《API函数自搜索实验》实验报告

姓名：梁婧涵 学号：2112155 班级：1120

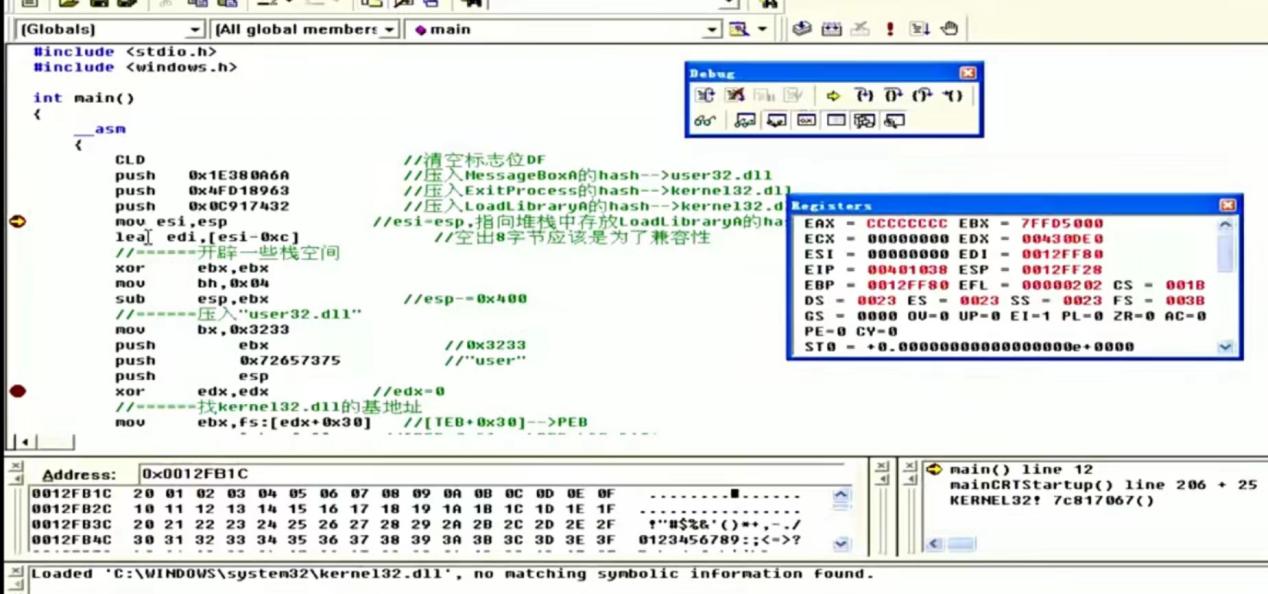
实验名称：API函数自搜索实验

实验要求：

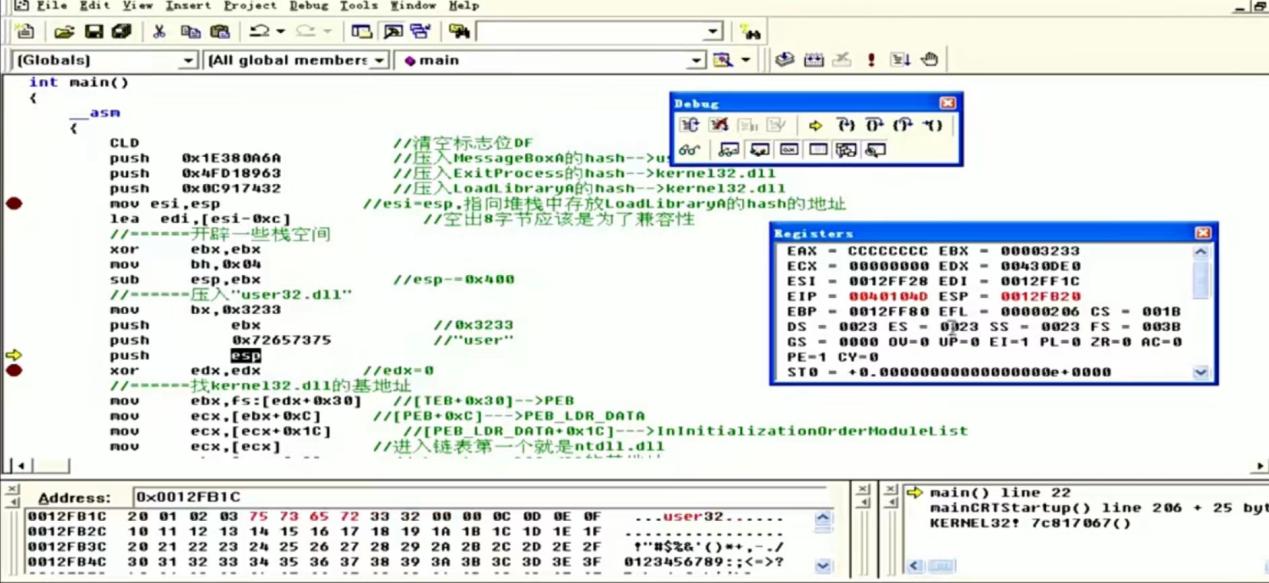
复现第五章实验七，基于示例5-11，完成API函数自搜索的实验，将生成的exe程序，复制到windows 10操作系统里验证是否成功

