# 攻防世界76号

原创

四代机您发多少 于 2021-10-15 11:02:54 发布 31 收藏

文章标签： 安全

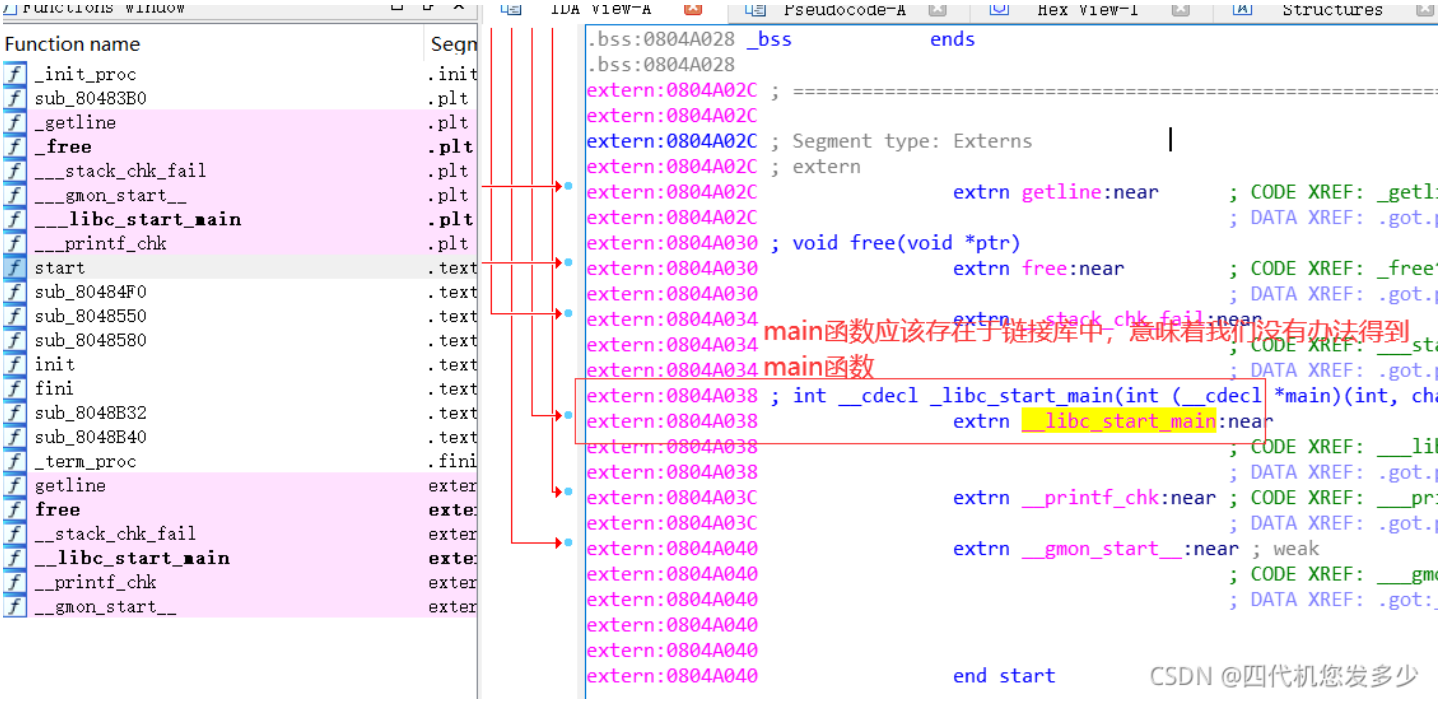


本文链接：https://blog.csdn.net/njh18790816639/article/details/116095716

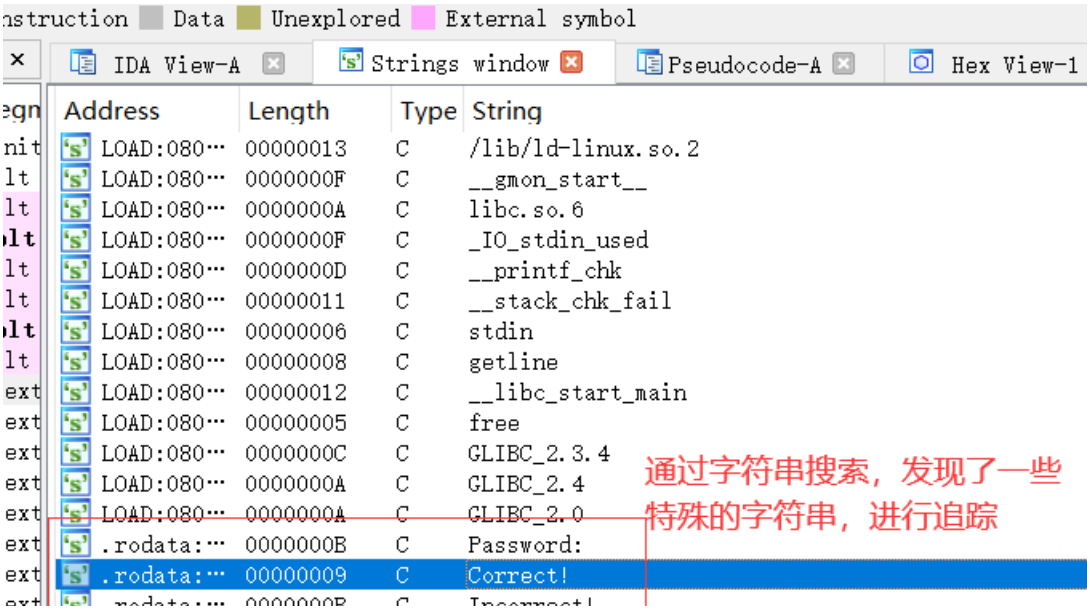


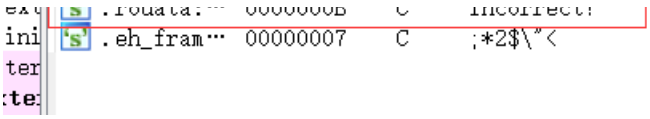
## 76号

经过探测，发现是32为ELF，但是ida32打开没有发现main函数！




刚开始的想法是main函数存在于此ELF文件中，但是被隐藏起来了，也就是main函数被加密过了，但是进行追踪，发现main函数应该是存在于链接库中，也就意味着本题我们无法得到main函数，但是我们可以猜测main函数要调用那些函数！








一直进行追踪，可以发现这个关键函数。

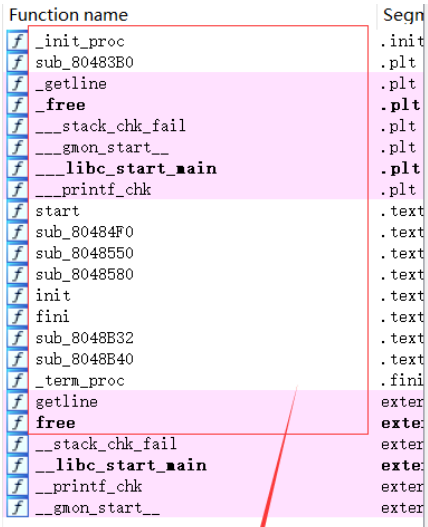
```
.text:08048467                  mov     [esp], eax
.text:0804846A                  call    _getline        ; 输入
.text:0804846F                  test    eax, eax
.text:08048471                  mov     ebx, eax
.text:08048473                  js      short loc_804848F
.text:08048475                  mov     eax, [esp+1Ch]
.text:08048479                  mov     dword ptr [esp+4], 0
.text:08048481                  mov     [esp], eax
.text:08048484                  call    sub_8048580     ; 进入函数
.text:08048489                  test    eax, eax
.text:0804848B                  mov     ebx, eax
.text:0804848D                  jnz     short loc_80484B6
.text:0804848F
.text:0804848F loc_804848F:                             ; CODE XREF: .text:08048473↑j
.text:0804848F                  mov     dword ptr [esp+4], offset aIncorrect ; "Incorrect!"
.text:08048497                  mov     dword ptr [esp], 1
.text:0804849E                  call    ___printf_chk   ; 输出错误
.text:080484A3
.text:080484A3 loc_80484A3:                             ; CODE XREF: .text:080484CA↓j
.text:080484A3                  mov     eax, [esp+1Ch]
.text:080484A7                  mov     [esp], eax
.text:080484AA                  call    _free
