Writeup

Mon equipe: Nekketsu

Challenge: Soft.reading

Auteur: unpasswd & W1z4rd

Pour ce challenge, un code python a ete mis a notre disposition. Voici ci-dessous le code en question.

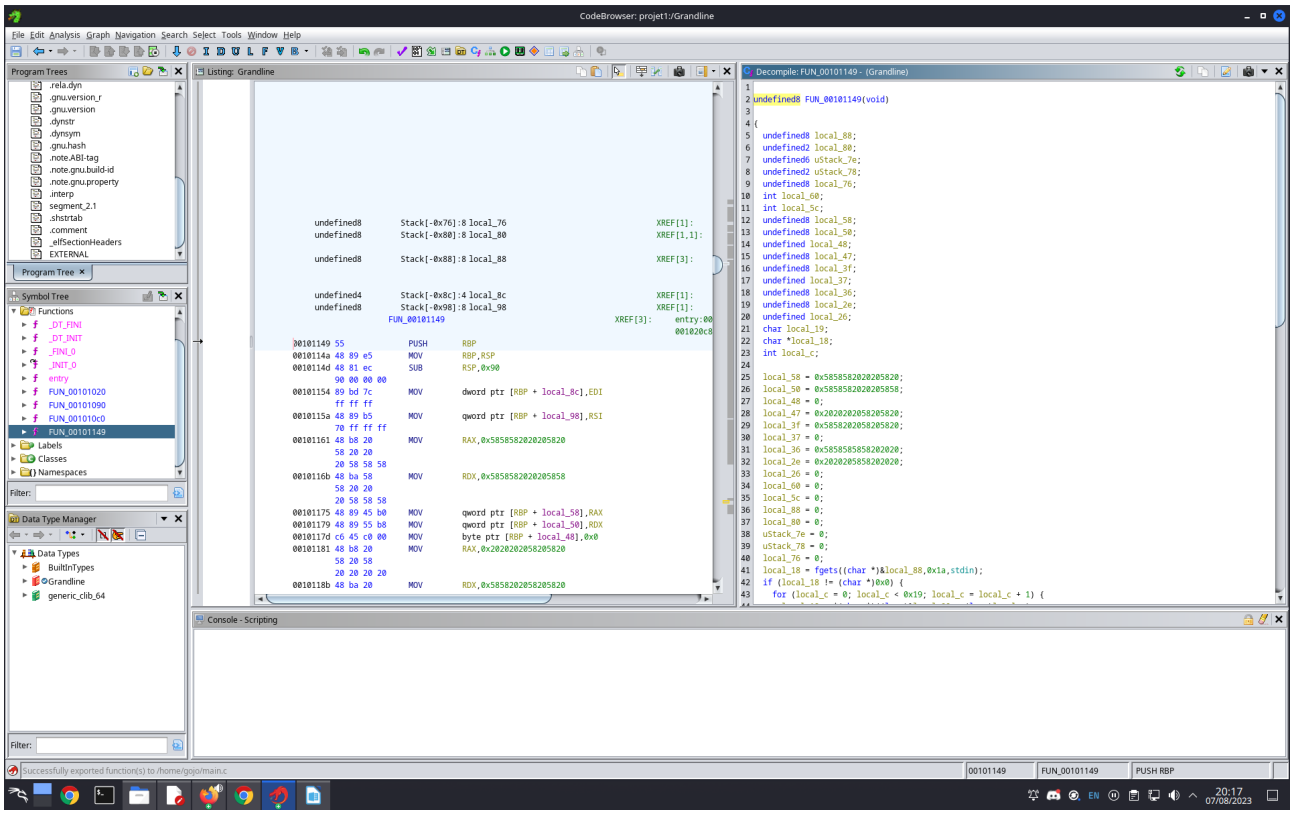
```python
import os

try:
    m = open("/flag.txt", "r")
except:
    print("Le fichier flag.txt n'est pas présent.")

if __name__ == '__main__':
    t_ = input("PATH du fichier à lire : ")
    if t_.startswith("/"):
        exit("\nLe PATH du fichier doit commencer par '/'")
    elif '..' in t_:
        exit("\nLe PATH du fichier ne doit pas contenir '..'")

    c_ = os.path.expanduser(t_)
    try:
        print(open(c_, "r").read())
    except:
        exit("\nImpossible d'ouvir le fichier")
```

Lorsqu'on analyse le code on constate qu'il y a un fichier nomme flag.txt succeptible de contenir une information. Cependant lors de l'execution du scrip, on nous demande d'entrer un repertoire mais lorsque le chemin d'accès commence par "/" ou lorsqu'il y a ".." dans le chemin le programme s'arrete.

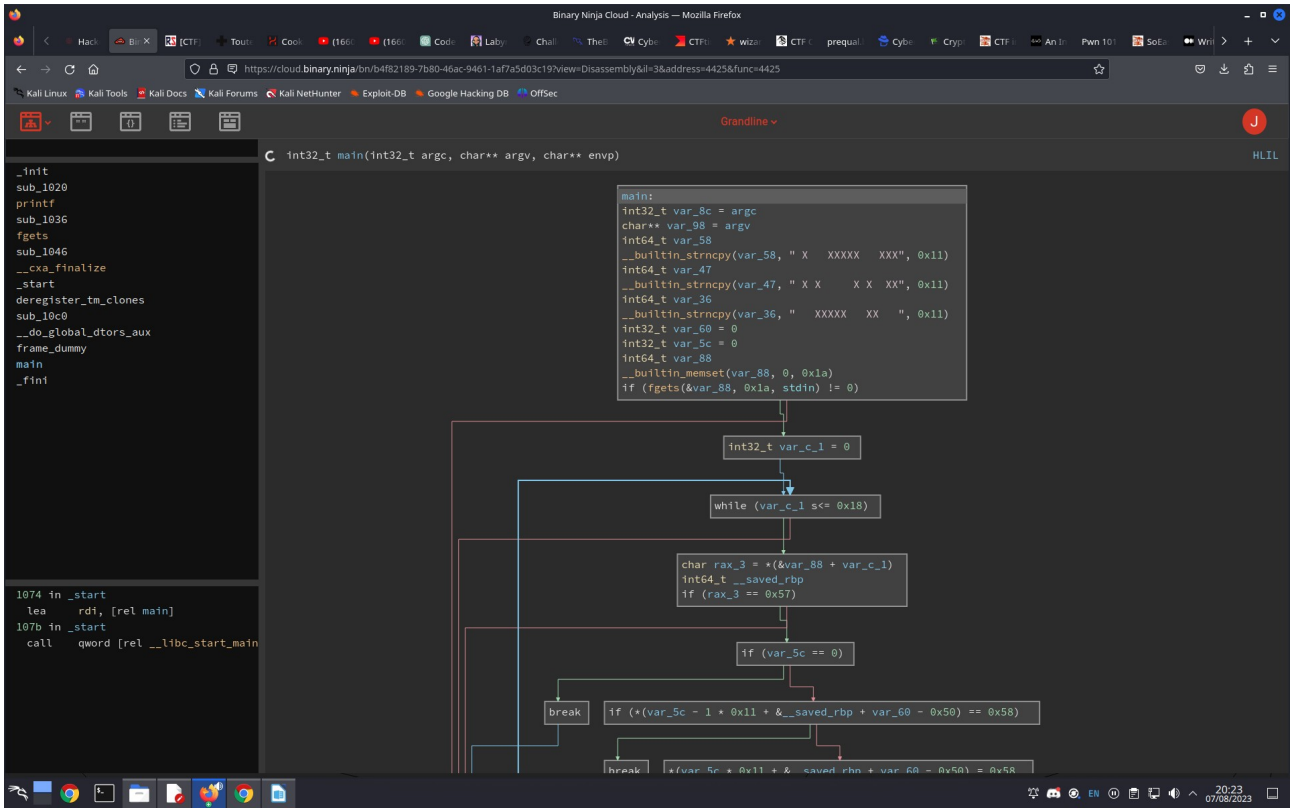
Nous avions donc fait des recherches sur les fonction starswith() et os.path.expanduser() qui ont ete utilise dans le script et nous sommes tombes sur un bon writeup. Nous avions finalement compris que ~ est votre répertoire personnel, mais ~user est le répertoire personnel de l'utilisateur. Quand nous avions verifie le repertoire personnel de sys en utilisant la commande echo ~sys c'etait /dev.
Nous avions donc utilise ces infos pour avoir le contenu du fichier flag.txt

Nous avions donc un lien mega sur lequel un binaire a ete mis a notre disposition. Il s’agissait du binaire d’un programme de labyrinth. Le programme réagit aux commandes de déplacement entrées par le joueur ('W', 'S', 'A', 'D') en ajustant la position du personnage ou de l'objet sur la grille, en fonction de certaines règles.

