**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：陈奎 学 号：2017221303023 指导教师：钱伟中**

**实验地点：信软学院**  **实验时间：19.10.30**

**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：**

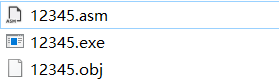
**实验代码**

|  |
| --- |
| .386  .model flat,stdcall  include msvcrt.inc  includelib msvcrt.lib  include user32.inc  includelib user32.lib  include kernel32.inc  includelib kernel32.lib  ASSUME FS:NOTHING  .data  s1 db "kernel32.dll base address:%x",0  s2 db "GetProcAddress() address:%x",0  s3 db "LoadLibrary() address:%x",0  s4 db "LoadLibrary",0  .code  start:  push ebp ;寄存器保护  mov ebp,esp  ;sub esp,0D8h  push ebx  push esi  push edi    mov eax,dword ptr fs:[30h] ;PEB  mov eax,dword ptr [eax+0ch]  mov esi,dword ptr [eax+1ch]  lodsd ;eax=InInitOrder.flink  mov ebp,[eax+8] ;ebp=kernel32.dll base address  mov edi, ebp ;同时保存kernel32.dll的基址到edi  invoke crt\_printf,offset s1,ebp    mov eax, [ebp+3Ch] ;eax = PE首部  mov edx,[ebp+eax+78h]  add edx,ebp ;edx = 引出表地址  mov ecx , [edx+18h] ;ecx = 输出函数的个数,确定循环次数  mov ebx,[edx+20h]  add ebx, ebp ;ebx ＝函数名地址，AddressOfName    searchGet:  dec ecx  mov esi,[ebx+ecx\*4]  add esi,ebp ;依次找每个函数名称  ;GetProcAddress  mov eax,50746547h  cmp [esi], eax; 'PteG'  jne searchGet  mov eax,41636f72h  cmp [esi+4],eax; 'Acor'  jne searchGet  ;如果是GetProcA，表示找到了  mov ebx,[edx+24h]  add ebx,ebp ;ebx = 序号数组地址,AddressOf  mov cx,[ebx+ecx\*2] ;ecx = 计算出的序号值  mov ebx,[edx+1Ch]  add ebx,ebp ;ebx＝函数地址的起始位置，AddressOfFunction  mov ebx,[ebx+ecx\*4]  add ebx,ebp ;利用序号值，得到出GetProcAddress的地址  invoke crt\_printf,offset s2,ebx      ;为局部变量分配空间  push ebp  sub esp, 50h  mov ebp, esp  ;查找LoadLibrary的地址  mov [ebp + 40h], ebx ;把GetProcAddress的地址保存到ebp + 40中  ;压入"LoadLibrary/0"的地址  mov eax,offset s4  push eax  push edi ;edi:kernel32的基址  call DWORD PTR[ebp + 40h] ;返回值(即LoadLibrary的地址)保存在eax中  invoke crt\_printf,offset s3,eax      pop edi  pop esi  pop ebx  mov esp,ebp  pop ebp  invoke ExitProcess,0h  end start |

**输出结果**



**编译链接文件**



**十、实验结论：**

利用PEB获得kernel32.dll的原理：

1.fs寄存器指向TEB结构

2.在TEB+0x30地方指向PEB结构

3.在PEB+0x0C地方指向PEB\_LDR\_DATA结构

4.在PEB\_LDR\_DATA+0x1C地方就是一些动态连接库地址了，如第一个指向ntdll.dll，第二个就是kernel32.dll的地址。

第二部分就是要动态获得函数地址：

动态获得函数地址，就是利用Kernel32.dll中的引出表。每个dll都有引出表，这样，外部程序可以调用dll里实现的函数，而不必关心实现的细节。而GetProcAddress是在kernel32.dll里实现的，所以我们可通过查找kernel32.dll的引出表来找到GetProcAddress函数。



我们查找函数地址时，先在函数名称数组中找到需要的函数名；然后在函数序号数组中得到对应的函数序号；最后根据这个序号，在函数地址数组中得到对应的地址值。

知道了kerner32.dll的基地址后，其PE头部偏移在kerner32.dll基址＋0x3C的地方；而输出表的位置在kerner32.dll基地址+PE头部地址+0x78的地方。

搜索函数地址流程：

a. 通过TEB/PEB获取kernel32.dll基址

b. 在(基址+0x3c)处获取e\_lfanewc就是PE标志。

c. 在(基址+e\_lfanew+0x78)处获取引出表地址(后面为描述方便简称export)

d. 在(基址+export+0x1c)处获取AddressOfFunctions、AddressOfNames、AddressOfNameOrdinalse。

e. 搜索AddressOfNames，确定“GetProcAddress”所对应的index

f. index = AddressOfNameOrdinalse [ index ];

g. 函数地址 = AddressOfFunctions [ index ];

最后调用GetProcAddress函数获得LoadLibrary的地址。

**十一、总结及心得体会：**

通过本次实验，成功找到了kernel32的基址以及GetProcAddress和LoadLibrary的地址。了解了API搜索函数地址的过程，对shellcode的编写有了简单的理解。

**十二、对本实验过程及方法、手段的改进建议：**

无

**报告评分：**

**指导教师签字：**