

攻防世界csaw2013reversing2Writeup

原创

bin_cat 于 2022-01-08 14:20:56 发布 231 收藏

文章标签: 安全

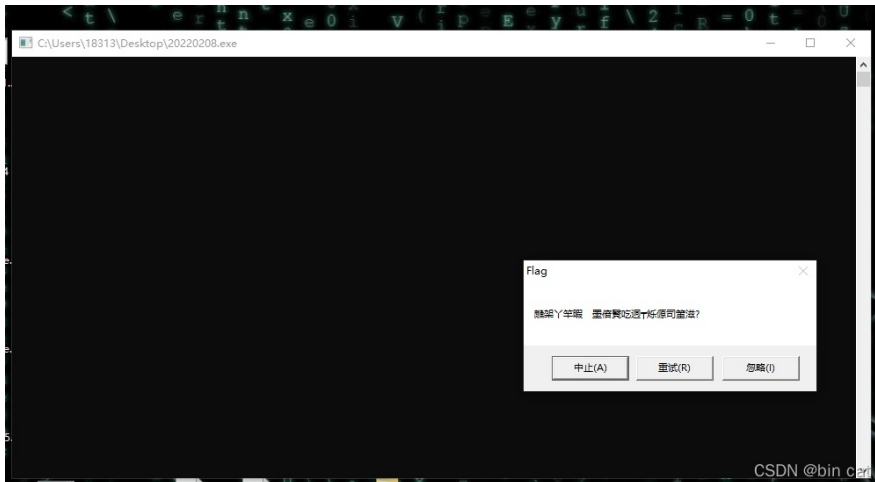
版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/m0_58348028/article/details/122378395

版权

32位无壳

先试着运行exe文件, 得到如下结果符合题目 描述输出的玩意是乱码



用ida解析

```
IDA View-A Pseudocode-A Hex View-I Structures Enum
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
    int v3; // ecx
    CHAR *lpMem; // [esp+8h] [ebp-Ch]
    HANDLE hHeap; // [esp+10h] [ebp-4h]

    hHeap = HeapCreate(0x40000u, 0, 0);
    lpMem = (CHAR *)HeapAlloc(hHeap, 8u, SourceSize + 1);
    memcpy_s(lpMem, SourceSize, &unk_409B10, SourceSize);
    if ( !sub_40102A() && !IsDebuggerPresent() )
    {
        MessageBoxA(0, lpMem + 1, "Flag", 2u);
        HeapFree(hHeap, 0, lpMem);
        HeapDestroy(hHeap);
        ExitProcess(0);
    }
    _debugbreak();
    sub_401000(v3 + 4, lpMem);
    ExitProcess(0xFFFFFFFF);
}
```

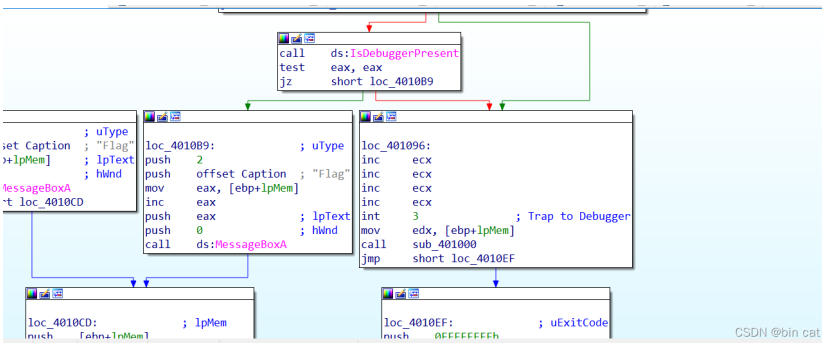
CSDN @bin cat

主函数逻辑, 跟进sub_40102A()函数中

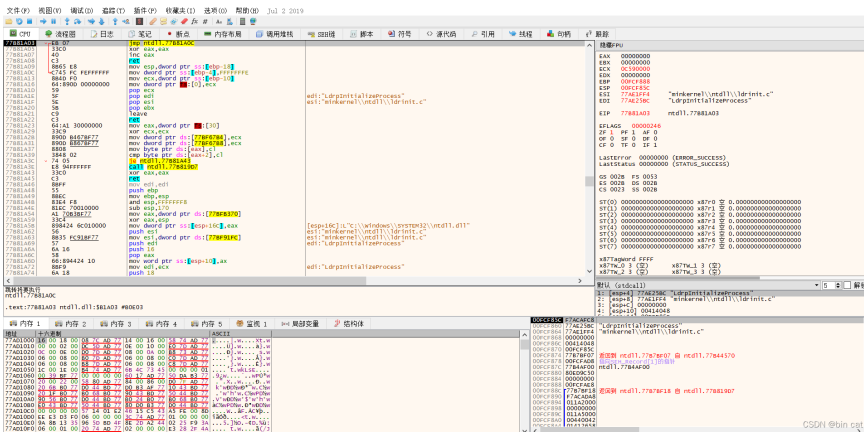
```
IDA View-A Pseudocode-A
1 int sub_40102A()
2 {
3     return 0;
4 }
```

CSDN @bin cat

发现函数的返回值默认为0, 但是!0就意味着if所包含的语句会一直被默认执行, 联想到我们一开始看到程序运行起来的乱码, 我们就确定if所包含的语句就是导致乱码发生的原因



结合汇编发现if语句的默认正是跳过了sub_401000所以我们猜测这个被跳过的藏有flag



开始动态调试

| | | | |
|---|---------------|--|--|
| B | E8 88000000 | call 20220208.391108 | |
| 0 | 83C4 10 | add esp,10 | |
| 3 | E8 A2FFFFFF | call 20220208.39102A | |
| 8 | 85C0 | test eax,eax | |
| A | 75 0A | jne 20220208.391096 | |
| C | FF15 14603900 | call dword ptr ds:[<&IsDebuggerPresent>] | |
| 2 | 85C0 | test eax,eax | |
| 4 | 74 23 | je 20220208.3910B9 | |
| 6 | 41 | inc ecx | |
| 7 | 41 | inc ecx | |
| 8 | 41 | inc ecx | |
| 9 | 41 | inc ecx | |
| A | 90 | nop | |
| B | 8B55 F4 | mov edx,dword ptr ss:[ebp-C] | |
| E | E8 5DFFFFFF | call 20220208.391000 | |
| 9 | EB 4A | jmp 20220208.3910EF | |
| 5 | 6A 02 | push 2 | |
| 7 | 68 20783900 | push 20220208.397820 | |
| C | FF75 F4 | push dword ptr ss:[ebp-C] | |
| F | 6A 00 | push 0 | |
| 1 | FF15 E4603900 | call dword ptr ds:[<&MessageBoxA>] | |

397820:"Flag"
CSDN @bin cat

执行完成后面就是退出不需要再执行此时edx地址中就存在flag

| 地址 | 十六进制 | ASCII |
|----------|-------------|------------------|
| 03180598 | 00 00 00 00 | |
| 031805A8 | 00 00 00 00 | |
| 031805B8 | 00 00 00 00 | |
| 031805C8 | 66 6C 61 67 | flag{reversing_ |
| 031805D8 | 73 5F 6E 6F | is_not_that_hard |
| 031805E8 | 21 7D 00 00 | !}..... |
| 031805F8 | 00 00 00 00 | |
| 03180608 | C0 00 18 03 | A...A...ipibibib |
| 03180618 | EE FE EE FE | ibibibibibibib |
| 03180628 | EE FE EE FE | ibibibibibibib |
| 03180638 | EE FE EE FE | ibibibibibibib |

CSDN @bin cat 出flag

flag{reversing_is_not_that_hard!}

```

00391099 41          inc ecx
0039109A 90          nop
0039109B 8B55 F4     mov  edx,dword ptr ss:[ebp-c]
0039109C E8 5DFFFFFF call 20220208.391000 ← ①
003910A3 90          nop
003910A4 4A          dec  eax
003910A5 6A 02      push 2
003910A7 68 20783900 push 20220208.397820
003910AC FF75 F4     push dword ptr ss:[ebp-c]
003910AF 6A 00      push 0
003910B1 FF15 E4603900 call dword ptr ds:[&MessageBoxA]
003910B7 EB 14      jmp 20220208.3910CD
003910B9 6A 02      push 2
003910BB 68 20783900 push 20220208.397820
003910C0 8B45 F4     mov  eax,dword ptr ss:[ebp-c]
003910C3 40          inc  eax
003910C4 50          push eax
003910C5 6A 00      push 0
003910C7 FF15 E4603900 call dword ptr ds:[&MessageBoxA]
003910CD FF75 F4     push dword ptr ss:[ebp-c]
003910D0 6A 00      push 0
003910D2 FF75 FC     push dword ptr ss:[ebp-4]
003910D5 FF15 08603900 call dword ptr ds:[&HeapFree]
003910DB 8945 F8     mov  dword ptr ss:[ebp-8],eax
003910DE FF75 FC     push dword ptr ss:[ebp-4]
003910E1 FF15 0C603900 call dword ptr ds:[&HeapDestroy]
003910E7 6A 00      push 0
003910E9 FF15 00603900 call dword ptr ds:[&ExitProcess]
003910EF 6A FF      push FFFFFFFF
003910F1 FF15 00603900 call dword ptr ds:[&ExitProcess]
003910F7 C9          leave
003910F8 C3          ret
003910F9 3B0D 04903900 cmp  ecx,dword ptr ds:[399004]
003910FF 75 02      jne 20220208.391103
00391101 F3:C3     ret
00391103 E9 09020000 jmp 20220208.391311
00391108 8BFF     mov  edi,edi
    
```

CSDN @bin-cat

也可以将1处跳转nop掉单步走到003910C7去触发messageboxA

这样也能得flag

The screenshot shows a debugger window with the following components:

- Assembly View:** Shows the execution path from instruction 0039109C (call 20220208.391000) to 003910C7 (call dword ptr ds:[&MessageBoxA]). A red arrow points to the call instruction at 0039109C.
- Registers Window:** Shows EIP at 003910C7, indicating the current instruction being executed.
- Dialog Box:** A message box titled "Flag" displays the text "flag{reversing_is_not_that_hard!}" with buttons for "中止(A)", "重试(R)", and "忽略(O)".
- Stack Window:** Shows the current function call stack, with the top entry being the current function.
- Command Line:** Shows the command ".text:003910C7 20220208.exe:510C7 #4C7".
- Memory View:** Shows a hex dump of memory at address 77AD1000.

CSDN @bin cat