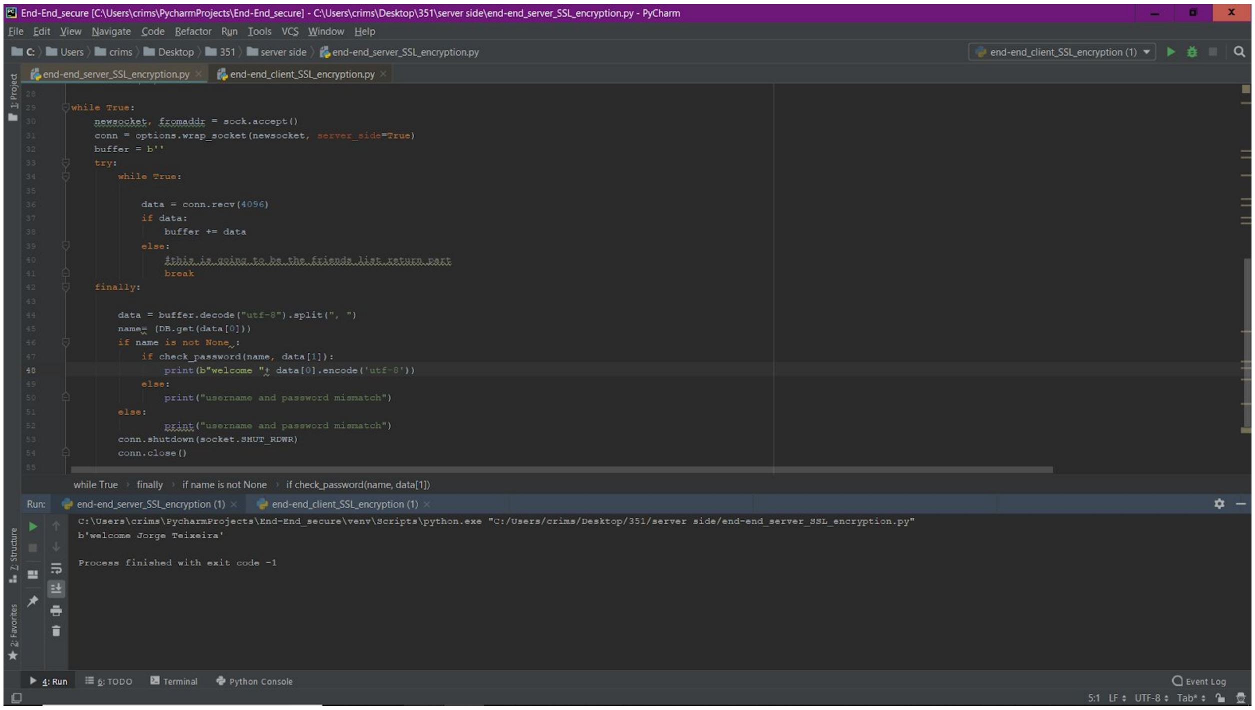
Jorge Teixeira
31385227

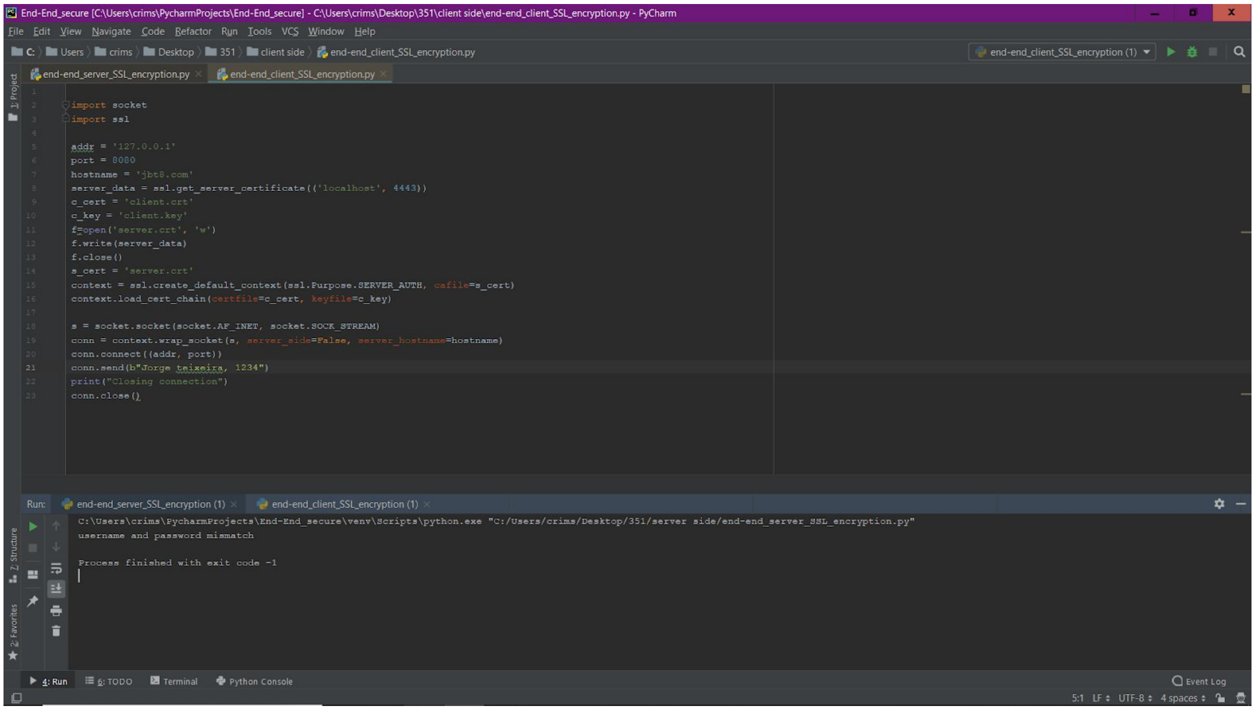
So far here is everything I have done. currently the things I have are
1) SSL certificates for clientA and server
2)private keys for the server and clientA
3)salted/hashed passwords in a hardcoded db
4)SSL three way handshake
5)user authentication by using a hashing method that works (check sources for website)
6)server authentication(through ssl certificate)
7) cipher list and support
For the SSL certificates/private keys I followed a tutorial online which helped me create both through the linux command terminal. Once there I started on my SSL/TLS three way handshake this proved to be much easier and quite tricky at the same time. I managed to get the SSL encryption working but there is a small hiccup. Since I have not purchased the SSL through a trusted source if I try to verify the SSL certificate it will error out and exit the code since it was "self created". Other than that it does go through cipher lists and it chooses one with Server priority but if you would like I can change that to client priority very quickly. Additionally they create the symmetric key to decode all messages. The Server certificate is the only other hurdle so far because I am not able to have it in one document. There are currently two documents for the server. One is to host the https which the client calls and gets the certificate from and the other is the server itself this of course will be fixed as I further work on the project. From here the user name and password are sent from the client to the server and the server is able to check if the client has an account through a hard-coded DB (I emailed you about this but your answer was very confusing so I left it how it currently is but I do have a SQL DB ready to use if you want me to change it just in case). I feel the need to say that the password is not in plaintext in my code and in fact it has been hashed and salted using some code that I found online as a starting point and have modified to be unique and more original (check sources for more details). The given password is then salted and hashed and then compared to the non plain text password to see if they are a match and if it is then a message will confirm the user and if not a failed message will be displayed (pictures below). I have documented all of my sources below just to make sure there is no plagiarism in my work and if there are any questions feel free to email me.

Screenshots of it working


End-End_secure [C:\Users\crims\PycharmProjects\End-End_secure] - C:\Users\crims\Desktop\351\server side\end-end_server_SSL_encryption.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
C: Users crims Desktop 351 server side end-end_server_SSL_encryption.py
end-end_client_SSL_encryption (1)
end-end_server_SSL_encryption.py
end-end_client_SSL_encryption.py
1: Project
while True
finally
if name is not None
if check_password(name, data[1])
Run:
end-end_server_SSL_encryption (1)
end-end_client_SSL_encryption (1)
Z: Structure
2: Favorites
4: Run
6: TODO
Terminal
Python Console
Event Log
5:1 LF UTF-8 Tab*

```python

while True:
    newsocket, fromaddr = sock.accept()
    conn = options.wrap_socket(newsocket, server_side=True)
    buffer = b''
    try:
        while True:

            data = conn.recv(4096)
            if data:
                buffer += data
            else:
                #this is going to be the friends list return part
                break
    finally:

        data = buffer.decode("utf-8").split(", ")
        name= (DB.get(data[0]))
        if name is not None :
            if check_password(name, data[1]):
                print(b"welcome "+ data[0].encode('utf-8'))
            else:
                print("username and password mismatch")
        else:
                print("username and password mismatch")
        conn.shutdown(socket.SHUT_RDWR)
        conn.close()

```

```
C:\Users\crims\PycharmProjects\End-End_secure\venv\Scripts\python.exe "C:/Users/crims/Desktop/351/server side/end-end_server_SSL_encryption.py"
b'welcome Jorge Teixeira'

Process finished with exit code -1
```


End-End_secure [C:\Users\crims\PycharmProjects\End-End_secure] - C:\Users\crims\Desktop\351\client side\end-end_client_SSL_encryption.py - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
C: Users crims Desktop 351 client side end-end_client_SSL_encryption.py
end-end_client_SSL_encryption (1)
end-end_server_SSL_encryption.py
end-end_client_SSL_encryption.py
1: Project
Run:
end-end_server_SSL_encryption (1)
end-end_client_SSL_encryption (1)
Z: Structure
2: Favorites
4: Run
6: TODO
Terminal
Python Console
Event Log
5:1 LF UTF-8 4 spaces

```python

import socket
import ssl

addr = '127.0.0.1'
port = 8080
hostname = 'jbt8.com'
server_data = ssl.get_server_certificate(('localhost', 4443))
c_cert = 'client.crt'
c_key = 'client.key'
f=open('server.crt', 'w')
f.write(server_data)
f.close()
s_cert = 'server.crt'
context = ssl.create_default_context(ssl.Purpose.SERVER_AUTH, cafile=s_cert)
context.load_cert_chain(certfile=c_cert, keyfile=c_key)

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn = context.wrap_socket(s, server_side=False, server_hostname=hostname)
conn.connect((addr, port))
conn.send(b"Jorge teixeira, 1234")
print("Closing connection")
conn.close()
```

```
C:\Users\crims\PycharmProjects\End-End_secure\venv\Scripts\python.exe "C:/Users/crims/Desktop/351/server side/end-end_server_SSL_encryption.py"
username and password mismatch

Process finished with exit code -1
```

Sources

How to set up key agreement protocol-
https://sublimerobots.com/2015/01/simple-diffie-hellman-example-python/

How to set up SSL -
https://www.electricmonk.nl/log/2018/06/02/ssl-tls-client-certificate-verification-with-python-v3-4-sslcontext/

salted passwords -
https://stackoverflow.com/questions/9594125/salt-and-hash-a-password-in-python

server preferences and how the ciphers worked -
https://www.programcreek.com/python/example/65033/OpenSSL.SSL.OP_NO_SSLv3

general functions and options - https://docs.python.org/3/library/ssl.html

hwo to open and write to a file -
https://www.guru99.com/reading-and-writing-files-in-python.html

to make sure i signed my certificates correctly -
http://pankajmalhotra.com/Simple-HTTPS-Server-In-Python-Using-Self-Signed-Certs

examples for how to get my certificates over hostnames -
https://www.programcreek.com/python/example/62606/ssl.get_server_certificate

certificate handling - https://docs.python.org/2/library/ssl.html

non secure tcp connection - https://pymotw.com/3/socket/tcp.html

ssl handshake -
https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_7.1.0/com.ibm.mq.doc/sy10660_.htm

how to decode byte to string -
https://stackoverflow.com/questions/606191/convert-bytes-to-a-string

more salting examples - https://www.vitoshacademy.com/hashing-passwords-in-python/