

Rapport solution la Roue Romaine

1. Établissement de la connexion (Socket)

Le script initie une connexion TCP brute (Raw TCP) via la bibliothèque socket.

- **Famille d'adresses** : AF_INET (IPv4).
- **Type de socket** : SOCK_STREAM (TCP). Une fois le "Handshake" TCP effectué avec le serveur (52021), un buffer de données est récupéré (recv) pour obtenir l'énoncé.

2. Analyse Syntaxique (Parsing)

Le serveur envoie une phrase contenant du "bruit" (texte inutile) autour de la donnée cible (ex: *"My string is 'URYYB'..."*). Pour isoler la chaîne chiffrée de manière robuste, nous utilisons une **Expression Régulière (Regex)** :

- Pattern : `r"(.*)"`
- Ce motif capture tout contenu situé entre deux guillemets simples, peu importe la longueur ou le contenu du texte environnant.

3. Algorithme de Déchiffrement (ROT13)

La chaîne extraite est chiffrée avec **ROT13** (Rotate by 13 places).

- C'est un chiffre de substitution monoalphabétique symétrique.
- L'alphabet est décalé de 13 positions ($A \leftrightarrow N$, $B \leftrightarrow O$, etc.).
- **Implémentation** : Plutôt que de recoder une boucle de substitution manuelle, nous utilisons la bibliothèque standard codecs de Python qui gère cette transformation nativement et efficacement.

4. Protocole de Réponse

Pour que le serveur valide la réponse, deux conditions techniques sont respectées lors de l'envoi :

1. **Encodage** : La chaîne Python (Unicode) est convertie en bytes (UTF-8) avant d'être mise sur le réseau.
2. **Délimiteur** : Un caractère de saut de ligne (`\n`) est concaténé à la réponse. Sans ce caractère, le serveur attendrait la suite du message jusqu'au timeout, causant l'échec du challenge.