实验四

实验要求

1. **利用工具软件实现PE文件节表免疫、节间免疫。**
2. **编程实现：在实验三的基础上实现节表免疫、节间免疫。**

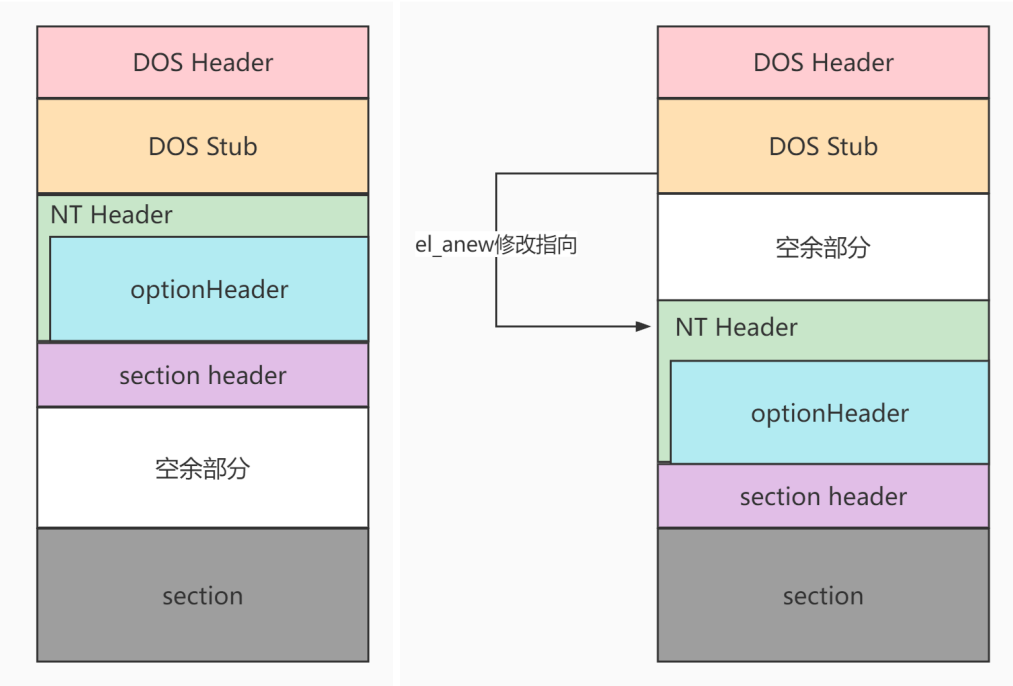
工具实现

首先解释一下节表免疫。

