP connection to Google’s SMTP server, remember this connection is not secured, so you won’t be able to send emails.



1. Establish a secure connection with SSL certificate, and login with your credentials. After that you can send start sending requests to the server.





Since we stored all requests in the dictionary then the loop will take care of making the requests.

Important Things:

* Starttls – sends requests for authentication, login must be encoded in base64, otherwise server won’t be able to recognize login.
* Base64 – Encryption algorithm