CS 315: Computer Networks Lab
Spring 2023-24, IIT Dharwad
Assignment-8
Socket Programming: SMTP
February 29, 2024

Lab Instructions

- Please leave your bags on the Iron shelf near the SP16 entrance.
- Login to the Ubuntu OS on your machine. The login credentials are as follows:
  - Username: user
  - Password: 123456
- Mark your attendance in the attendance sheet before leaving the lab.
- Handle the lab resources with utmost care.
- Please go through the following exercises in today's lab.
- It is recommended that you complete all the following exercises during the lab slot itself.
- If you face any difficulties, please feel free to seek help online or from your peers or TAs.
- After finishing all exercises, please carry your solutions with you (via email/pen drive) for future reference, and delete the files from the desktop.

Introduction

By the end of this lab, 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 lab, because it hide 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 both to your university mail server and to a popular Webmail server, such as an AOL mail server. You may also try making your connection both from your home and from your university campus.

Code

Below you will find the skeleton code for the 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 Notes

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.

Skeleton Python Code for the Mail Client

```
#add in prompt
userEmail = "smtplab23@gmail.com"
userPassword = "lmvgusmmhxkmzoti"
userDestinationEmail = input("Enter Email Destination: ")
userSubject = input("Enter Subject: ")
userBody = input("Enter Message: ")
msg = '{}.\r\n I love computer networks!'.format(userBody)

# Choose a mail server (e.g. Google mail server) and call it mailserver
#Fill in start

#Fill in end

# 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.
heloCommand = 'HELO Alice\r\n'
clientSocket.send(heloCommand.encode())
recv1 = clientSocket.recv(1024).decode()
print(recv1)
if recv1[:3] != '250':
      print('250 reply not received from server.')

#account authentication
clientSocket.send("STARTTLS\r\n".encode())
clientSocket.recv(1024)
sslClientSocket = ssl.wrap_socket(clientSocket)
sslClientSocket.send("AUTH LOGIN\r\n".encode())
print(sslClientSocket.recv(1024))
sslClientSocket.send(b64encode(userEmail.encode()) + "\r\n".encode())
print(sslClientSocket.recv(1024))
```

```
sslClientSocket.send(b64encode(userPassword.encode()) + "\r\n".encode())
print(sslClientSocket.recv(1024))

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

#Fill in end

# Send RCPT TO command and print server response.
#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
```

Submission Details

- Submit the client code with your roll number prefixed as <Roll_number>_client.py
- Screenshots at the recipient inbox proving the successful mail delivery.