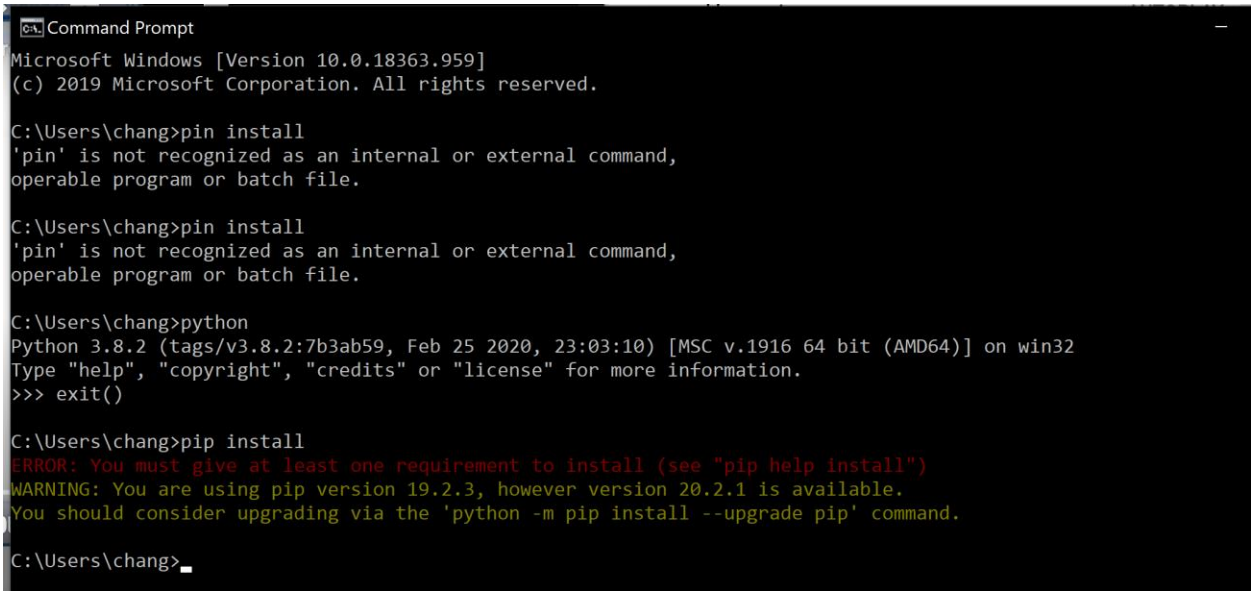
Name: Chang Pao Herr
Student ID: 19544
Course Term: Summer 2020 – CS535 - NETWORK SECURITY FUNDAMENTAL
Instructor: Dr. Chang Henry

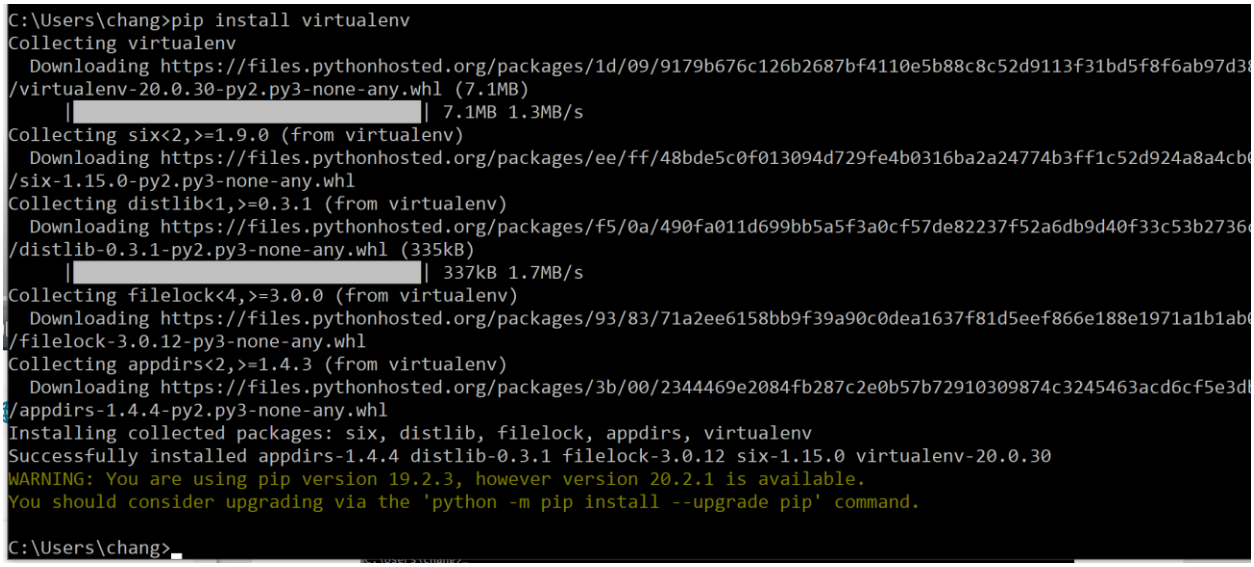
Week# 11 Homework#: 9
Due Date: 7/28/2020 11:30:00 PM
Homework Subject: Project HTTPS (II)

