# Module Five:
# SMTP with Python

## Grading Criteria

*Total Points: 20 points*

***In this lab, you will be writing a simple dictionary-based password cracker. Hopefully, this helps illustrate how scary easy it can be to use a dictionary to crack simple passwords.***

☐ **[2.5 points]** Your submission is labeled as "cyb404_module05_lab00_[nau_id]_[lastname]_[firstname].zip." For example, if I were to submit a file, it would be labeled as cs460_module14_lab00_mv668_vigil-hayes_morgan.zip. **FAILURE TO COMPLY WITH THIS STEP CAN LEAD TO A ZERO GRADE.**

☐ **[2.5 points]** Your submission files are in the correct formats as specified.

- ☐ [1 points] Password cracker code is all contained in a file called smtp_spammer.py
- ☐ [1.5 points] There is a comment for every functional unit of code that details the purpose of the code.

☐ **[15 points]** Code correctly takes in a Google username, password, and target email address and sends an email to that address.

- ☐ [2.5 points] File runs using command: python3 smtp_spammer.py -t [target email] -l [Gmail login] -p [Gmail password]
- ☐ [2.5 points] There are no runtime errors
- ☐ [5 points] Screenshot of all SMTP packets generated by your SMTP SPAM mailer.
- ☐ [5 points] Screenshot of an email sent by your SMTP SPAM mailer to your NAU or personal inbox.

## Part 1: Create an automatic SPAM emailer

Start by reading through the smtplib API to get an understanding of how it works.

https://docs.python.org/3/library/smtplib.html

The process of sending an email normally involves opening one's client of choice, clicking new, and then clicking send. Behind the scenes, the client connects to the server, possibly logs in, and exchanges information detailing the sender, recipient, and the other necessary data. The Python library, smtplib, will handle this process in your program. You will create a Python email

client to use to send SPAM emails to your target. This client will be very basic but will make sending emails simpler for Module 13.

The starter script is comprised of two functions, sendMail() and main(). sendMail() is a function that takes in parameters like a destination address, subject, and message and then uses the smtlib package to craft and send an email. You will follow the #TODO comments in the starter script to help identify which functionality you need to implement using the smtplib package. Main() is a simple function that parses commandline arguments. You can modify some of the variables in main to create more compelling subjects and messages.

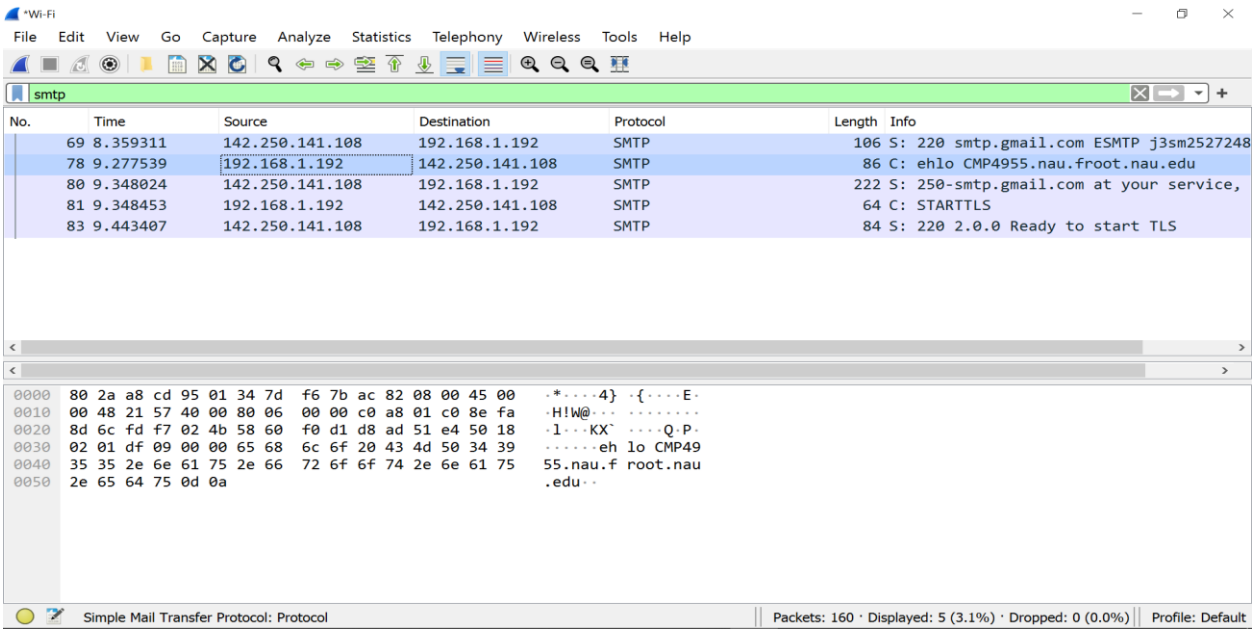
For this lab, you will use the Google Gmail SMTP server; you will need to create a Google Gmail account to use this script or modify the settings. You can test email functionality by running a local SMTP debugging server, using the smtpd module that comes pre-installed with Python. Rather than sending emails to the specified address, it discards them and prints their content to the console. Running a local debugging server means it's not necessary to deal with encryption of messages or use credentials to log in to an email server.

You can start a local SMTP debugging server by typing the following

```
python -m smtpd -c DebuggingServer -n localhost:1025
```

When you have your SMTP SPAM mailer working, turn on Wireshark and capture packets being sent from your mailer. Add a screenshot of the captured SMTP packets below:

[SCREENSHOT GOES HERE]



Try sending an email to your NAU account or your personal email. Take a screenshot of the email sent by the SMTP SPAM mailer as it appears in your inbox:

[SCREENSHOT GOES HERE]