[REVERSE]

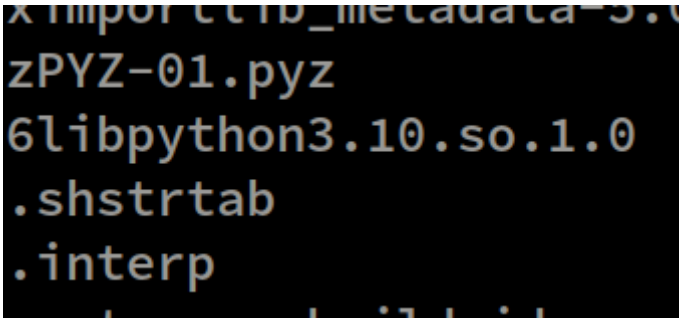
Invert Writeup

```
file invert
```

```
[eazy@archlinux output]$ file invert
invert: ELF 64-bit LSB executable, x86-64, version 1 (SYSV)
dID[sha1]=8203f620bc278dd6a87dfca5cfeebceecc1998bd, for GNU
```

```
strings invert
```

On remarque que c'est un executable python :



Decompiler le binaire python

```
git clone https://github.com/extremecoders-re/pyinstxtractor
/opt/pyinstxtractor
cd /opt/pyinstxtractor/
mv /opt/pyinstxtractor/pyinstxtractor.py .
python pyinstxtractor.py invert
```

Entrée possible (invert.pyc) :

```
[eazy@archlinux output]$ python pyinstxtractor.py invert
[+] Processing invert
[+] Pyinstaller version: 2.1+
[+] Python version: 3.10
[+] Length of package: 6774269 bytes
[+] Found 62 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: pyi_rth_inspect.pyc
[+] Possible entry point: invert.pyc
[+] Found 134 files in PYZ archive
```

```
cd invert_extracted/cd PYZ-01.pyz_extracted/
```

Fichier à garder en vue :

```
chall.pyc
importlib_metadata-
invert.pyc
ld-linux-x86-64.so.
```

invert.pyc contient le code du script.

On a maintenant besoin du fichier .pyc pour avoir le code.

```
pycdc invert.pyc > invert-cracked.py
```

Code invert-cracked.py :

Définition de result

```
result = 6633
```

Vérifie si le resultat est égale à 6633

```
if p == result:
    print(noice + fun08731(it,P,x))
```

Formattage du flag et return le flag formaté :

```
def fun08731(nop,hello,n):
    m = ' '
    y = nop[0] + nop[18]+ nop[19] + n + nop[17]
    o = nop[17] + nop[4] + nop[21]+ nop[4] + nop[17] + nop[18] + n
    itx = nop[5] + nop[11] + nop[0] + nop[6]
    u = ':'
    return itx + u + m + o + hello + nop[12] + y
```

```
./invert

entrée utilisateur : 739

entrée utilisateur - 2 = 737
737 * 9 = 6633
comparaison entre 6633 et le calcul de la saisie
print le flag
```

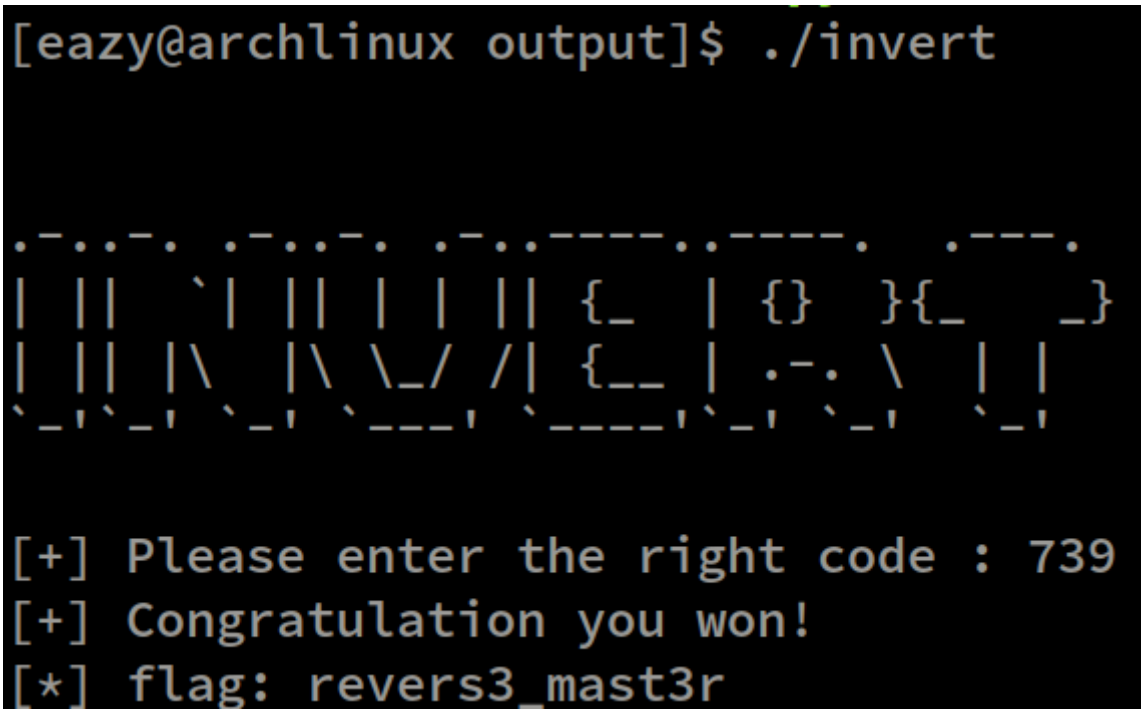
Automatisation de la solution

Generation de la clé

Faire un petit algorithme qui calcule la valeur finale :

Code solver :

```
print("| Code Solver |")
for x in range(0,10000):
        code = x
        x = x - 2
        x = x * 9
        if x == 6633:
                print("\n[+] Compared to :", x, "\n[+] Code is : ", code)
```



flag : revers3_mast3