Calculateur express! (EASY)

Auteur(s) :

`Hokanosekai`

Catégorie :

`Pwn`

Description :

Résoudre tous les calcules en moins de 1min.

`nc 161.35.21.37 40010`

Flag : `UHOCTF{fake_flag}`

Instructions :

Le joueur doit résoudre différents calcul en trouvant l'opérateur associés aux nombres et au résultat de l'application de l'opérateur passé sur les deux nombres.

En cas de bonne réponse, on renvoie un nouveau problème tant qu'il n'a pas réussi 100 calculs. Une fois les 100 calculs atteint le joueur recevra le flag.

En cas de mauvaise réponse la connexion est coupée, un message d'erreur est envoyé et le joueur doit recommencer a zéro.

Si jamais le joueur venait a dépassé 1 min de connexion alors elle est coupée et un message de timeout est envoyé.

PROF

Dans cette première version les opérateurs sont limités a 4 :

- `+` : une addition de deux nombres
- `-` : une soustraction de deux nombres
- `*` : une multiplication de deux nombres
- `/` : une division de deux nombres

Tout autres caractères renverra une erreur.

Exemple d'utilisation :

```
$ nc {ip} {port}
Bienvenue sur le calculeur express !
Vous avez 1 minute pour résoudre un maximum de calculs !
5 ? 3 = 15
```

```
> *
2 ? 3 = 5
> +
4 ? 1 = 3
> -
18 ? 9 = 2
> /
```

Solution :

Le joueur doit donc résoudre 100 calculs en moins d'une minute. Pour cela il doit trouver l'opérateur qui a été appliqué sur les deux nombres pour obtenir le résultat.

Pour cela il suffit de faire un script qui va se connecter au serveur et résoudre les calculs.

Nous allons utiliser le package python `pwntools` pour nous connecter au serveur et envoyer les réponses.

```python
from pwn import *

# run the process locally
# p = process('./calculateur_express')

# connect to remote server
p = remote('ip', port)

op = ['+','-','*','/']
i = 0

while True:
    w = p.recvline().decode().strip()

    if 'flag' in w:
        print(w)
        break
    if '?' not in w:
        continue

    i += 1
    w = w.split('=')
    res = w[1]
    w = w[0]

    p.recvuntil(b'>')

    for x in op:
        if (round(float(eval(w.replace('?', x))), 2) == float(res)):
            print(f"{i} | {w.replace('?', x)} = {float(res)}")
            p.sendline(x.encode())
            continue
```

PROF

On attend de recevoir une ligne du serveur, puis on vérifie si le mot `flag` est présent dans la ligne reçue. Si c'est le cas on affiche la ligne et on sort de la boucle.

Si le la ligne contient un `?` alors nous allons devoir résoudre le calcul. Pour cela on récupère le résultat et le calcul, puis on envoie le résultat au serveur.

On peut alors lancer le script et récupérer le flag.

```
hoka@hoka ~/c/U/U/P/Calculateur-Express (main)> python3 solve.py
[+] Starting local process './calculateur_express': pid 6870
97 | 64 / 59  = 1.08
98 | 86 * 41  = 3526.0
99 | 19 - 38  = -19.0
100 | 23 / 86  = 0.27
UHOCTF{fake_flag}
[*] Stopped Process './calculateur_express' (pid 6870)
```

Flag : `UHOCTF{C4lcul3s_3t_3xpr3ss10ns_M4gn1f1qu3s!}`

Hosting

```
sudo docker build -t calculateur_express .
sudo docker network create -d bridge calculateur_express
```

The command to start the challenge is:

```
sudo docker run -p 40010:40010 --detach --name calculateur_express --
network calculateur_express calculateur_express:latest
```

PROF

The command to stop the challenge (since CTRL+C won't work) is:

```
sudo docker stop calculateur_express
```