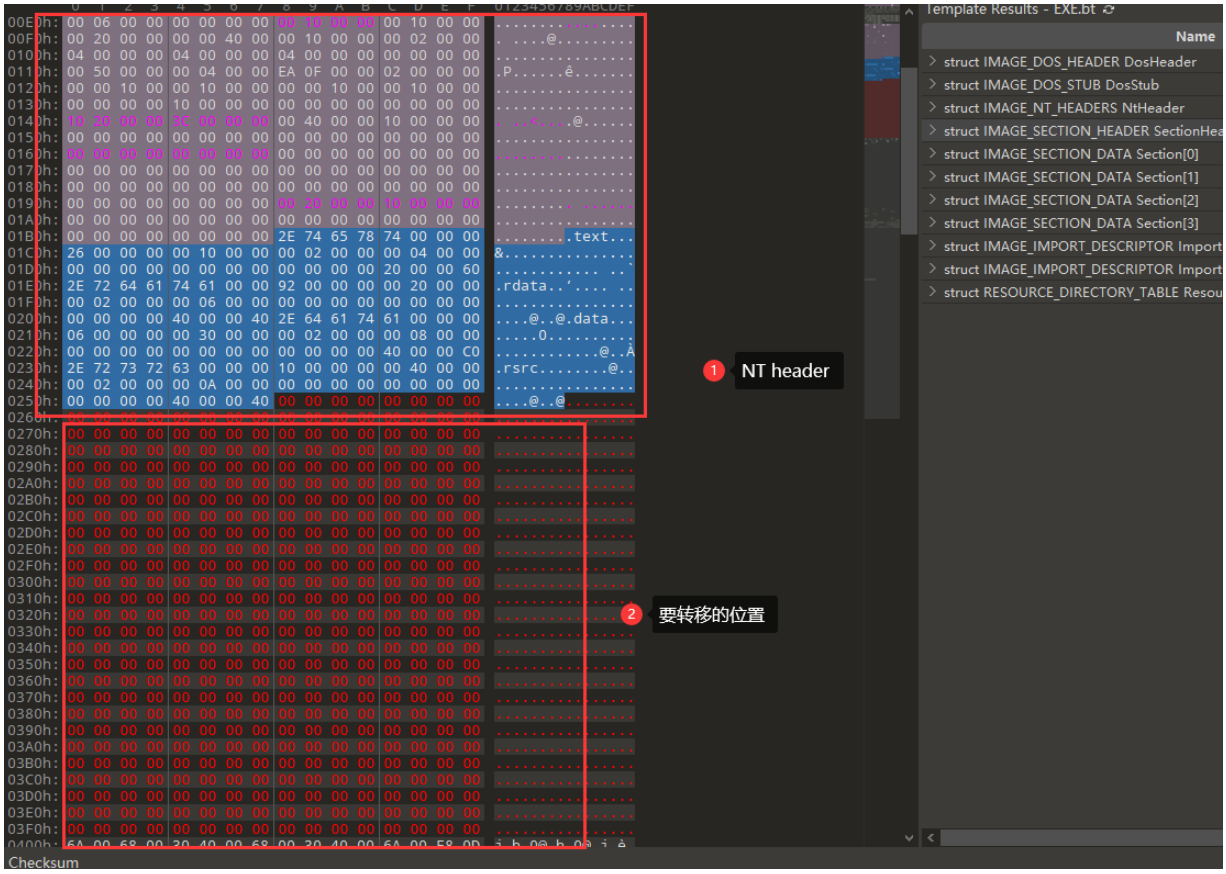
因为病毒要隐藏在exe文件中，为了隐藏代码，它会选择隐藏在option header也就是NtHeader之中，所以免疫就是为了让病毒感染PE文件时找不到位置容纳病毒代码的位置

