korp_station

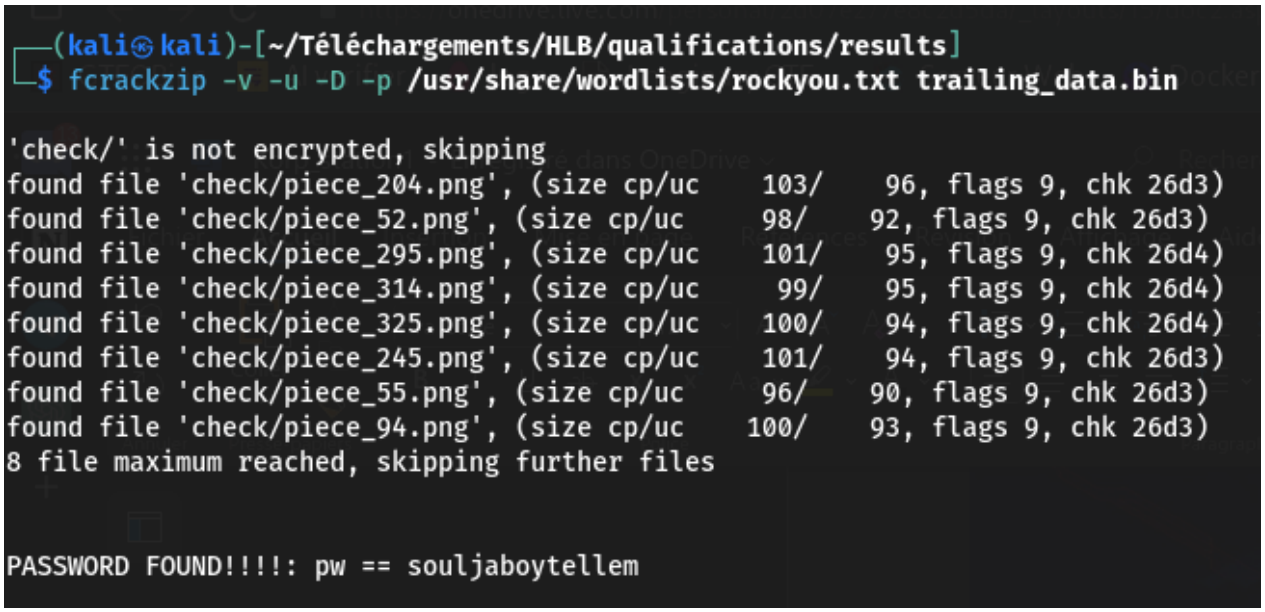
Writeup pour bjICS Tragedy: Suspicious file 1

On nous donne une image.

Bon le challenge est normalement un défi osint mais franchement y’a 0 osint dans ma méthode de résolution.

J’applique mon outil favorite stegoveritas sur l’image et je me retrouve avec un fichier zip trailing_data.bin. Le fichier est crypté et aucun mot de passe n’avait été spécifié dans la description du challenge.

Je passe alors fcrackzip pour cracker le zip.



Je me retrouve avec un dossier check contenant 340 parties d'un code qr. Bon là avec l'aide de smtg j'ai écrit un script python pour rassembler les parties du code qr.

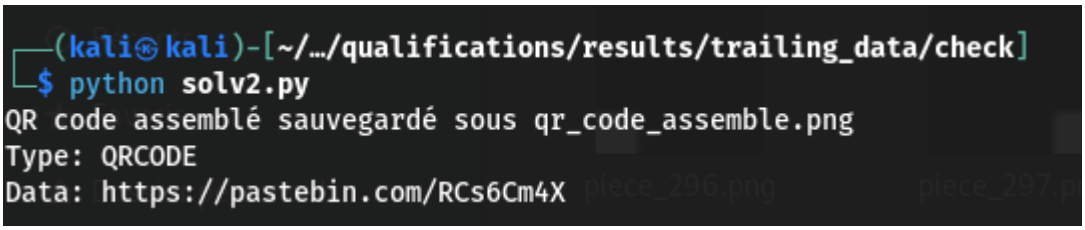
Voici mon script de résolution

```
from PIL import Image
import os
from pyzbar.pyzbar import decode

def assemble_qr_code(input_folder, output_file, total_pieces):
    # Liste pour stocker toutes les pièces
    parts = []

    # Chargement de toutes les pièces
    for i in range(total_pieces):
        filename = f"piece_{i}.png"
```

```
        file_path = os.path.join(input_folder, filename)
        if os.path.exists(file_path):
            parts.append(Image.open(file_path))
        else:
            print(f"Attention : {filename} non trouvé")

    if not parts:
        print("Aucune pièce trouvée. Vérifiez le dossier d'entrée.")
        return

    # Dimensions de chaque partie (supposons qu'elles sont toutes identiques)
    part_width, part_height = parts[0].size

    # Calcul des dimensions du QR code final
    grid_size = 20
    qr_width = part_width * grid_size
    qr_height = part_height * grid_size

    # Création d'une nouvelle image pour le QR code assemblé
    qr_code = Image.new('RGB', (qr_width, qr_height))

    # Placement de chaque partie dans l'image finale
    for i, part in enumerate(parts):
```

```
            row = i // grid_size
            col = i % grid_size
            x = col * part_width
            y = row * part_height
            qr_code.paste(part, (x, y))

    # Sauvegarde du QR code assemblé
    qr_code.save(output_file)
    print(f"QR code assemblé sauvegardé sous {output_file}")

    # Décodage du QR code
    decoded_objects = decode(qr_code)
    if decoded_objects:
        for obj in decoded_objects:
            print("Type:", obj.type)
            print("Data:", obj.data.decode("utf-8"))
    else:
        print("Aucun code QR détecté")

# Utilisation du script
input_folder = "."
output_file = "qr_code_assemble.png"
total_pieces = 340  # De piece_0.png à piece_339.png

assemble_qr_code(input_folder, output_file, total_pieces)
```

Une fois le code exécuté je me retrouve avec le lien:



Une fois sur le lien on obtient notre flag

Flag: HLB2024{£@5y_W4y_t0_f1nD_th3_FlaG}