实验过程：

1. 定位kernel32.dll
2. 首先通过段寄存器FS在内存中找到当前的线程环境快
3. 线程环境快中偏移地址为0x30的地方存放着指向进程环境快的指针
4. 进程环境块中偏移地址为0x0c的地方存放着指向PEB\_LDR\_DATA结构体的指针，其中，存放着已经被进程装载的动态链接库的信息
5. PEB\_LDR\_DATA结构体中偏移地址为0x1C的地方存放着指向模块初始化链表的头文件InInitializationOrderModuleList
6. 模块初始化链表InInitializationOrderModuleList中按顺序存放着PE装入运行时初始化模块的信息，第一个链表结点是ntdll.dll，第二个链表结点是kernel32.dll
7. 找到属于kernel32.dll的结点后，在其基础上再偏移0x08就是kernel32.dll在内存中的加载基址



Push的是三个hush值，esi的值为0012FF28，bh影响的是ebx（04,00），exp抬高

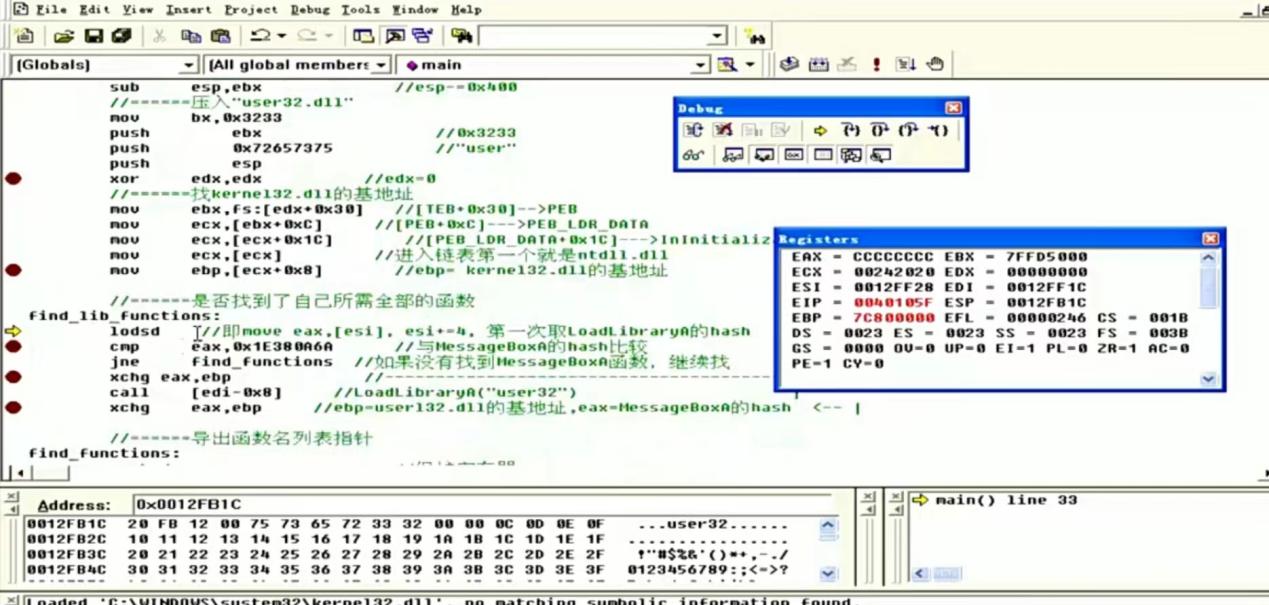


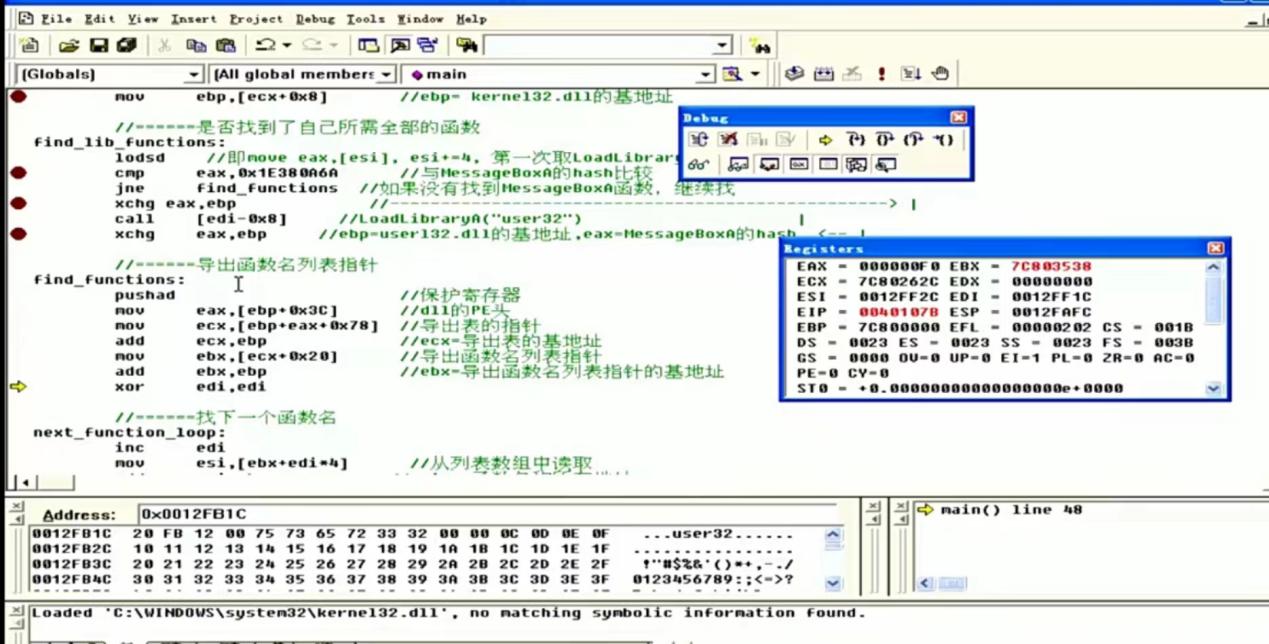
Esp再push变小 20 FB 12 00

1. 定位kernel32.dll导出表
2. 从kernel32.dll加载基址算起，偏移地址为0x3c的地方就是其PE头的指针
3. PE头中偏移地址为0x78的地方存放着指向函数导出表的指针
4. 获得导出函数地址为偏移地址（RVA）列表、导出函数名列表：

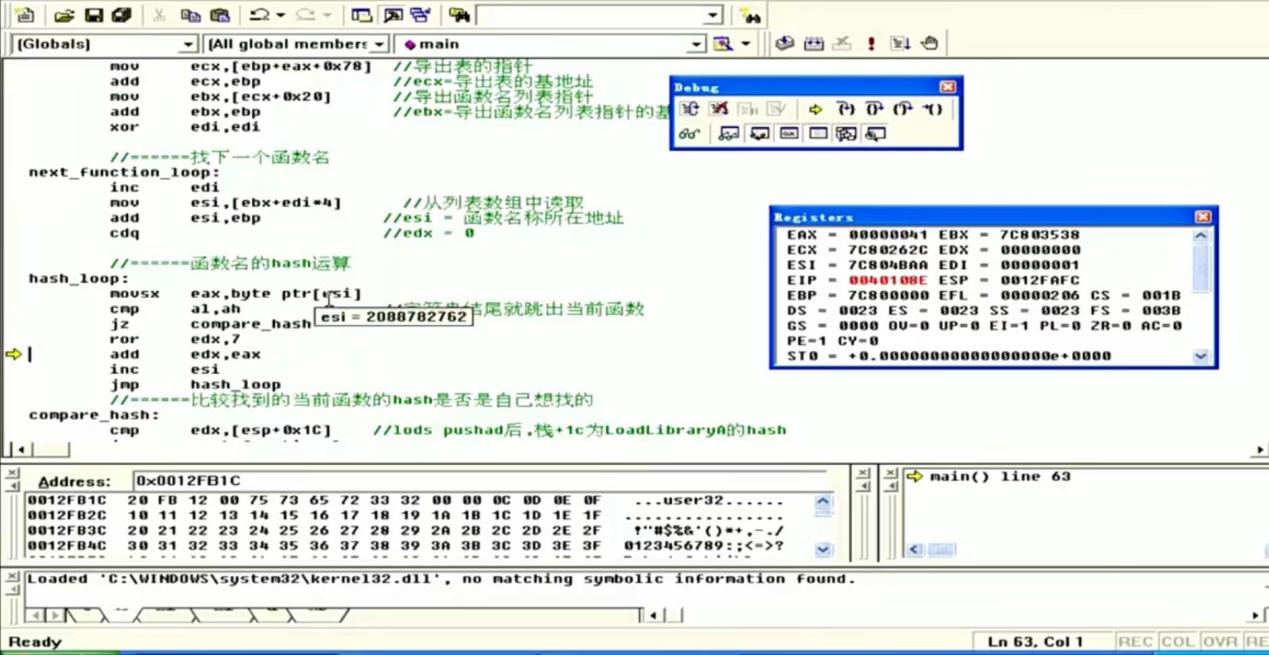
<1>导出表中偏移地址为0x1c处的指针指向存储导出函数偏移地址（RVA）的列表

