Projet Développer Chatbox App Sécurité

Viet NGUYEN -- L3B -- 20006303

L'Objectif est de créer une application de chat de base avec une interface simple utilisant la bibliothèque '**tkinter**' et un serveur utilisant '**socket**' avec les idées pour développer des systèmes de sécurité supplémentaires suivants (Les idées 2, 3, 4 sont facltatives) :

1) Crypter les messages avant de les envoyer:

- Utilisez des algorithmes de chiffrement tels que AES ou RSA pour chiffrer les messages avant de les envoyer du client au serveur et les décrypter lors de la réception des messages.
- Il doit y avoir un mécanisme pour échanger en toute sécurité les clés de chiffrement entre le client et le serveur.

2) Cryptage de connexion (optional):

- Utilisez SSL/TLS pour crypter la connexion entre le client et le serveur. Cela garanti que les messages ne sont pas capturés et lus en ligne.

3) Authentification d'utilisateur (optional):

- Ajoutez un système de connexion qui permet l'authentification des utilisateurs. Cela fournit une couche de sécurité supplémentaire et permet d'empecher d'autres utilisateurs malveillants.

4) Stockage des messages (optional):

- Archiver des messages, assurer qu'ils sont cryptés avant de les enregistrer dans la base de données.

---

1)

- J'utilise un projet SmallChatAppPython sur mon GitHub avec 2 fichiers **serveur.py** et **client.py** disponibles.
- Installer 'pycryptodome':

```
viet@vietLaptop:~$ pip3 install pycryptodome
```

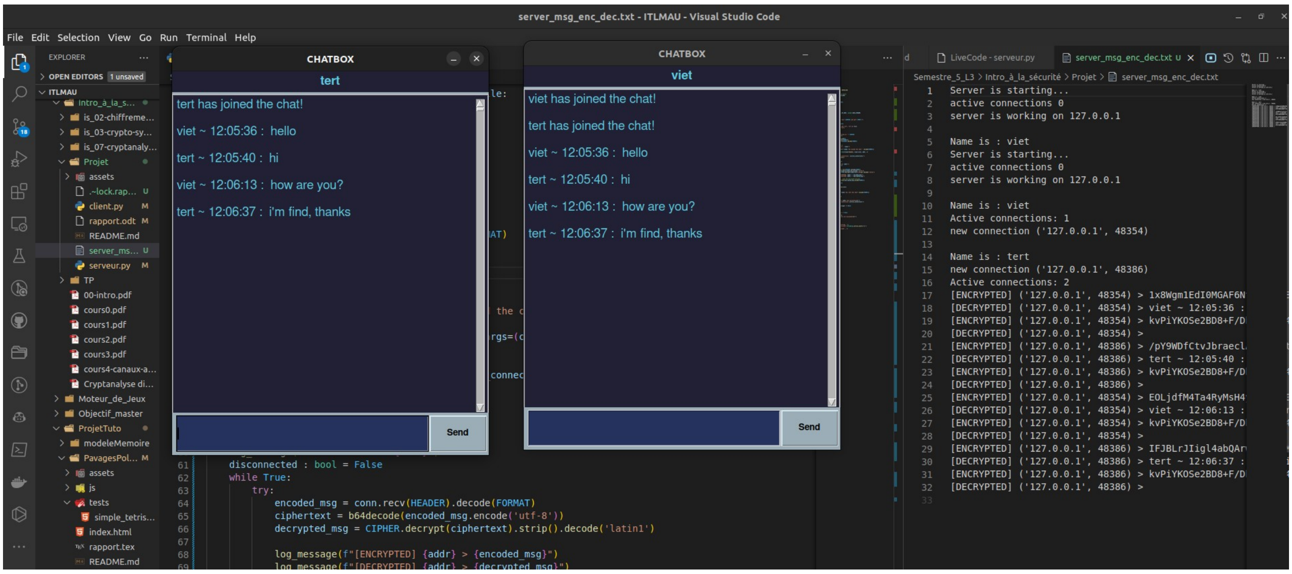
- Importer dans les fichiers **serveur.py** et **client.py**:

```
from Crypto.Cipher import AES
from base64 import b64encode, b64decode

# AES
KEY = b'0123456789abcdef'  # 16 bytes
CIPHER = AES.new(KEY, AES.MODE_ECB)
```

- Il existe 2 algo que j'envisagerais d'utiliser pour ce projet AES (Advanced Encryption Standard) et RSA (Rivest-Shamir-Adleman). Cependant, RSA n'est pas adapté au chiffrement de données volumineuses car il est lent et nécessite plus de mémoire qu'AES. En plus, AES est l'un des algo de chiffrement les plus populaires et les plus fiables, et il largement accepté dans le monde entier. ==> On va utiliser AES !

- Ici j'ai utilisé le mode ECB d'AES, mais ce n'est pas le mode le plus sécurisé. Cependent, ce n'est qu'un projet d'étude et la premier fois que j'utilise AES donc je vais commencer par les base :))



- J'ai créer une fonction supplémentaires pour créer un journal des messages pour **serveur.py** : "**server_msg_enc_dec.txt**", il contiendra tous les messages, l'encodage et le décodages.