

LinkedIn: [Kelvin Kimotho](#)

crackme-py

Medium

Reverse Engineering

picoCTF 2021

AUTHOR: SYREAL

Hints ?

Description

(None)

crackme.py

Solution

I downloaded the **crackme.py** python script using **wget** command line tool.

```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
└─$ wget https://mercury.picoctf.net/static/fd0e358d4b82695c220c0d6013c11484/crackme.py
--2025-02-15 15:59:58-- https://mercury.picoctf.net/static/fd0e358d4b82695c220c0d6013c11484/crackme.py
Resolving mercury.picoctf.net (mercury.picoctf.net) ... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)[18.189.209.142]:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1463 (1.4K) [application/octet-stream]
Saving to: 'crackme.py'

crackme.py                               100%[>] 1.43K --.-KB/s in 0s

2025-02-15 16:00:00 (19.7 MB/s) - 'crackme.py' saved [1463/1463]

(kali@kali)-[~/Desktop]
└─$ ls
crackme.py

(kali@kali)-[~/Desktop]
└─$
```

I went ahead inspecting the file metadata and its strings contents using **exftool** and **strings** tool respectively but found no flag.

```
kali@kali: ~/Desktop
$ exiftool crackme.py
ExifTool Version Number      : 12.76
File Name                    : crackme.py
Directory                    : .
File Size                     : 1463 bytes
File Modification Date/Time   : 2021:03:16 01:16:33+00:00
File Access Date/Time        : 2025:02:15 16:00:00+00:00
File Inode Change Date/Time   : 2025:02:15 16:00:00+00:00
File Permissions              : -rw-rw-r--
File Type                    : TXT
File Type Extension          : txt
MIME Type                    : text/plain
MIME Encoding                 : us-ascii
Newlines                     : Unix LF
Line Count                   : 54
Word Count                   : 182

(kali@kali)~/Desktop
$ strings crackme.py | grep pico
pico

(kali@kali)~/Desktop
$
```

I opened the file using **nano** editor to inspect the code and understand how the program works.

```
GNU nano 8.1 crackme.py
# Hiding this really important number in an obscure piece of code is brilliant!
# AND it's encrypted!
# We want our biggest client to know his information is safe with us.
bezos_cc_secret = "A:4@r%uL M~"M0c0AbcM-MFE0554ce"eN"

# Reference alphabet
alphabet = "!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN0PQRSTUVWXYZ+ \\"
          "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~"

def decode_secret(secret):
    """ROT47 decode

    NOTE: encode and decode are the same operation in the ROT cipher family.
    """

    # Encryption key
    rotate_const = 47

    # Storage for decoded secret
    decoded = ""

    # decode loop
    for c in secret:
        index = alphabet.find(c)
        original_index = (index + rotate_const) % len(alphabet)
        decoded = decoded + alphabet[original_index]

    print(decoded)

def choose_greatest():
    """Echo the largest of the two numbers given by the user to the program

    Warning: this function was written quickly and needs proper error handling
    """

    user_value_1 = input("What's your first number? ")
    user_value_2 = input("What's your second number? ")
    greatest_value = user_value_1 # need a value to return if 1 & 2 are equal

    if user_value_1 > user_value_2:
        greatest_value = user_value_1
    elif user_value_1 < user_value_2:
        greatest_value = user_value_2

    print( "The number with largest positive magnitude is "
          + str(greatest_value) )

choose_greatest()
```

The program had two methods implemented namely **decode_secret(secret)** which takes secret as a parameter

```
def decode_secret(secret):
    """ROT47 decode

    NOTE: encode and decode are the same operation in the ROT cipher family.
    """

    # Encryption key
    rotate_const = 47

    # Storage for decoded secret
    decoded = ""

    # decode loop
    for c in secret:
        index = alphabet.find(c)
        original_index = (index + rotate_const) % len(alphabet)
        decoded = decoded + alphabet[original_index]

    print(decoded)
```

and `choose_greatest()` which takes no argument. I also learnt about a secret stored under a variable name ' `bezos_cc_secret`' .

```
# And it's encryption!
# We want our biggest client to know his information is safe with us.
bezos_cc_secret = "A!4@r%uL M~^M0c0AbcM-MFE055a4ce"eN"
```

I also learnt that the `decode_secret` method was implemented but never called. I went ahead and called the method passing `bezos_cc_secret` as the input to the method.

```
+ str(greatest_value) )
decode_secret(bezos_cc_secret)
choose_greatest()
```

Running the program called the `decode_secret()` method where the `bezos_cc_secret` was decoded and printed as the flag.

```
File Actions Edit View Help
(kali@kali)-[~/Desktop]
└─$ python crackme.py
picoCTF{1|V|_4_p34|\|ut_dd2c4616}
What's your first number?
```

Thats how I captured the flag.