NTNU
Norwegian University of
Science and Technology
Department of Telematics

TTM4100

Communication – Services and Networks

Programming lab 3: “SMTP Client”

There are three programming labs in this course, and you have to do at least two of them. The exercises are designed to focus on learning and understanding – not debugging code. The labs are good exercise for the upcoming project, as they cover some of the same tasks. All lab-answers should be delivered on itslearning for review. Any questions about the exercises can be posted on the forum.

Deadline of submission: 07.02.2016

Lab 3: SMTP Lab

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 hides the details of SMTP and socket programming.

In order to limit spam and malicious behavior, most mail servers do not accept TCP connections from arbitrary sources (these "open mail relays" are quite scarce nowadays). Hence, the simple interaction you learnt in the textbook/theory classes (which was used in the early Internet) will not work in standard mail servers, as you'll need to authenticate in order to use the SMTP protocol and send an email.

Luckily for us, the old NTNU's student mail server, although "officially dead", is still up and running. It is not exactly an open mail relay, but the original SMTP interaction you have learned in theory classes will work as long as you are connected to NTNU/eduroam networks (either physically in the campus or through VPN).

N.B. It is strongly recommended to read **Sect. 2.7.2** of the textbook before (and while) doing this programming lab.

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. The same code can also be found in the file "Skeleton SMTPClient.py".

Additional Notes

As pointed out before, make sure that you are connected to NTNU/eduroam (through VPN if you are outside the campus) before trying out the code. This is because this SMTP server is not exactly an open mail relay. It allows you to send emails without authentication as long as you are in a secure network (NTNU/eduroam), assuming that the fact that you are using that network makes you a non-malicious user.

The mail server address is *smtp.stud.ntnu.no*. The port, you'll have to find by yourselves.

If, even after *smtp.stud.ntnu.no* commits to deliver the message, you cannot see the email in your mailbox; do not panic. If you try to send to a recipient outside NTNU domain (gmail, yahoo, etc…), your email will most likely be blocked. It won't even appear on the spam folder, it will simply be rejected by the receiving mail server. This is intentionally done by some ISPs and mail servers to block spam and malicious behavior, and does NOT mean your code is wrong. As long as you receive the correct reply code from *smtp.stud.ntnu.no* when sending the email, your code is correct: you email has been sent and received. The fact that this receiving mail server will accept the email after receiving it is a different matter, in which other protocols in addition to SMTP are involved.

If you still want to see how sending an email works nowadays, you can try the optional exercise. It is not straightforward and there is little information about it in the Internet, so ask if you get stuck.

What to Hand in

In your submission, you are to provide the complete code for your SMTP mail client as well as a screenshot showing that you indeed receive the e-mail message. If, as explained before, you are not receiving the email in your mailbox, but you are getting the correct reply codes from *smtp.stud.ntnu.no*, a screenshot of the received replies is sufficient.

Skeleton Python Code for the Mail Client

The same code can also be found in the file "Skeleton SMTPClient.py".

```
from socket import *

# Message to send
msg = "\r\n I love computer networks!"
endmsg = "\r\n.\r\n"

# Our mail server is smtp.stud.ntnu.no
mailserver = 'smtp.stud.ntnu.no'

# Create socket called clientSocket and establish a TCP connection
# (use the appropriate port) with mailserver
# FILL IN START

# FILL IN END

recv = clientSocket.recv(1024)
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)
recv1 = clientSocket.recv(1024)
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.
# 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
```

Optional Exercises

1. Mail servers like office365 for your student accounts (address: smtp.office365.com, port: 587) requires your client to add a Transport Layer Security (TLS) or Secure Sockets Layer (SSL) for authentication and security reasons, before you send MAIL FROM command. Add TLS/SSL commands to your existing ones and implement your client using office365 mail server at above address and port.