WriteUp du challenge "Soft.reading"

Equipe : N4t10n

Dans un premier temps, nous avons un programme python qui affiche le contenu d'un chemin qu'on lui passe en paramètre. Mais s'il commence par « .. » ou « / » il n'affiche rien et s'arrête. Après quelques recherche, on trouve qu'il y a une vulnérabilité au niveau de la fonction **os.path.expanduser.** Pour exploiter cette vulnérabilité il fallait passer en paramètre **~sys/fd/6** au programme. Une fois cela fait, on obtient un lien vers un binaire sur mega.

Le challenge consistait à résoudre un jeu de labyrinthe représenté par un programme binaire nommé "Grandline", contenu dans un dossier zippé appelé "Grandline_Road_To_OnePiece". Lors de la première exécution du programme, celui-ci demandait à l'utilisateur d'entrer un chemin. Pour résoudre le challenge, nous avons décidé de décompiler le programme en utilisant Ghidra, ce qui nous a permis de comprendre la logique du jeu. Nous avons découvert qu'il s'agissait d'un labyrinthe, où l'on devait utiliser les touches WSAD pour se déplacer et atteindre la sortie de coordonnées (2;15), tout en évitant les mines représentées par des X.

Première approche :

Dans un premier temps, nous avons envisagé d'implémenter un algorithme de recherche en profondeur (DFS) en utilisant Python pour résoudre le labyrinthe. Cependant, nous avons rapidement réalisé qu'une carte du labyrinthe serait un atout précieux pour trouver rapidement le chemin optimal.

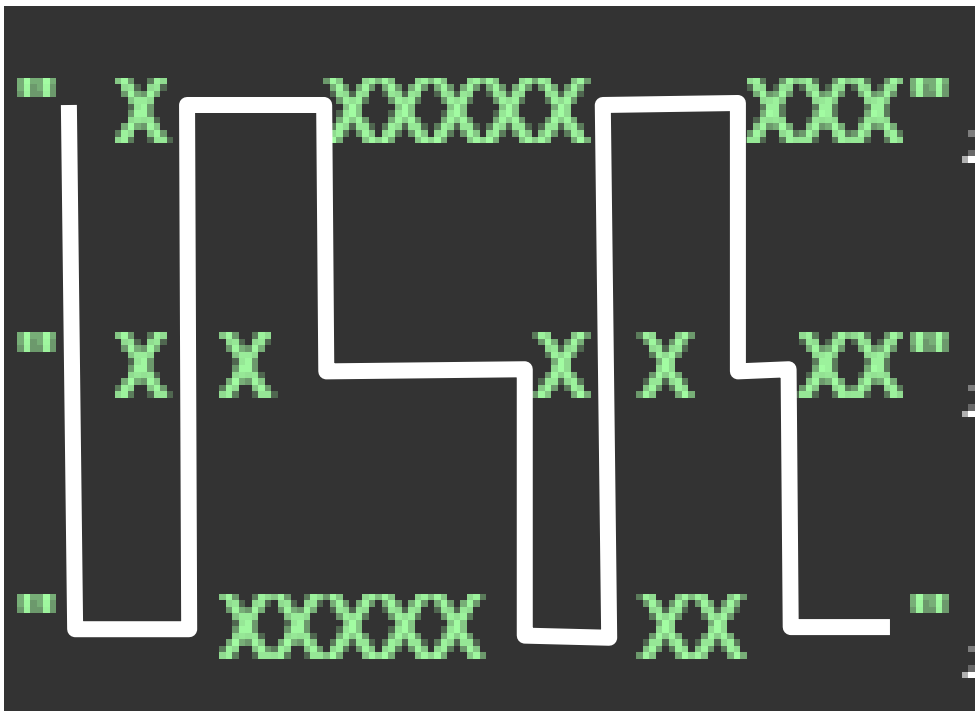
Découverte de la carte :

Pour obtenir la carte du labyrinthe, nous avons utilisé d'autres outils de décompilation tels que BinaryNinja. Cela nous a permis de lire attentivement le code du programme binaire. Voici un aperçu de la sortie obtenue :

```
```
    char** var_98 = argv;
    int64_t var_58;
    __builtin_strncpy(var_58, " X   XXXXX   XXX", 0x11);
    int64_t var_47;
    __builtin_strncpy(var_47, " X X   X X XX", 0x11);
    int64_t var_36;
    __builtin_strncpy(var_36, "   XXXXX   XX   ", 0x11);
```
```

```
    int32_t var_60 = 0;
    int32_t var_5c = 0;
```

En analysant cette sortie, nous avons réussi à reconstituer la carte du labyrinthe. Voici la représentation du labyrinthe :



Grâce à la carte, nous avons pu déduire le bon chemin pour atteindre la case finale de coordonnées (2;15). Le chemin est le suivant: SSDDWWDDSDDDDSDDWWDDSDSDD. Le format du flag étant CTF_userentry, nous avons pu trouver le flag : **CTF_SSDDWWDDSDDDDSDDWWDDSDSDD**. Donc nous n'avons pas eu besoin d'un script pour la résolution de ce challenge