节表免疫

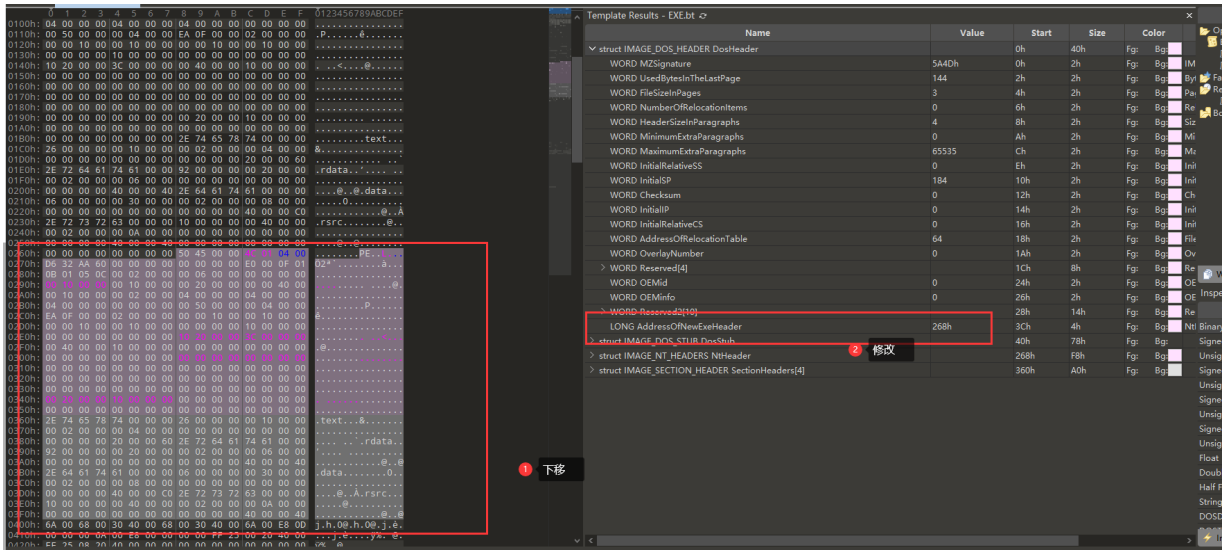
1. **关于第一种免疫方式**



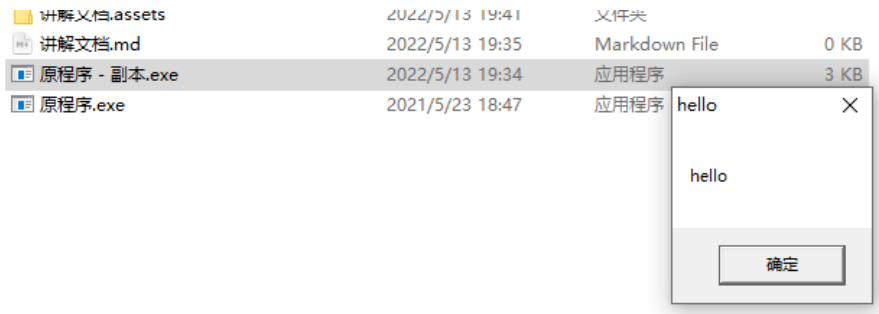
可以选择将PE文件的NTheader向下偏移，紧邻着节表。



同时设置DosHeader之中的Nt header `e_lfanew`。

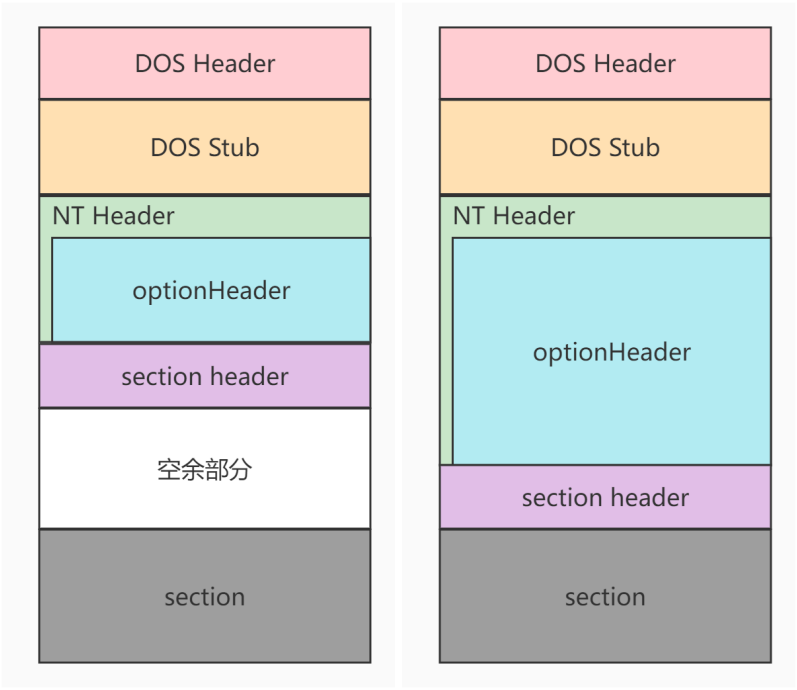


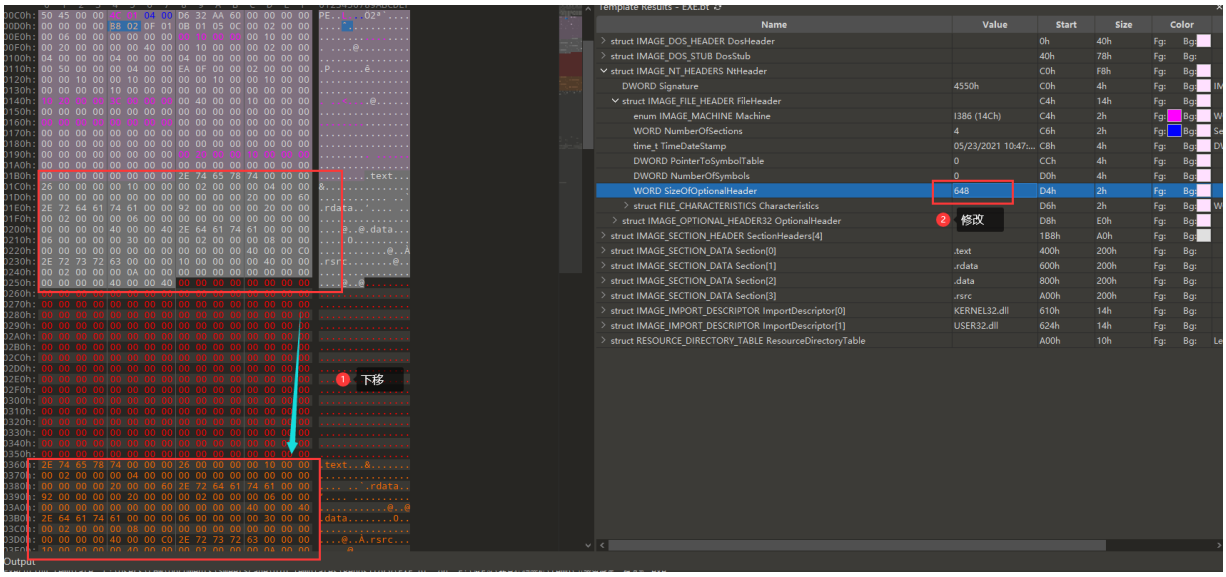
然后进行测试exe文件的 运行：



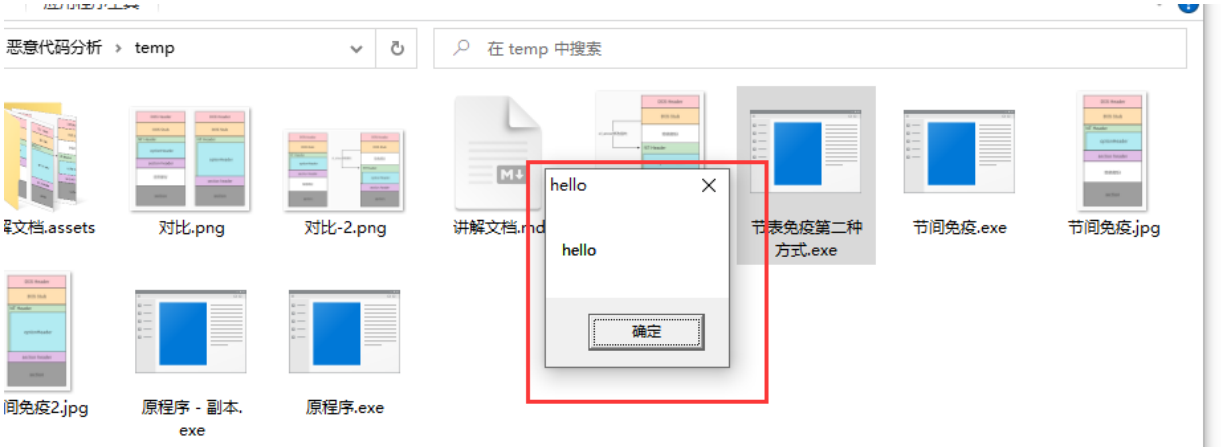
成功运行。

2. **关于第二种免疫**



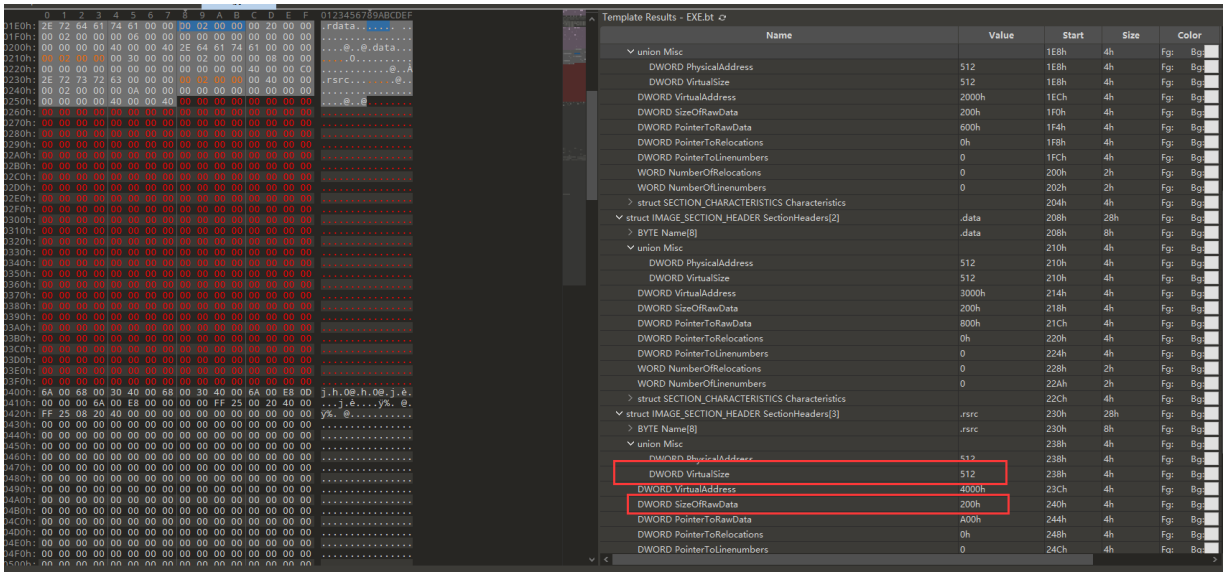


可以运行

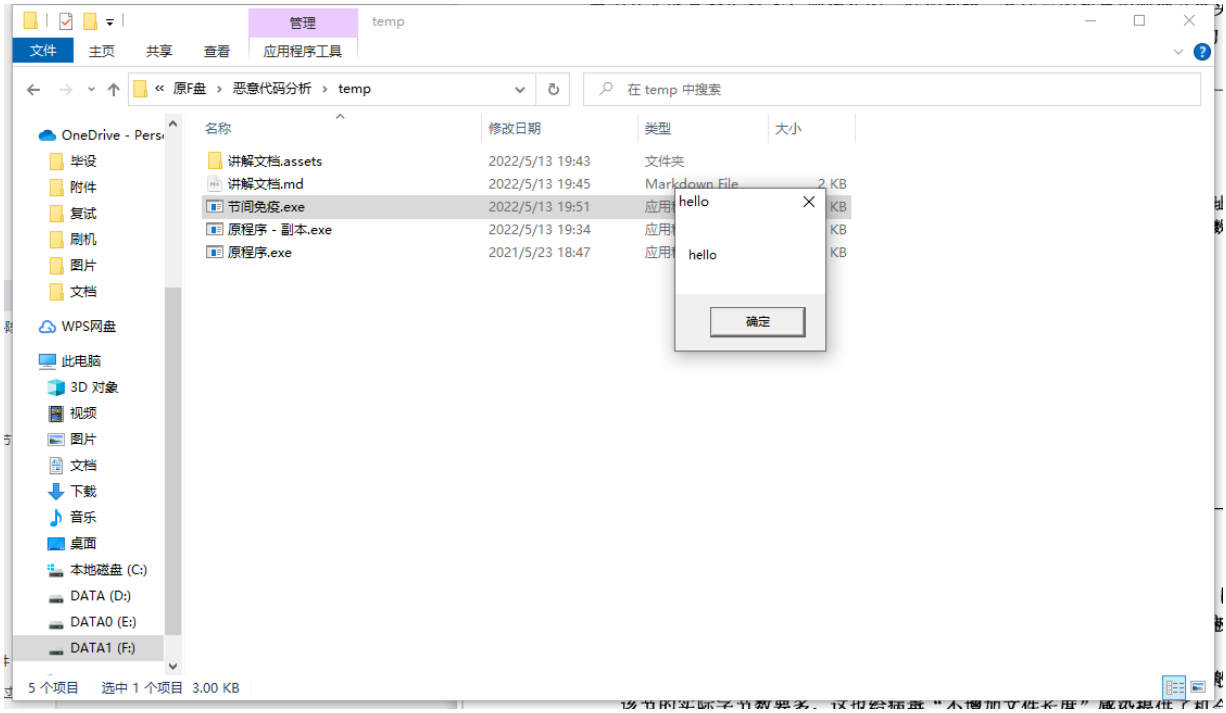


节间免疫

节间免疫的思想很简单，就只需要将每个节表的virtualSize修改为文件对齐大小，因为你的VirtualSize表示为这文件实际使用的字节，那如果让病毒发现你实际使用的字节已经占满这个节，那他就无法插入了。这也病毒代码向插入节间多的那部分就没有办法。



测试运行没有问题



代码实现

节表免疫

1. **方法1**

