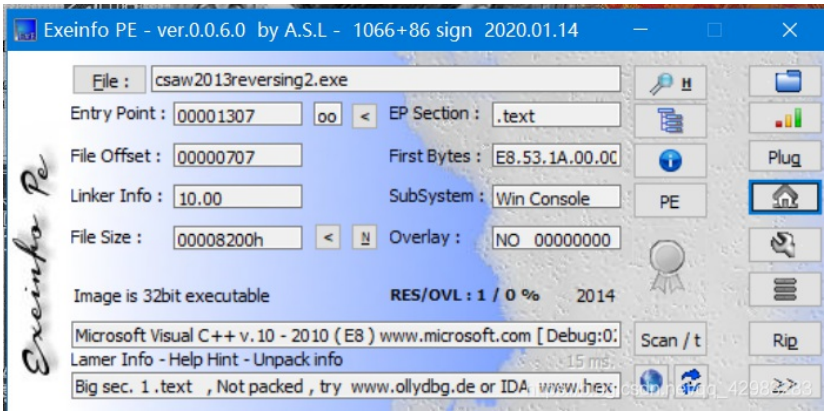
xctf攻防世界csaw2013reversing2 writeup

原创

qq_112419837 于 2020-11-05 17:15:18 发布 73 收藏

下载文件，查看：



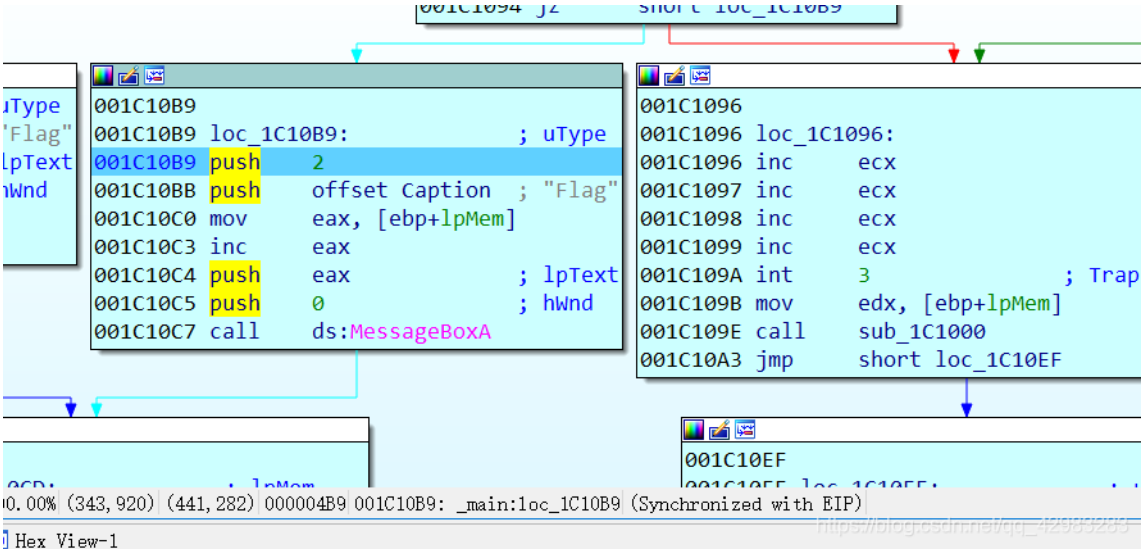
ida打开，f5：

大意是，sub_40102A或者IsDebuggerPresent这两个函数返回值有一个为1的话，程序貌似就要中断，但是最后的flag好像是固定的，也不需要输入啥的，直接setip大法：

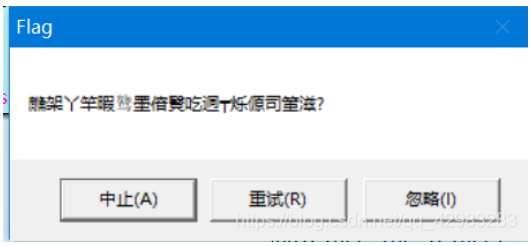
```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
  int v3; // ecx
  CHAR *lpMem; // [esp+8h] [ebp-Ch]
  HANDLE hHeap; // [esp+10h] [ebp-4h]

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

if那里下断点，动态运行，断下之后setip到flag这个程序段：



运行后是乱码：

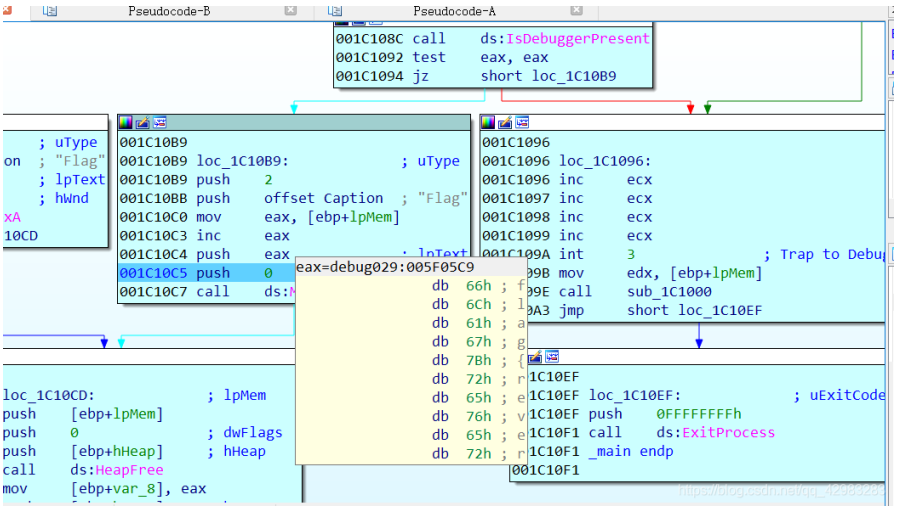


感觉得执行一下sub_1C1000，因为这个函数对第二个参数有一个赋值，也就是main中的lpMem：

```
unsigned int __fastcall sub_1C1000(int a1, int a2)
{
  int v2; // esi
  unsigned int v3; // eax
  unsigned int v4; // ecx
  unsigned int result; // eax

  v2 = dword_1C9B38;
  v3 = a2 + 1 + strlen((const char *)(a2 + 1)) + 1;
  v4 = 0;
  result = ((v3 - (a2 + 2)) >> 2) + 1;
  if ( result )
  {
    do
      *(_DWORD *)(a2 + 4 * v4++) ^= v2;
    while ( v4 < result );
  }
  return result;
}
```

那其实也好办，setip两次就行：



isdebuggerpresent那里不执行，直接到1c1000，然后就setip到1c10b9，eax里面就是flag。

双击lpmem也行，不过要+1: