

LOTTERY

The lottery shows its source code at the beginning of the execution. Also, it makes you to wait 5 minutes to enter a lottery number, so it evades bruteforce attacks.

```
C:\Users\giygas\Desktop\prng>python lottery.py
Trust in Lotto-Win. Trust in open source:
def next(seed):
    # Max winner number
    MAX = 2000000000
    # Min winner number
    MIN = 1

    # Change of the seed
    seed = seed * seed + seed

    # Truncation of the seed
    if (seed > 0xFFFFFFFF):
        seed = int(hex(seed)[-8:], 16)

    # Return of [new seed, new winner number]
    return [seed, seed % (MAX - MIN) + MIN]

# Initialization of random seed
seed = random.randint(0x1337, 0xFFFFFFFF)

Last Lottery numbers:
19088396
361347183
203687859
1866050424
108956123
Wait 5 minutes to play the next round of lottery...
```

After seeing the source code, we can tell that is vulnerable to number prediction since it's using a very weak "Pseudorandom Number Generator".

We only need to know the seed's value so we can generate a number equal to the first winner number (19088396), and then predict a sequence of numbers with that seed.

To do so, we will generate a number for every single value the seed is able to own until we generate a number that matches the first winner number.

Once that occurs, we will know the initial seed value, and we will be able to predict the sequence of number, and win the lottery.

A documented PoC exploit can be found in this folder, by the name of prng_exploit.py

The image shows two terminal windows side-by-side. The left window, titled 'Símbolo del sistema', shows the execution of a Python script named 'lottery.py'. The script defines a function 'next(seed)' that generates a new seed and a winning number based on a pseudo-random number generator. It also shows the 'Last Lottery numbers' and a prompt for the user to enter a lottery number. The right window, titled 'Selecionar C:\Windows\system32\cmd.exe', shows the execution of a Python script named 'prng_exploit.py'. This script predicts the seed value '19088396' and lists a series of predicted winning numbers. Blue arrows indicate the flow of information from the exploit's predicted seed to the lottery program's output.

```
C:\Users\giygas\Desktop\prng>python lottery.py
Trust in Lotto-Win. Trust in open source:
def next(seed):
    # Max winner number
    MAX = 2000000000
    # Min winner number
    MIN = 1

    # Change of the seed
    seed = seed * seed + seed

    # Truncation of the seed
    if (seed > 0xFFFFFFFF):
        seed = int(hex(seed)[-8:], 16)

    # Return of [new seed, new winner number]
    return [seed, seed % (MAX - MIN) + MIN]

# Initialization of random seed
seed = random.randint(0x1337, 0xFFFFFFFF)

Last Lottery numbers:
19088396
361347183
203687859
1866850424
108956123
Wait 5 minutes to play the next round of lottery...
Enter your lottery number: 123541244
Keep trying ;)

Last Lottery numbers:
19088396
361347183
203687859
1866850424
108956123
995202943
Wait 5 minutes to play the next round of lottery...
Enter your lottery number: 259908996
wow! YOU WON THE LOTTERY!! GET YOUR REWARD WITH THE CODE "THHC{Luck13r_th4n_Pelayo}"!!!!1111

C:\Users\giygas\Desktop\prng>
```

```
C:\Users\giygas\Desktop\prng>python prng_exploit.py
Seed: 079913830
19088396
361347183
203687859
1866850424
108956123
995202943
259908996
1291193224
2896301
1968532112
717296979
975938455
400714620
1254273440
77310757
314565543
1872682252
1370506415
15860469
107313079
1006382363
106536095
453711043
1513120712
406589676
232202961
878053523
1186429911
526204604
55315808
760348259
952011752
828173963
1456591087
233926195
1414454264
319445596
1320025088
1599426563
363499527
863222828
1664587535
19367892
448888856
1107199483
1479818783
1131858339
409534504
943884747
2708593
1437520243
181200695
737509176
1449610303
1862282051
991548488
415680364
357622607
1152301844
167868761
112793915
923368031
1881355491
967104728
1251116299
```