Part#2 Question #4: Symmetric Key Crypto is secured but difficult to exchange keys

- References
  - Part 2: Symmetric Key Crypto is secured but difficult to exchange keys
  - Answers - 2020 Summer

Finally, found this video to guide me install python environment so I can work in command prompt shell. https://www.youtube.com/watch?v=Yk0hcKb2Jxc





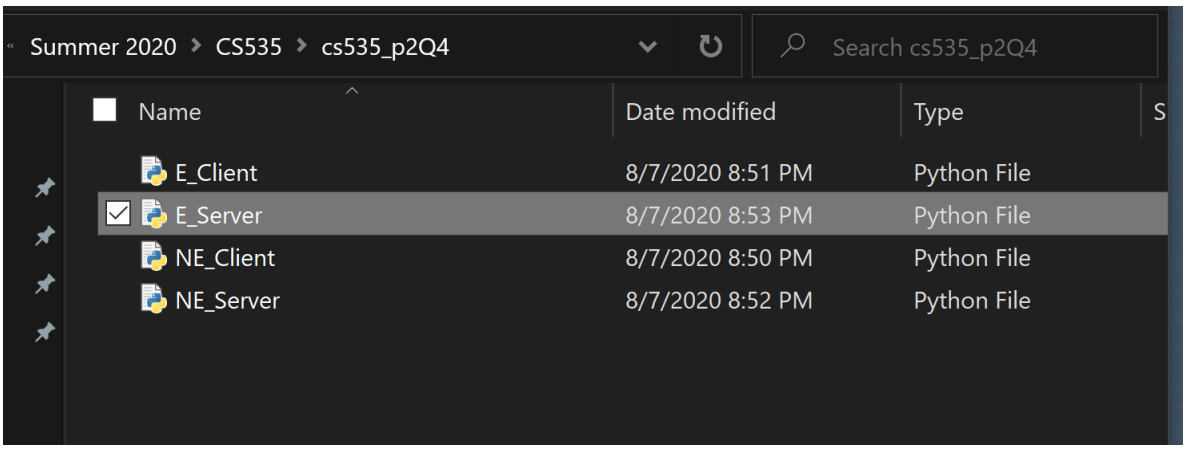
The server.py have some error, message stated ass below.

```
C:\Users\chang>cd desktop

C:\Users\chang\Desktop>MSEE Program
'MSEE' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\chang\Desktop>MSEE Program
'MSEE' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\chang\Desktop>cd MSEE Program

C:\Users\chang\Desktop\MSEE Program>cd Summer 2020

C:\Users\chang\Desktop\MSEE Program\Summer 2020>cd CS535

C:\Users\chang\Desktop\MSEE Program\Summer 2020\CS535>cd cs535_p2Q4

C:\Users\chang\Desktop\MSEE Program\Summer 2020\CS535\cs535_p2Q4>python E_Server.py
Traceback (most recent call last):
  File "E_Server.py", line 3, in <module>
    from flask import Flask
ModuleNotFoundError: No module named 'flask'
```

Install Cryptography

```
C:\WINDOWS\system32>pip install cryptography
Collecting cryptography
  Downloading https://files.pythonhosted.org/packages/00/fc/ed8cf3e3d3817707c11da167a3478f9cb834afed5e8af450516752b
/cryptography-3.0-cp38-cp38-win_amd64.whl (1.5MB)
     |████████████████████████████████| 1.5MB 2.2MB/s
Collecting cffi!=1.11.3,>=1.8 (from cryptography)
  Downloading https://files.pythonhosted.org/packages/40/ad/eb98b5ec6129ffdabedca218ded2c529d59b935dac7cc6108366e37
/cffi-1.14.1-cp38-cp38-win_amd64.whl (178kB)
     |████████████████████████████████| 184kB ...
Requirement already satisfied: six>=1.4.1 in c:\users\chang\appdata\local\programs\python\python38\lib\site-package
om cryptography) (1.15.0)
Collecting pycparser (from cffi!=1.11.3,>=1.8->cryptography)
  Downloading https://files.pythonhosted.org/packages/ae/e7/d9c3a176ca4b02024debf82342dab36efadfc5776f9c8db077e8f6e
/pycparser-2.20-py2.py3-none-any.whl (112kB)
     |████████████████████████████████| 112kB ...
Installing collected packages: pycparser, cffi, cryptography
Successfully installed cffi-1.14.1 cryptography-3.0 pycparser-2.20
WARNING: You are using pip version 19.2.3, however version 20.2.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\WINDOWS\system32>_
```

Create Server.py and Client.py file.



