Impossible Password

Answer: **HTB{40b949f92b86b18}**

Instructions:

1. Run the program.
   1. Insert any value, and the program will return the value.
2. Run:
   1. string impossible\_password.bin
      1. This will print out all the printable strings.
3. From the printable string, we see that there is a special string ‘SuperSeKretKey’
4. Enter that as an input, and the program will prompt the user for another input, but this time after inputting any value, the program will end.
5. We can take a closer look into the reason why this happens by running ltrace on the program or by using a reverse engineer tool and looking at the assembly code.
   1. TLDR, it is creating a random string and comparing it to the input (I THINK)
   2. And only printing the flag is the input and the string are equal, however the string is randomly generated
6. To fix the input issue and reach the flag, we can change the assembly code using a reverse engineer tool.
7. Start by installing radare2.
8. Run the command:
   1. r2 -A -w impossible\_password.bin
9. Go to the instruction:
   1. 0x00400968 750c jne 0x400976
      1. After analysing the assembly code, we can see that after inputting ‘SuperSekretKey’ as the first input, for our second input, it will execute a series of instructions then eventually reaching the instructions specified.
      2. This instruction will cause the program to jump to the end of the function, causing it to skip over the instruction to print the flag.
10. Insert instruction beforehand by running:
    1. S 0x00400966
    2. wa jmp 0x0040096a
       1. By running these two commands, we insert an instruction that allows the program to jump to the instruction to print the flag instead of ending the program.
11. Exit radare2 by running ‘q’.
12. Execute the file.
    1. Input ‘SuperSeKretKey’ for the first input
    2. Input anything for the second input

Tutorials:

<https://medium.com/@shaswata56/impossible-password-hackthebox-reversing-challenge-8c98b8da6db6>

Baby RE:

Answer: **HTB{B4BY\_R3V\_TH4TS\_EZ}**

Instructions:

1. After running the program, it will ask the user for an input.
2. Run ltrace on the file.
3. You will see that it tries to compare the input with “abcde122313”.
4. Enter “abcde122313” as the input, and we will get the flag
5. Easy 😊

Tutorial:

<https://www.aldeid.com/wiki/HackTheBox-Challenges-Reversing-Baby-RE>

Exation

Answer: **HTB{l3g1c3l\_sh1ft\_l3ft\_1nsr3ct1on!!}**

Instructions:

1. First install radare2 and upx
   1. sudo apt install radare2
   2. sudo apt install upx
2. Unpack the file
   1. upx -d exation\_v1
3. Run radare2 on the file
   1. r2 -A exation\_v1
   2. s main
   3. pdf
4. Go to address 0x00404d2d
   1. This is the encrypted password
      1. We know this by reading the instructions from address 0x00404cf0 to 0x00404d5c
      2. The instructions between these addresses are the password verification process.
5. Encrypted password should be:
   1. 1152 1344 1056 1968 1728 816 1648 784 1584 816 1728 1520 1840 1664 784 1632 1856 1520 1728 816 1632 1856 1520 784 1760 1840 1824 816 1584 1856 784 1776 1760 528 528 2000
6. To decrypt the password (Simplest way lol):
   1. Each number is a multiple of 16, so divide each number by 16
   2. Then the value given is the ASCII code for the password
   3. For example:
      1. 1152 🡪 72 🡪 H
      2. 1344 🡪 84 🡪 T
      3. And so on
7. The decrypted password will be the flag

Tutorials:

<https://medium.com/@0x1Overflow/hackthebox-exatlon-reversing-challenge-274d254d843>

<https://www.youtube.com/watch?v=v7jPLeYspUk>

misdirection

Answer: **HTB{DIR3ctLy\_1n\_Pl41n\_Si7e}**

Instructions:

1. Unzip the file (When unzipping, don’t use windows, use Linux lol)
2. Each folder is a letter/number in the encoded string.
3. Inside the folder, contains files with numbers as the filename.
   1. These numbers are the position number of the folder name.
4. For example:
   1. The folder ‘S’ contains:
      1. 1
      2. This means that, in the encoded string, position 1 is the letter ’S’
   2. Meanwhile the folder ‘s’ contains:’
      1. 24
      2. This means that at position 24, is the letter ‘s’
5. Encoded string should be:
   1. SFRCe0RJUjNjdEx5XzFuX1BsNDFuX1NpN2V9
6. Decode the string using Base64 format
   1. <https://www.base64decode.org/>
7. Decoded string should be the solution

Tutorials:

<https://cyruslab.net/2020/04/22/hacktheboxmisdirection/>

Longbottom’s Locker

Answer: **HTB{n3v1LL3\_Da\_burM3s3-pyth0n\_sL4y3r}**

Instructions:

1. Unzip the file.
2. If you take a closer look at the socute.jpg, it has zip files embedded into it.
3. Change the socute.jpg to socute.zip.
4. Inside the zip file should contain a file name ‘donotshare’.
5. This is a pickle file, so you can extract the content simply by writing a python script.

import pickle

with open('donotshare', 'rb') as f:

o = pickle.load(f)

outstr = ''

for line in o:

for char,n in line:

outstr += char\*n

outstr += '\n'

print(outstr)

1. After outputting the content in the ‘donotshare’ file, full screen the output, and you should see a big text made up of smaller symbols. This is the password to the html website:
   1. Password should be: Gu1d0-v4N-R055Um.
2. Enter password to the html website, and it will give the flag

Tutorials:

<https://medium.com/@s4u1j3y/hackthebox-longbottom-locker-writeup-4b85cdd07edc>

Query

Answer:

Instructions:

Tutorials: