Un travail pas terminé !

Auteur	Catégorie	Niveau
Cécile	Crypto	Facile

Énoncé :

Intro

Un membre de notre équipe avait commencé à écrire un script pour déchiffrer le fichier message-chiffre.txt de notre jeu de piste, il n'était, semble t'il, pas loin mais n'avait pas terminé. Peux-tu nous aider à le finir ?
Le format du flag est NHM2I{messagedechiffre}.

Pièce Jointe

- Un fichier chiffrement.py contenant le code suivant :

```
import string

def encryption(plaintext) :
        texte_chiffre = ""
        for lettre in plaintext :
                if lettre.isalpha():
                        if lettre.isupper():
                                rang_lettre = ord(lettre) - ord('A')
                                lettre_chiffree = chr((25 - rang_lettre) % 26 + ord('A'))
                        else:
                                rang_lettre = ord(lettre) - ord('a')
                                lettre_chiffree = chr((25 - rang_lettre) % 26 + ord('a'))
                else:
                        lettre_chiffree = " "

                texte_chiffre += lettre_chiffree
        return texte_chiffre

def main() :
        plaintext = "C0URAGE!"
        ciphertext = encryption(plaintext)
        print (ciphertext)

main()
```

- Un fichier message-chiffre.txt contenant :

```
Zgy4hs
```

Préparation du challenge

Objectif

Comprendre la logique du chiffrement et voir les lignes manquantes pour pouvoir déchiffrer. Les outils comme decode ne fonctionnent pas pour identifier le type de chiffrement.

Flag

NHM2I{Atb4sh}

Démarche

Faire un script en Atbash qui fonctionne, qui gère la ponctuation et les nombres. Enlever les lignes qui gèrent ces 2 particularités. Mettre un indice avec un message court qui ne permet pas d'utiliser des outils comme l'outil de reconnaissance de code de dcode.

Résolution :

Solution débutant :

Essayer le solveur de code, voir que ça marche pas.
Essayer de comprendre le script.

Voir que les chiffres et la ponctuation sont manquants, voir comment le gérer en python.
Une fois qu'on a compris que c'est du Atbash, il suffit d'utiliser de nouveau le code pour déchiffrer le message.

Script corrigé :

```
def encryption(plaintext) :
        texte_chiffre = ""
        for lettre in plaintext :
                if lettre.isalpha():
                        if lettre.isupper():
                                rang_lettre = ord(lettre) - ord('A')
                                lettre_chiffree = chr((25 - rang_lettre) % 26 + ord('A'))
```

```
                else:
                        rang_lettre = ord(lettre) - ord('a')
                                lettre_chiffree = chr((25 - rang_lettre) % 26 + ord('a'))
                else:
                        if lettre in string.punctuation + string.digits:
                                lettre_chiffree = lettre
                        else:
                                lettre_chiffree = " "
                texte_chiffre += lettre_chiffree
        return texte_chiffre

def main() :
        plaintext = "C0URAGE!"
        ciphertext = encryption(plaintext)
        print (ciphertext)
        plaintextUn = "Atb4sh"
        ciphertextUn = encryption(plaintextUn)
        print (ciphertextUn)
        plaintextUn = encryption(ciphertextUn)
        print (plaintextUn)

main()
```