Encrypted Server file: E_Server.py

E_Server.py - C:\Users\chang\Desktop\MSEE Program\Summer 2020\CS535\cs535_p2Q4\E_S...

File Edit Format Run Options Window Help

```
# symmetric_server.py
import os
from flask import Flask
from cryptography.fernet import Fernet

SECRET_KEY = os.environb[b"SECRET_KEY"]
SECRET_MESSAGE = b"fluffy tail"
app = Flask(__name__)

my_cipher = Fernet(SECRET_KEY)

@app.route("/")
def get_secret_message():
    return my_cipher.encrypt(SECRET_MESSAGE)
```

Encrypted Client file: E_Client.py

E_Client.py - C:\Users\chang\Desktop\MSEE Program\Summer 2020\CS535\cs535_p2Q4\E_C...

File Edit Format Run Options Window Help

```
# symmetric_client.py
import os
import requests
from cryptography.fernet import Fernet

SECRET_KEY = os.environb[b"SECRET_KEY"]
my_cipher = Fernet(SECRET_KEY)

def get_secret_message():
    response = requests.get("http://127.0.0.1:5683")

    decrypted_message = my_cipher.decrypt(response.content)
    print(f"The codeword is: {decrypted_message}")

if __name__ == "__main__":
    get_secret_message()
```

Execute Cryptography. Fernet : key

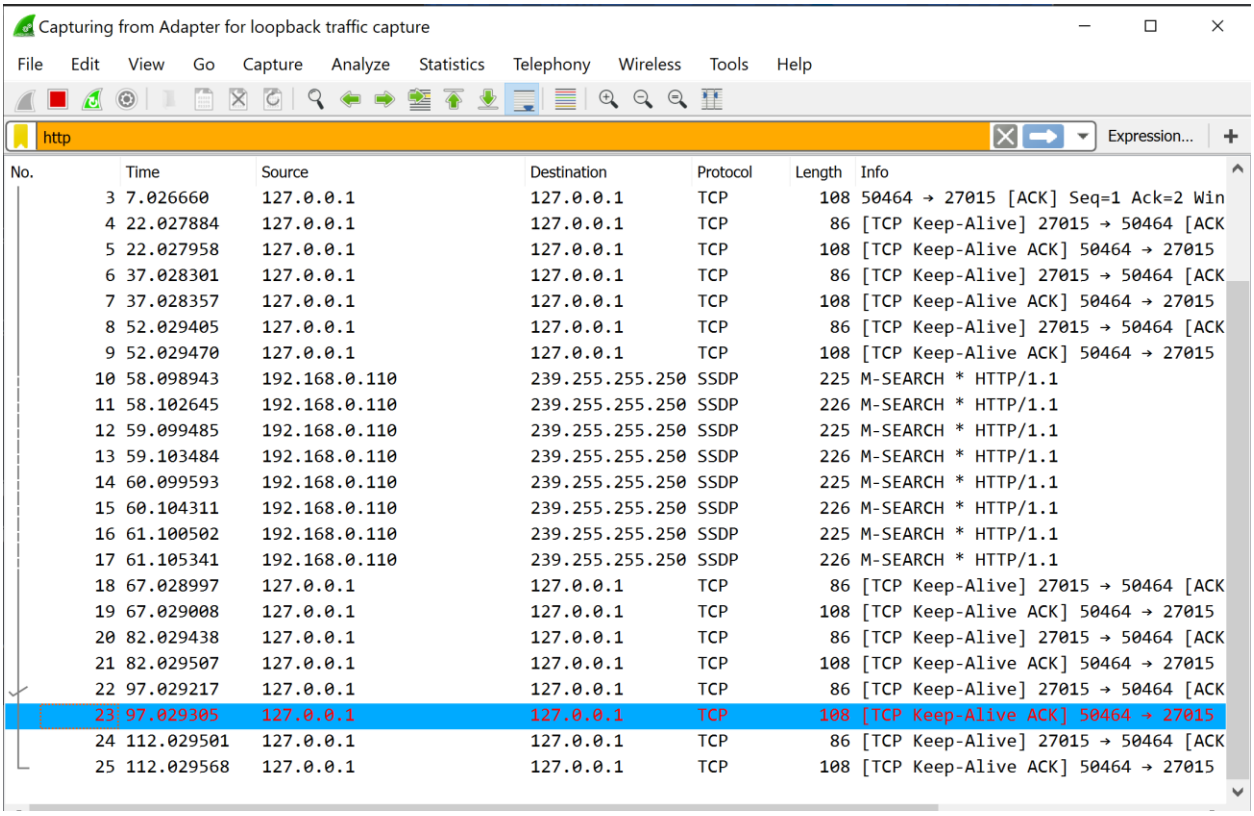
```
C:\WINDOWS\system32>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from cryptography.fernet import Fernet
>>> key = Fernet.generate_key()
>>> key
b'6Vz5wghLq2EUhZnLkOs1EH4-GEF6_-Zl_wAnQRC6X5k='
>>> b'8jtTR9QcD-k3RO9Pcd5ePgmTu_itJQt9WKQPzqjrcoM='
```

