Project 3: Python SMTP Project

Due: 11:59pm, February 9 (Thursday), 2023

(total points: 100)

By the end of this project, you will have acquired a better understanding of SMTP protocol. You will also gain experience in implementing a standard protocol using Python. Your task is to develop a simple mail client that sends email to any recipient. Your client will need to connect to a mail server, dialogue with the mail server using the SMTP protocol, and send an email message to the mail server. *Python provides a module, called smtplib, which has built in methods to send mail using SMTP protocol. However, we will not be using this module in this project, because it hides the details of SMTP and socket programming.* In order to limit spam, some mail servers do not accept TCP connection from arbitrary sources. For the experiment described below, you may want to try connecting `mail.egr.msu.edu` mail server.

Question 1: [60 points]

Below you will find the skeleton code for the mail client. You are to complete the skeleton code. The places where you need to fill in code are marked with #Fill in start and #Fill in end. Each place may require one or more lines of code.

Additional Note: In some cases, the receiving mail server might classify your e-mail as junk. Make sure you check the junk/spam folder when you look for the e-mail sent from your client.

```
#import socket module
from socket import *
msg = "\r\n I love computer networks!"
endmsg = "\r\n.\r\n"

# Choose a mail server (e.g. Google mail server) and call it mailserver
mailserver = mail.egr.msu.edu
mailport = 25

# Create socket called clientSocket and establish a TCP connection with mailserver
#Fill in start

#Fill in end

recv = clientSocket.recv(1024).decode()
print(recv)
```

```
if recv[:3] != ’220’:
    print(’220 reply not received from server.’)

# Send HELO command and print server response.
#Fill in start

#Fill in end
recv1 = clientSocket.recv(1024).decode()
print(recv1)
if recv1[:3] != ’250’:
    print(’250 reply not received from server.’)

# Send MAIL FROM command and print server response.
# Fill in start

# Fill in end

# Send RCPT TO command and print server
# response. [replace "xxxx" with a valid account]
rcptto = "RCPT TO: <xxxx@egr.msu.edu> \r\n"
# Fill in start

# Fill in end

# Send DATA command and print server response.
# Fill in start

# Fill in end

# Send message data.
# Fill in start

# Fill in end

# Message ends with a single period.
# Fill in start

# Fill in end

# Send QUIT command and get server response.
# Fill in start

# Fill in end
```

Your tasks: Please read below for instructions.

- Please complete the above code and make sure it can be used to send email to an egr.msu.edu account. It may not be able to send email to outside account (e.g., @msu.edu, @gmail.com) because of the security measures deployed on mail.egr.msu.edu server.
- Add the screen shots of terminal to your report to show that you successfully send an email to another egr.msu.edu account. Provide a description and explanation as needed.

Question 2: [40 points]

SMTP protocol with Port 25 does not require authentication and thus may cause many security issues. Therefore, many mail servers do not support access via port 25 any more. Instead, they use port 587 to connect with clients in a secure manner. Mail servers like `mail.egr.msu.edu` and `mail2.cse.msu.edu` require your client to add a Transport Layer Security (TLS) or Secure Sockets Layer (SSL) for authentication and security reasons. Without authentication, those servers do not allow you to send emails outside their domains. To add TLS security into your mail client, the below code can be used as a reference.

```python
# Send STARTTLS command to server and print server response
command = "STARTTLS\r\n"
clientSocket.send(command)
recv = clientSocket.recv(1024)
print("START TLS: ", recv)
if recv[:3] != '220':
    print ('220 reply not received from server.')

# log in
scs = ssl.wrap_socket(clientSocket, ssl_version=ssl.PROTOCOL_SSLv23)
```

Your tasks: Please read below for instructions.

- Modify your Python code from Question 1 so that it can send email on your `egr.msu.edu` account via port 587. To do so, you need to add the secure socket connection. the above code can be used as your reference. You do not have to use the STARTTLS command. You are good as long as you can send emails in your `egr.msu.edu` email account via port 587. Through the port 587 and with the security implementation, your client should be able to send email to any email servers (e.g., @gmail.com, @msu.edu).
- Add the screen shots of terminal, console, or others to your report to show that you successfully complete the Python code. Provide a describe and explanation as needed.

What to Hand in

You need to submit two Python code files (`YourLastName_SMTP_client_port25.py` and `YourLastName_SMTP_client_port587.py`) and a SINGLE PDF report.