Projet final - RAT

Vous devez réaliser un système de RAT (Remote Administration Tool) exclusivement en Python. Il doit répondre aux exigences suivantes. Il est composé de deux éléments : **un serveur et un client**. Les deux programmes sont à développer en Python.

Le projet est à réaliser par groupe de 2 **UNIQUEMENT** !

Le rendu du projet est le **dimanche 31 juillet 2025 à 23h59**. Vous devez rendre le code ainsi qu'une vidéo prouvant le bon fonctionnement du projet.

Exigences client (10 points)

- Le client doit communiquer avec le serveur à l'aide d'une socket TCP **chiffrée et sécurisée**.
- Il doit être opérationnel sur les systèmes **Windows** et **Linux**.
- Il doit embarquer les fonctionnalités suivantes :
  - `help` : afficher la liste des commandes disponibles. **(0.5 point)**
  - `download` : récupération de fichiers de la victime vers le serveur. **(1 point)**
  - `upload` : récupération de fichiers du serveur vers la victime. **(1 point)**
  - `shell` : ouvrir un shell (bash ou cmd) interactif. **(0.5 point)**
  - `ipconfig` : obtenir la configuration réseau de la machine victime. **(0.5 point)**
  - `screenshot` : prendre une capture d'écran de la machine victime. **(1 point)**
  - `search` : rechercher un fichier sur la machine victime. **(0.5 point)**
  - `hashdump` : récupérer la base SAM ou le fichier shadow de la machine en fonction de l'OS. **(1 point)**
  - `keylogger` : enregistrer les frappes clavier de la victime. **(1 point)**
  - `webcam_snapshot` : prendre une photo à l'aide de la webcam de la victime. **(1 point)**
  - `webcam_stream` : diffuser en direct le flux vidéo de la webcam de la victime. **(1 point)**
  - `record_audio` : enregistrer l'audio à l'aide du micro de la victime. **(1 point)**

Exigences serveur (6 points)

- Il doit agir à travers une interface interactive lorsque l'agent rentre en contact avec le serveur. **(1 point)**

Exemple :

```
[*] Listening on 8888...
[+] Agent received !
rat > Taper votre commande ici
```

- Le serveur est en écoute sur un port TCP. **(1 point)**
- Le serveur doit être en mesure d'accepter plusieurs agents en parallèle. **(2 points)**

```
rat >
[+] Agent received !
rat > sessions
[*] Agent 1
[*] Agent 2
rat > interact agent1
rat agent 1 > Taper la commande pour l'agent 1
```

- Le serveur doit être capable de gérer les connexions et déconnexions des agents. **(1 point)**
- Le serveur doit être capable de gérer les erreurs de commande et d'afficher un help des commandes disponibles. **(1 point)**

Notation

Généralités

- Qualité du code (fonctions, classes, organisation, etc)
- Fonctionnement du code
- Respect du cahier des charges
- Commentaires au sein du code
- Présence de tests unitaires
- Bonnes pratiques de développement (classes, fonctions, etc)
- Utilisation de `poetry` pour la gestion des dépendances
- Utilisation de `precommit` pour le formatage du code
- Utilisation de `logger` pour les logs
- Utilisation de `pytest` pour les tests unitaires
- Présence d'un `README.md` expliquant le fonctionnement du projet
- Présence d'une vidéo de démonstration du projet

Bonus facultatif

/!\ Uniquement accessible si les fonctionnalités obligatoires sont présentes et fonctionnelles. /!\

- Toute fonctionnalité supplémentaire rapporte **+1 point**.
  - Interface web ? **+1 point**
  - Ou interface graphique ? **+1 point**
  - Autres fonctionnalités ? **+1 point**
- Utilisation de `docker` pour le déploiement du projet : **+1 point**.
- Contournement d'antivirus : **+2 points**.

Malus

- Absence de chiffrement : **-2 points**
- Absence de vidéo : **-2 points**
- Absence de tests unitaires : **-2 points**
- Absence d'utilisation de `poetry` : **-1 point**
- Absence de `README.md` : **-1 point**
- Absence d'utilisation de precommit : **-1 point**

- Utilisation de `logger` à la place de `print` : **-1 point**
- Absence d'utilisation de `Git` : **-1 point**

Règles

Votre code est analysé dynamiquement par un programme automatique permettant de détecter le partage. Toute tentative de triche amenera à une note de **01/20** pour l'ensemble du ou des groupes.