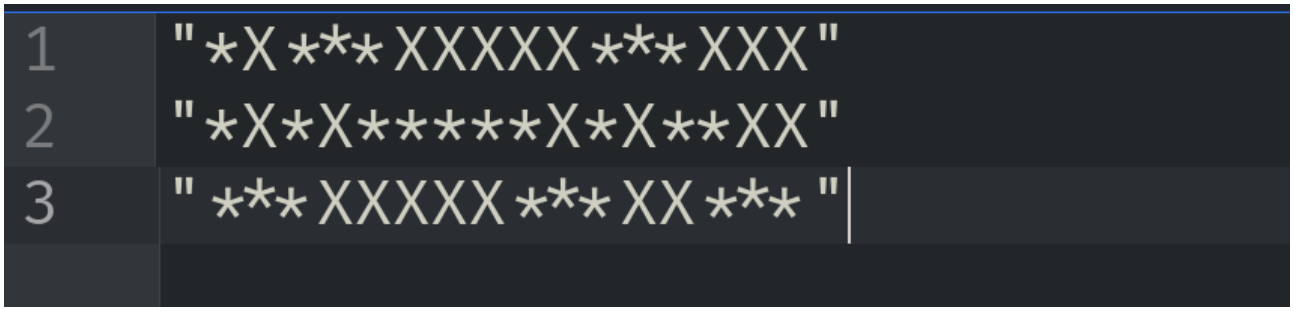
Nous avions premierement procede a la decompilation du binaire avec Ghidra.



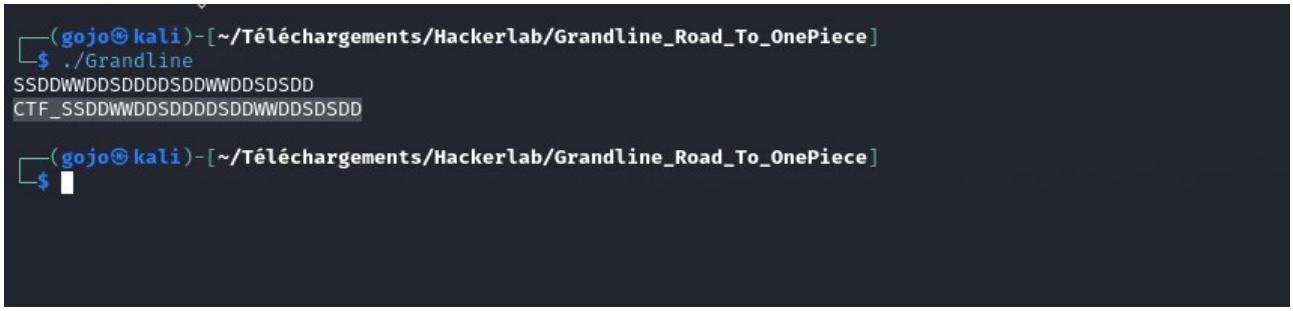
Lorsqu’on analyse le main on constate qu’il s;agit en effet d’une grille de 03 lignes et 16 colonnes. Pour afficher le flag, il fallait atteindre la case situee a la 03eme ligne et 16eme colonne. Nous avions aussi pense a decompile le binaire avec binary Ninja pour avoir plus d’informations et avec binary ninja nous obtenons notre labyrinthe.



C’est donc a nous de determiner le chemin a emprunter sachant qu’il ne fallait pas entrer plus de 25 caracteres. Nous avions donc reecris notre grille pour avoir une vu claire.



Apres analyse nous avions pu determiner le chemin a emprunter pour atteindre la case contenant le flag qui eest en effet la derniere case.



Flag: **CTF_SSDDWWDDSDDDDSDDWWDDSDSDD**