Write Up PicoCTF Reverse Engineering

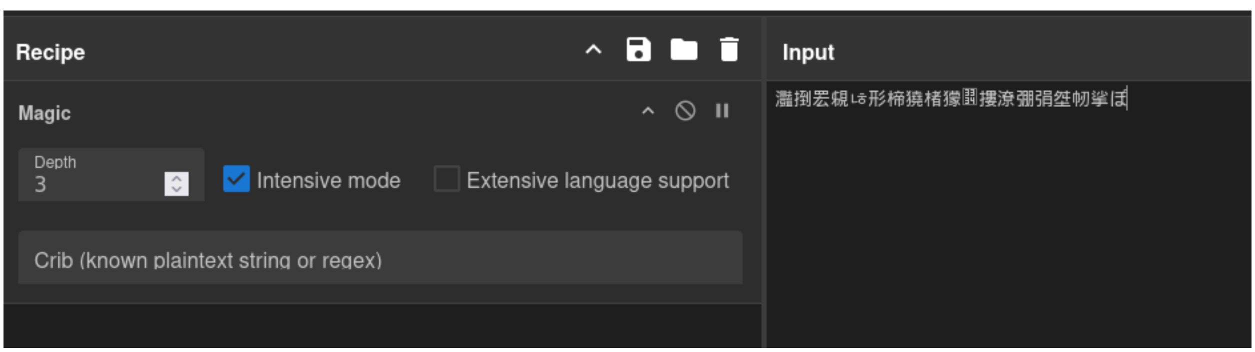
Mutiara Setya Rini

Daftar Isi

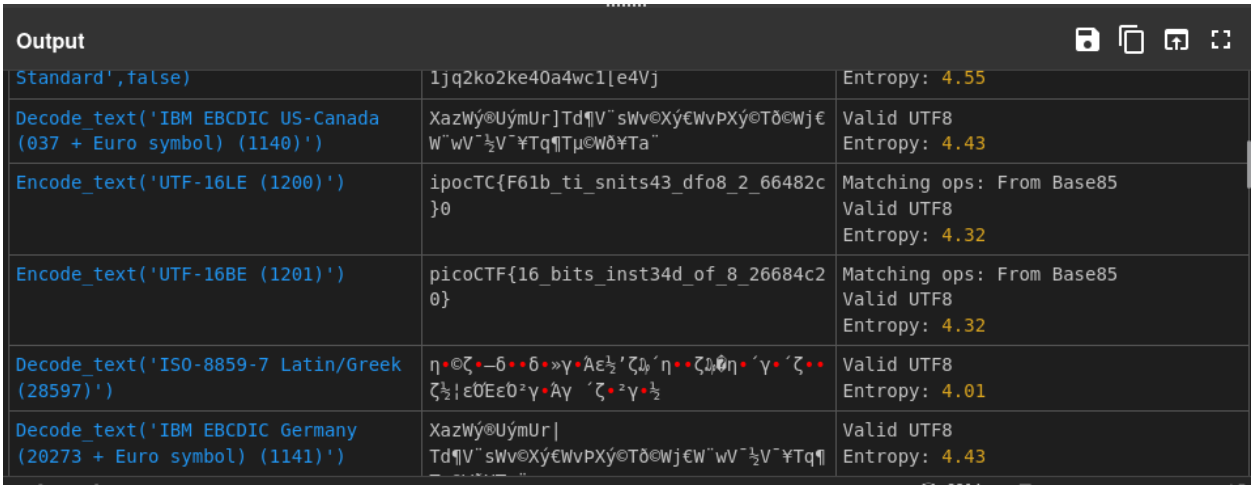
Transformation

Pertama, download file enc pada attachment challenge. Coba lihat isinya dengan command cat dan didapatkan seperti ini :

Kemudian, karena hint pada challenge ini menyuruh kita untuk menggunakan decoder online, coba kita decode menggunakan cyberchef. Copy-paste isi dari file enc ke dalam input cyberchef dan pilih mode Magic intensive.



Akan muncul banyak output yang keluar, cari dan pilih output yang sesuai dengan template string flag, yakni picoCTF{<flag>}.