需要做的事情

1. 获取NT Header的大小，获取checkSum

```
DWORD optionHeaderSize = nt_header->FileHeader.SizeOfOptionalHeader;//扩展头大小
DWORD NTHeaderSize = optionHeaderSize + 24;//获取nt header大小
DWORD HeaderCheckSum = nt_header->OptionalHeader.CheckSum;    //PE头里的校验值
nt_header->OptionalHeader.CheckSum = 0;
```

   24是因为NT Header本身除了OptionHeader外，都是固定的，固定的那部分就24大小。

2. 获取要拷贝的偏移量

   第一个节的PointerToRawData:表示文件中第一个节的偏移，然后需要减去一个值，这个值表示了NT Header和section Header的大小。

   这个大小包含所有节的大小，因为节的大小是固定的，所以你只需要获取节的数量即可。包含刚刚获取的NTHeader，就是要拷贝的位置

3. 拷贝节表头和NT Header

```
memcpy((UINT8*)pMapping + PointerCopy, (UINT8*)nt_header, 40 * numOfSections+NTHeaderSize);
```

4. 修改el_lfanew

el_lfanew就是刚刚我没获取的要拷贝的偏移量，因为这个偏移量表示在文件中的偏移，和e_lfanew意义一致。

5. 修改checkSum

```
DWORD CheckSum = 0;      //计算下来的校验值
MapFileAndCheckSum(path2, &HeaderCheckSum, &CheckSum);
nt_header->OptionalHeader.CheckSum = CheckSum;
```

2. **方法2**

整体的意义是这样的。

所以你需要做的事情顺序是：

1. 找到扩展头大小，section header头指针，checkSum

```
DWORD optionHeaderSize = nt_header->FileHeader.SizeOfOptionalHeader;//扩展头大小
DWORD offsetOfFirst = optionHeaderSize + 216;//节表项开头的位置,因为optionHeader前有216字节，sectionHeader的位置
DWORD HeaderCheckSum = nt_header->OptionalHeader.CheckSum;    //PE头里的校验值
nt_header->OptionalHeader.CheckSum = 0;
```

这里216字节试固定的，因为你的DOS头加上NTHeader中OptionHeader前面一点一共是216。

校验值实际上是能直接置为0的，但是我们后续会处理

2. 获取section header拷贝的地址

第一个节的PointerToRawData:表示文件中第一个节的偏移，然后减去所有节的大小，因为节的大小是固定的，所以你只需要获取节的数量即可。

3. 拷贝section header

使用memcpy

4. 修改SizeOfOptionalHeader大小

起始就只需要你将要拷贝的地址减去section原本的地址，再把这部分增量加回`nt_header->FileHeader.SizeOfOptionalHeader`

5. 修改check sum

```
DWORD CheckSum = 0;      //计算下来的校验值
MapFileAndCheckSum(path2, &HeaderCheckSum, &CheckSum);
nt_header->OptionalHeader.CheckSum = CheckSum;
```

6. 将改动flush回去，记得关闭句柄，视图。

```
FlushViewOfFile(pMapping, 0);
```

节间免疫

修改VirtualSize的大小即可。