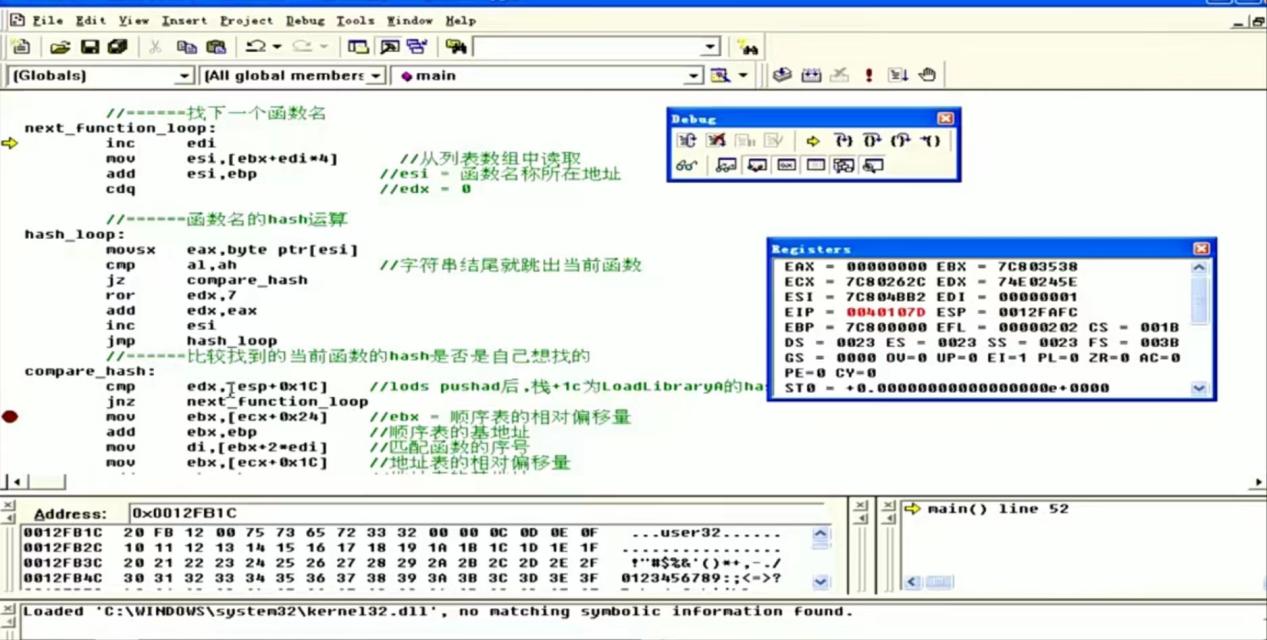
<2>导出表中偏移地址为0x20处的指针指向存储导出函数名的列表

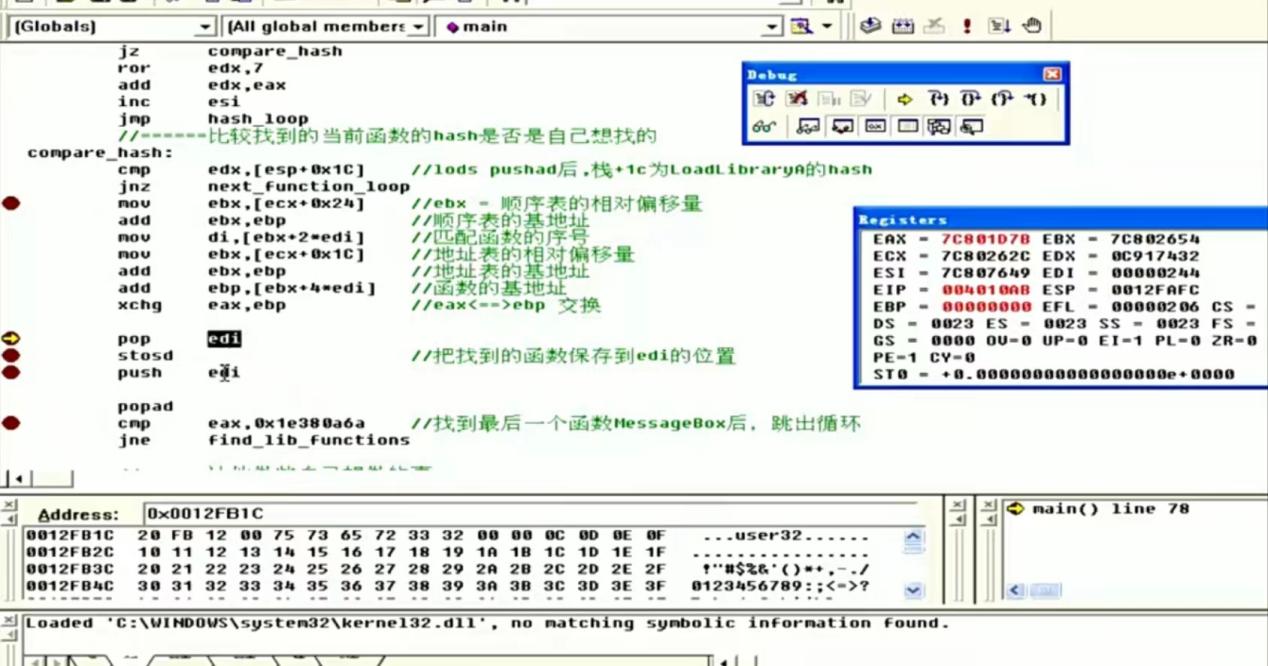


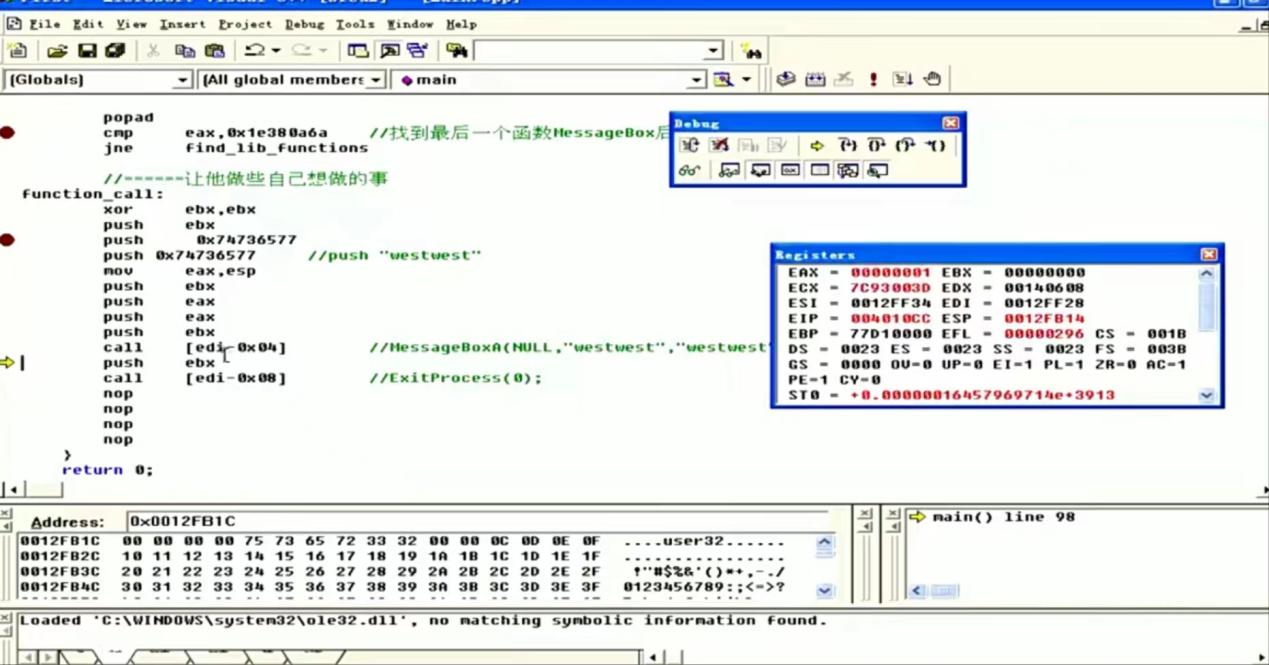


1. 搜索定位目标函数
2. 函数的RVA和名字按照顺序存放在上述两个列表中，我们可以在函数名的列表中定位到所需函数是第几个，然后再函数偏移地址的列表中找到对应的RVA
3. 获得RVA后，再加上前边已经得到的动态链表库的加载地址，就获得了所需API此刻在内存中的虚拟地址，这个地址就是最终在shellcode中调用时需要的地址。









心得体会：

为了让shellcode更加通用，能被大多数缓冲区容纳，所以希望shellcode尽可能短。因此，一般情况下并不会用MessageBoxA等这么长的字符串进行直接比较。所以会对所需API函数名进行hash运算，这样只要比较hash所得的摘要就能判断是不是我们所需的API了。

API函数自搜索实验的原理基于代码相似性匹配技术。该技术通过计算代码之间的相似度，识别出与查询代码最相似的一组代码集合，并从中推断出最可能的API函数。具体来说， API函数自搜索实验通常遵循以下步骤：

1.构建代码表示：将每个代码片段转换为相应的向量表示形式。通常，这涉及到使用一些自然语言处理和机器学习技术来提取代码特征，并将其转换为向量形式，以便进行比较和匹配。

2.计算代码相似度：使用某种相似度度量方法（如余弦相似度、Jaccard相似度等），计算查询代码和每个候选代码之间的相似度得分。

3.选择相似度最高的代码集合：从所有候选代码中，选择与查询代码最相似的代码集合。通常，这涉及到设置一个阈值，只选择相似度得分高于该阈值的代码。

4.推断API函数：对于每个候选代码集合，通过对其进行解析和分析，识别其中调用的API函数。通常，这需要结合API文档和代码语法，以确定调用的函数及其参数。

总体而言，API函数自搜索实验的原理基于代码相似性匹配和解析技术，它可以帮助程序员在编写代码时快速找到和使用合适的API函数，从而提高代码质量和开发效率。