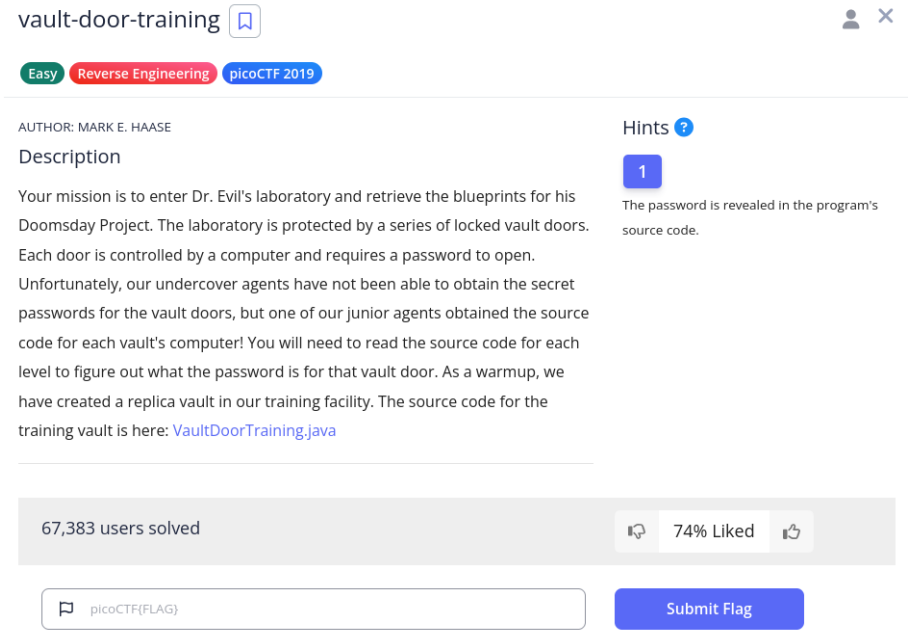


Output

Standard',false)	1jq2ko2ke4Oa4wc1[e4Vj	Entropy: 4.55
Decode_text('IBM EBCDIC US-Canada (037 + Euro symbol) (1140)')	XazWý®UýmUr]Td¶V¨sWv©Xý€WvÞXý©Tð©Wj€W¨wV¯½V¯¥Tq¶Tµ©Wð¥Ta¨	Valid UTF8 Entropy: 4.43
Encode_text('UTF-16LE (1200)')	ipocTC{F61b_ti_snits43_dfo8_2_66482c}0	Matching ops: From Base85 Valid UTF8 Entropy: 4.32
Encode_text('UTF-16BE (1201)')	picoCTF{16_bits_inst34d_of_8_26684c20}	Matching ops: From Base85 Valid UTF8 Entropy: 4.32
Decode_text('ISO-8859-7 Latin/Greek (28597)')	η•©ζ•–δ••δ•»γ•Áε½'ζ‫´η••ζ‫®η•´γ•´ζ••ζ½¦εΌΕεΌ²γ•Áγ ´ζ•²γ•½	Valid UTF8 Entropy: 4.01
Decode_text('IBM EBCDIC Germany (20273 + Euro symbol) (1141)')	XazWý®UýmUr|Td¶V¨sWv©Xý€WvÞXý©Tð©Wj€W¨wV¯½V¯¥Tq¶	Valid UTF8 Entropy: 4.43

Didapatkan flag `picoCTF{16_bits_inst34d_of_8_26684c20}`

valut-door-training



"You will need to read the source code for each level to figure out what the password is for that vault door."

Dari deskripsi challenge ini, kita mendapatkan petunjuk untuk melihat source code dari file yang ada di attachment. Berikut adalah isi dari file attachment tersebut.

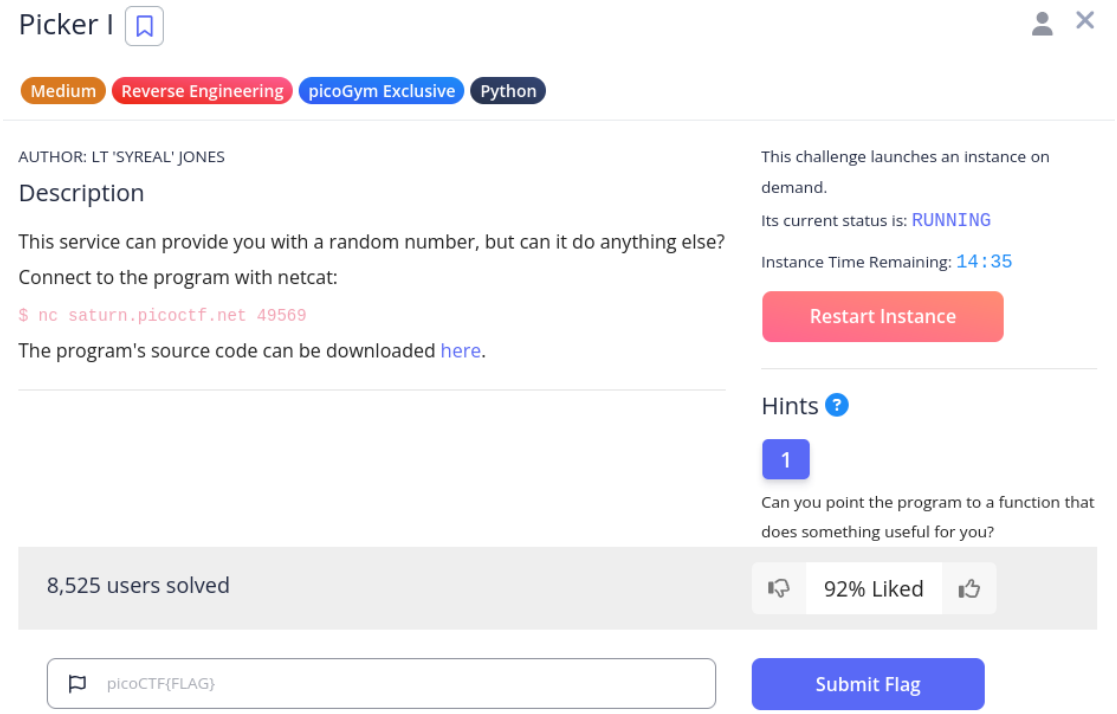
```
┌──(mutzy㉿kali)-[~/Downloads]
└─$ cat VaultDoorTraining.java
import java.util.*;

class VaultDoorTraining {
    public static void main(String args[]) {
        VaultDoorTraining vaultDoor = new VaultDoorTraining();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter vault password: ");
        String userInput = scanner.next();
        String input = userInput.substring("picoCTF{".length(),userInput.length()-1);
        if (vaultDoor.checkPassword(input)) {
            System.out.println("Access granted.");
        } else {
            System.out.println("Access denied!");
        }
    }

    // The password is below. Is it safe to put the password in the source code?
    // What if somebody stole our source code? Then they would know what our
    // password is. Hmm... I will think of some ways to improve the security
    // on the other doors.
    //
    // -Minion #9567
    public boolean checkPassword(String password) {
        return password.equals("w4rm1ng_Up_w1tH_jAv4_eec0716b713");
    }
}
```

Voilaa, flag ada di dalam file tersebut. Flagnya adalah `picoCTF{w4rm1ng_Up_w1tH_jAv4_eec0716b713}`

Picker I



Launch instance dan connect terminal dengan netcat yang tersedia.

```
┌──(mutzy㉿kali)-[~]
└─$  nc saturn.picoctf.net 49569
Try entering "getRandomNumber" without the double quotes...
⟹ getRandomNumber
4
Try entering "getRandomNumber" without the double quotes...
⟹
```

saat kita memasukkan getRandomNumber maka akan muncul ouput angka 4. Kemudian, coba kita lihat source code dari file attachment challenge.

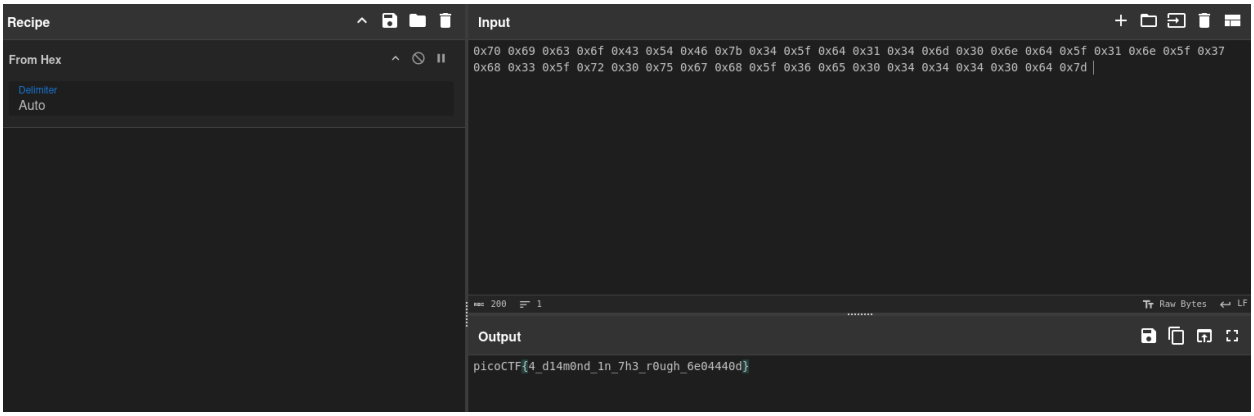
```
def win():
  # This line will not work locally unless you create your own 'flag.txt' in
  #   the same directory as this script
  flag = open('flag.txt', 'r').read()
  #flag = flag[:-1]
  flag = flag.strip()
  str_flag = ''
  for c in flag:
    str_flag += str(hex(ord(c))) + ' '
  print(str_flag)


def esoteric2():
  esoteric = \
  '''
```

Ada satu function yang menarik, yaitu function win. Mari kita coba untuk run function win.

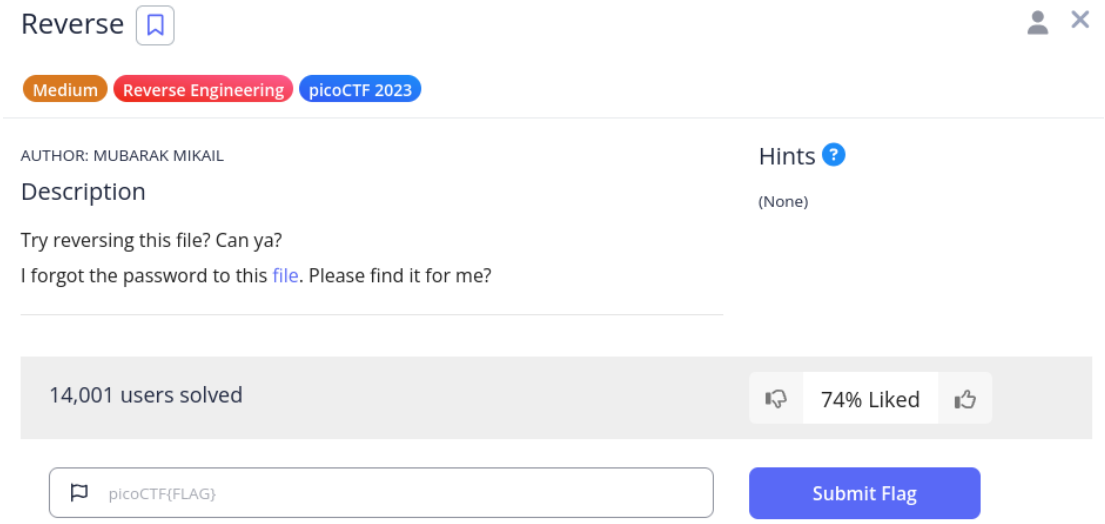
```
⟹ win
0×70 0×69 0×63 0×6f 0×43 0×54 0×46 0×7b 0×34 0×5f 0×64 0×31 0×34 0×6d 0×30 0×6e 0×64 0×5f 0×31 0×6e 0×5f 0×37 0×68 0×33
0×5f 0×72 0×30 0×75 0×67 0×68 0×5f 0×36 0×65 0×30 0×34 0×34 0×34 0×30 0×64 0×7d
Try entering "getRandomNumber" without the double quotes ...
⟹
```

Kita mendapatkan kumpulan hex code dari function win tersebut. Selanjutnya kita ubah hex tersebut menjadi character ascii menggunakan cyberchef.



voila, kita berhasil mendapatkan flagnya, yaitu `picoCTF{4_d14m0nd_1n_7h3_r0ugh_6e04440d}`

Reverse



Pertama, download file attachment pada challenge. Kemudian cek tipe filenya dengan command file ret.

```
┌──(mutzy㉿kali)-[~/Downloads]
└─$ file ret
ret: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.
2, BuildID[sha1]=32195c65c0c8ca5bd239fa824d4d79231cca5f78, for GNU/Linux 3.2.0, not stripped
```

