COMPUTER NETWORK

TOPIC-PROXY FIREWALL

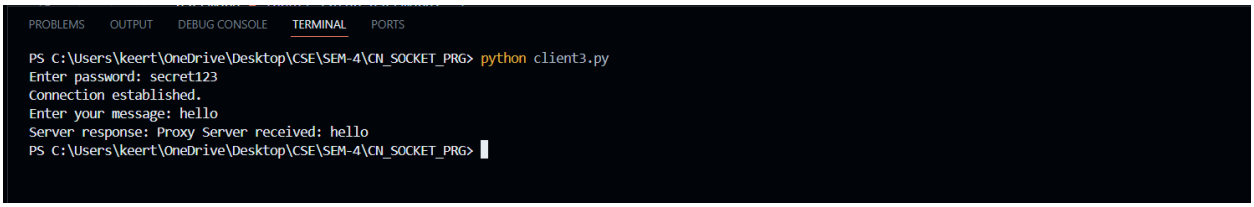
Name :

Keerthan P.V - PES2UG23CS272

M Niranjan – PES2UG23CS308

This project includes two servers: a **Secure Proxy Server** and a **Plain Server**. The Secure Proxy Server uses SSL to protect communication and checks if the client provides the correct password (token) before allowing access. If the client is authorized, they can send a message, and the server replies back. The Plain Server, on the other hand, runs without security and simply receives and responds to messages. The client program can connect to either server based on a setting, allowing users to test both secure and non-secure communication.

1.Client3 conecting to secure proxy server:



Server side of secure proxy server:

```
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python proxy_server_ssl.py
>>
Secure Proxy Firewall Server is running on port 8443...
[Proxy] Client says: hello
```

2.client is blocked due to incorrect password:

```
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python client3.py
Enter password: secret
Incorrect password. You are blocked.
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG>
```

3.client is connecting to plain server:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python client3.py
Server response: Plain Server received: Hello from Plain Client3!
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG>
```

Server side of plain server:

```
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python plain_server.py
Plain Server is running on port 8888...
[Plain] Connection from ('127.0.0.1', 64847)
[Plain] Client says: Hello from Plain Client3!
```

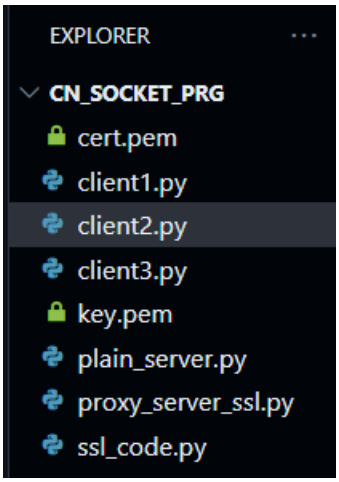
3.Multiclients sending message to secure proxy server:

```
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python client1.py
Enter password: secret123
Connection established.
Enter your message: hello
Server response: Proxy Server received: hello
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python client2.py
Enter password: secret123
Connection established.
Enter your message: hi
Server response: Proxy Server received: hi
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python client3.py
Enter password: secret123
Connection established.
Enter your message: cn project
Server response: Proxy Server received: cn project
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG>
```

```
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python proxy_server_ssl.py
>>
Secure Proxy Firewall Server is running on port 8443...
[Proxy] Client says: hello
[Proxy] Client says: hi
[Proxy] Client says: cn project
```

4.Multiclients connecting to plaine server :

```
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python client1.py
Server response: Plain Server received: Hello from Plain Client1!
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python client2.py
Server response: Plain Server received: Hello from Plain Client2!
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG> python client3.py
Server response: Plain Server received: Hello from Plain Client3!
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-4\CN_SOCKET_PRG>
```



The cert.pem file contains the server's public SSL certificate, which is used to prove the server's identity to clients, while the key.pem file holds the private key, which is used to securely encrypt and decrypt data during communication. Together, these files enable secure (SSL/TLS) connections in Python socket programs, ensuring that data exchanged between the client and server is encrypted and protected from eavesdropping.