.text:080484AF                  mov     eax, ebx
.text:080484B1                  mov     ebx, [ebp-4]
.text:080484B4                  leave
.text:080484B5                  retn
.text:080484B6 ; ---------------------------------------------------------------------------
.text:080484B6
.text:080484B6 loc_80484B6:                             ; CODE XREF: .text:0804848D↑j
.text:080484B6                  mov     dword ptr [esp+4], offset aCorrect ; "Correct!"
.text:080484BE                  mov     dword ptr [esp], 1
.text:080484C5                  call    ___printf_chk
```

其为关键函数

输出成功



然后我们进去看看！




找不到main函数，还有一种方法，就是点开每个函数看看，发现只有sub-8048580函数存在实体且较长，而且有加密的痕迹

```
34  char v33; // [esp+80h] [ebp-2Ch]
35  char v34; // [esp+81h] [ebp-2Bh]
36  char v35; // [esp+82h] [ebp-2Ah]
37  char v36; // [esp+83h] [ebp-29h]
38  char v37; // [esp+85h] [ebp-27h]
39  unsigned int v38; // [esp+8Ch] [ebp-20h]
40
41  v38 = __readgsdword(0x14u);
42  v2 = a2;                                  // v2接受了a2的数据
43  while ( 2 )
44  {
45    memset(v5, 0, 128u);
46    v3 = *(_BYTE *)(a1 + v2);               // v3也就是a1和a2的组合
47    v5[(v3 + 64) % 128] = 1;                // v5无用
48    if ( (unsigned __int8)(v3 - 10) <= 'p' )
49    {
50      switch ( v3 )                         // 对v3进行遍历
51      {
52        case '\n':                          // 换行键，v2=13，说明flag应该为13位
53          return v2 == 13 && v28 != 0;      // 因为return正确返回植为1，故v2=13
54        case '0':
55          if ( v2 || !v29 )
56            return 0;
57          v2 = 1;                           // 第一位，v2=1，以v2来进行推动遍历
58          continue;
59        case '1':
60          if ( v2 == 14 && v30 )
61            goto LABEL_12;
62          return 0;
```

我们不能执行return 0

那么就找到v2从1到13即可，组合到一起即为flag

flag{09vdf7wefijbk}