Coba kita lihat isi file tersebut dengan command cat ret.

```
┌──(mutzy㉿kali)-[~/Downloads]
└─$ cat ret
@@@@◆▓▓▓((UU   ◆◆◆-◆=◆=x◆◆-◆=◆=◆888 XXXDDS◆td888 P◆td◆ ◆ ◆ DDQ◆tdR◆td◆-◆=◆=hh/lib64/ld-linux-x86-64.so.2GNU◆GNU2\e◆◆◆[◆9
◆◆MMy#◆_xGNU
◆
◆e◆m ▒0MF◆
          ◆ 7"libc.so.6__isoc99_scanfputs__stack_chk_failprintf__cxa_finalizestrcmp__libc_start_mainGLIBC_2.7GLIBC_2.4G
iu▒i_2.2s◆◆@◆?◆?◆?◆?iste◆?MCloneTable__gmon_start___ITM_registerTMCloneTableii
◆H◆=◆◆◆.◆◆H◆=◆.H◆◆.H9◆tH◆◆.H◆◆t◆◆◆◆◆H◆=◆.H◆5◆.H)◆H◆◆H◆◆?H◆◆H◆H◆◆tH◆◆.H◆◆◆◆fD◆◆◆◆◆=◆.u+UH◆=b.H◆◆t%/D◆◆◆◆%
                                                                                                    H◆=f.◆◆◆◆◆◆d◆◆◆◆].]◆◆◆◆
◆◆◆◆H◆E◆H◆◆◆◆◆◆◆◆H◆◆◆B◆◆◆◆◆u▒H◆=◆CTF{H◆3lf_r3v3H◆E◆H◆U◆H◆r5ing_suH◆cce55fulH◆E◆H◆U◆H◆_c83965dH◆E◆H◆=◆
◆◆◆◆◆◆H◆M◆dH3    H◆=◆
           %(t◆◆◆◆◆◆◆f.◆◆◆AWL◆=◆*AVI◆◆AUI◆◆ATA◆◆UH◆-◆*SL)◆H◆◆◆◆◆H◆◆t1◆◆L◆◆L◆◆D◆◆A◆◆H◆◆H9◆u◆H◆[]A\A]A^A_◆ff.◆◆◆◆◆◆H◆H◆
◆Enter the password to unlock this file: %sYou entered: %s
Password correct, please see flag: picoCTF{3lf_r3v3r5ing_succe55ful_c83965de}Access denied@|◆◆◆t◆◆◆◆◆◆◆◆◆<◆◆◆\%◆◆◆◆,◆◆◆
◆◆◆◆◆4zRx
        ◆◆◆◆/D$4◆◆◆`F▒J
◆                      ◆?▒:*3$"\8◆◆◆t0◆◆◆P◆Q◆◆◆◆E◆C
D◆8◆◆◆eF◆I▒◆E ◆E(◆D0◆H8◆G@n8A0A(B B▒B◆`◆◆◆◆◆
H◆▒◆◆◆◆o◆◆◆
◆
 ▒◆?x◆◆ ◆◆◆◆◆◆o◆◆◆◆o◆◆◆◆o◆◆◆◆o◆=0@P`@GCC: (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0▒8X|◆◆◆    ◆
◆
 ◆
 ◆◆◆H ◆ ◆ ◆=◆=◆=▒◆?@▒@◆
                     @!◆7▒@F◆=m◆y◆=◆◆◆◆◆◆◆!◆◆◆◆=◆◆=◆◆=◆◆ ◆▒◆?◆
                                                          @ ◆ @.@@
                                                                   HGcw◆@◆◆ @◆ ◆◆e◆▒▒@◆◆/◆▒@◆◆◆@) C"crtstuff.cdereg
ister_tm_clones__do_global_dtors_auxcompleted.8061__do_global_dtors_aux_fini_array_entryframe_dummy__frame_dummy_init_ar
ray_entryret.c__FRAME_END____init_array_end_DYNAMIC__init_array_start__GNU_EH_FRAME_HDR_GLOBAL_OFFSET_TABLE___libc_csu_f
ini_ITM_deregisterTMCloneTableputs@@GLIBC_2.2.5_edata__stack_chk_fail@@GLIBC_2.4printf@@GLIBC_2.2.5__libc_start_main@@GL
IBC_2.2.5__data_startstrcmp@@GLIBC_2.2.5__gmon_start____dso_handle_IO_stdin_used__libc_csu_init__bss_startmain__isoc99_s
canf@@GLIBC_2.7__TMC_END___ITM_registerTMCloneTable__cxa_finalize@@GLIBC_2.2.5.symtab.strtab.shstrtab.interp.note.gnu.pr
operty.note.gnu.build-id.note.ABI-tag.gnu.hash.dynsym.dynstr.gnu.version.gnu.version_r.rela.dyn.rela.plt.init.plt.got.pl
t.sec.text.fini.rodata.eh_frame_hdr.eh_frame.init_array.fini_array.dynamic.data.bss.comment▒▒#886XX$I|| W◆◆◆o◆◆a
                                                                                                    ▒i◆◆◆q◆◆
◆  ◆◆ ◆ D◆◆ ◆◆◆◆◆◆=◆-◆◆?◆@0◆◆◆◆P◆◆◆e◆HH
```

Terdapat line yang menunjukkan adanya password untuk unlock file tersebut dan kelihatannya flag dari challenge ini berada pada line yang sama. Voila, flag telah ditemukan, yaitu `picoCTF{3lf_r3v3r5ing_succe55ful_c83965de}`