Encrypt the message: “fluffy tail”

```
>>>
>>> my_cipher = Fernet(key)
>>> ciphertext = my_cipher.encrypt(b"fluffy tail")
>>> ciphertext
b'gAAAAABfLlh8p1T6NDWI3zdoOlLiRy5c_bTwBjDp9E2b1vGp_aUTdonXyby4fj29LRaGaDBsqHgTPfxypjBF5Nv5k1dUEebzAw=='
>>>
```

Notes:

As for the second part of capture the message in Wireshark, I cannot capture this because my laptop cannot capture HTTP due to my router setting not allow. I was unable to getting to work capturing the message in HTTP, I do see the activities is capturing in TCP and SSDP. Next screen shows the traffic on TCP/SSDP.


Capturing from Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Expression...
No. Time Source Destination Protocol Length Info
3 7.026660 127.0.0.1 127.0.0.1 TCP 108 50464 → 27015 [ACK] Seq=1 Ack=2 Win
4 22.027884 127.0.0.1 127.0.0.1 TCP 86 [TCP Keep-Alive] 27015 → 50464 [ACK
5 22.027958 127.0.0.1 127.0.0.1 TCP 108 [TCP Keep-Alive ACK] 50464 → 27015
6 37.028301 127.0.0.1 127.0.0.1 TCP 86 [TCP Keep-Alive] 27015 → 50464 [ACK
7 37.028357 127.0.0.1 127.0.0.1 TCP 108 [TCP Keep-Alive ACK] 50464 → 27015
8 52.029405 127.0.0.1 127.0.0.1 TCP 86 [TCP Keep-Alive] 27015 → 50464 [ACK
9 52.029470 127.0.0.1 127.0.0.1 TCP 108 [TCP Keep-Alive ACK] 50464 → 27015
10 58.098943 192.168.0.110 239.255.255.250 SSDP 225 M-SEARCH * HTTP/1.1
11 58.102645 192.168.0.110 239.255.255.250 SSDP 226 M-SEARCH * HTTP/1.1
12 59.099485 192.168.0.110 239.255.255.250 SSDP 225 M-SEARCH * HTTP/1.1
13 59.103484 192.168.0.110 239.255.255.250 SSDP 226 M-SEARCH * HTTP/1.1
14 60.099593 192.168.0.110 239.255.255.250 SSDP 225 M-SEARCH * HTTP/1.1
15 60.104311 192.168.0.110 239.255.255.250 SSDP 226 M-SEARCH * HTTP/1.1
16 61.100502 192.168.0.110 239.255.255.250 SSDP 225 M-SEARCH * HTTP/1.1
17 61.105341 192.168.0.110 239.255.255.250 SSDP 226 M-SEARCH * HTTP/1.1
18 67.028997 127.0.0.1 127.0.0.1 TCP 86 [TCP Keep-Alive] 27015 → 50464 [ACK
19 67.029008 127.0.0.1 127.0.0.1 TCP 108 [TCP Keep-Alive ACK] 50464 → 27015
20 82.029438 127.0.0.1 127.0.0.1 TCP 86 [TCP Keep-Alive] 27015 → 50464 [ACK
21 82.029507 127.0.0.1 127.0.0.1 TCP 108 [TCP Keep-Alive ACK] 50464 → 27015
22 97.029217 127.0.0.1 127.0.0.1 TCP 86 [TCP Keep-Alive] 27015 → 50464 [ACK
23 97.029305 127.0.0.1 127.0.0.1 TCP 108 [TCP Keep-Alive ACK] 50464 → 27015
24 112.029501 127.0.0.1 127.0.0.1 TCP 86 [TCP Keep-Alive] 27015 → 50464 [ACK
25 112.029568 127.0.0.1 127.0.0.1 TCP 108 [TCP Keep-Alive ACK] 50464 → 27015