**电子科技大学信息与软件工程学院**

**标 准 实 验 报 告**

**（实验）课程名称：软件安全**

**电子科技大学教务处制表**

**学生姓名： 学 号： 指导教师：**

**实验地点：信软学院**  **实验时间：**

**一、实验室名称：信软学院实验室**

**二、实验项目名称： 搜索API函数地址**

1. **实验学时：4学时**

**四、实验原理：**

编写shellcode时，一般需使用一些API函数，例如CreateProcess(),socket()等，这些函数的入口地址位于系统的动态链接库中，由于不同操作系统的动态链接库的加载地址不同，shellcode中需增加API函数自搜索功能。

API函数自搜索功能，主要依赖于通过系统数据结构找到相关子项的地址偏移，进而找到对应的库加载基地址和函数地址。

**五、实验目的：**

1）理解API函数搜索原理；

2）掌握搜索kernel32.dll地址的方法；

3）掌握搜索LoadLibrary（）及GetProcAddress()地址的方法；

**六、实验内容：**

1) 编程实现求动态链接库kernel32.dll的虚拟地址。

2）编程实现求LoadLibrary（）及GetProcAddress()虚拟地址。

1. **实验器材（设备、元器件）：**

PC机

* 1. **实验步骤：**

1. 定位kernel32.dll地址
   1. 通过段选择字FS在内存中找到当前的线程控制模块TEB
   2. 线程控制块中偏移位置为0x30的地方存放着指向进程控制块PEB的指针
   3. 进程控制块中偏移地址0x0c的地址存放着指向PEB\_LDA\_DATA结构体的指针，其中存放着已经被装载的动态链接库信息。
   4. PEB\_LDA\_DATA结构体偏移地址为0x1c的地方存放着指向模块初始化链表的头指针InInitializationOrderModulelist。
   5. 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信息，第一个链表节点是ntdll.dll，第二个链表节点就是kernel32.dll。
   6. 找到属于kernel32.dll的节点后，在此基础上再便宜0x08就是kernel32.dll在内存中的加载基地址。
2. 定位LoadLibrary（）及GetProcAddress()地址
   1. 从kernel32.dll的加载基地址开始偏移0x3c的地方就是其PE头
   2. PE头偏移0x78的地方存放着指向函数导出表的指针。
   3. 按以下方式在导出表中算出所需函数的入口地址

（1）导出表偏移0x1c处的指针指向存储导出函数偏移地址（RVA）的列表

（2）导出表偏移0x20处的指针指向存储导出函数函数名的列表

（3）函数的RVA地址和名称按顺序放在RVA列表及函数名列表中，根据函数名在函数名称列表中搜索函数序号，再根据函数序号在RVA列表中搜索函数对应的RVA

（4）函数对应的RVA加上动态链接库的加载地址得到该函数的虚拟地址。

**九、实验数据及结果分析：**

**（1） 输出**kernel32.dll，LoadLibrary（）和GetProcAddress()地址

**（2） 实验分析**

**十、实验结论：**

**//学生自行补充**

**十一、总结及心得体会：**

**//学生自行补充**

**十二、对本实验过程及方法、手段的改进建议：**

**//学生自行补充**

**报告评分：**

**指导教师签字：**