《漏洞利用及渗透测试基础》实验报告

姓名：蒋薇　　学号：2110957　班级：张建老师班

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 操作系统验证是否成功。

实验过程：

完成 API 函数自搜索，
生成的 exe 程序复制到 windows 10 操作系统验证

通用机制，动态计算调用地址，获取加载后地址，计算函数入口地址
MessageBox()弹出对话框，先调用 loadlibrary(),检索 kernel32.dll 中 LoadLibrary()地址、ExitProcess（）地址，进一步检索 user32.dll 里 MessageBox()地址，依次完成 LoadLibrary()调用，MessageBox()调用，ExitProcess（）调用

//开辟栈空间
//压入“user32.dll”
//“找 kernel32.dll 的基地址”
//是否找到了自己所需的全部函数

```
#include <stdio.h>
#include <windows.h>

int main()
{
    __asm
    {
        CLD                         //清空标志位DF
        push    0x1E380A6A          //压入MessageBoxA的hash-->user32.dll
        push    0x4FD18963          //压入ExitProcess的hash-->kernel32.dll
        push    0x0C917432          //压入LoadLibraryA的hash-->kernel32.dll
        mov esi,esp               //esi-esp,指向堆栈中存放LoadLibraryA的hash的地址
        lea  edi,[esi-0xc]            //空出8字节应该是为了兼容性
        //------开辟一些栈空间
        xor     ebx,ebx
        mov     bh,0x04
        sub     esp,ebx             //esp-=0x400
        //------压入"user32.dll"
        mov     bx,0x3233
        push        ebx                 //0x3233
        push        0x72657375          //"user"
        push        esp
        xor     edx,edx           //edx=0
        //------找kernel32.dll的基地址
        mov     ebx,fs:[edx+0x30]   //[TEB+0x30]-->PEB
        mov     ecx,[ebx+0xC]     //[PEB+0xC]--->PEB_LDR_DATA
        mov     ecx,[ecx+0x1C]        //[PEB_LDR_DATA+0x1C]--->InInitializationOrderModuleLis
```

```
#include <stdio.h>
#include <window.h>
int main(){
_asm
{
CLD
push...
...
}
```

```
Push...//压入 MessageBoxA()的 hash-->user32.dll
       //压入 ExitProcess()的 hash-->kernel32.dll
       //压入 LoadLibraryA()的 hash-->kernel32.dll
    Mov esi,esp //esi -esp,指向堆栈中存放 LoadLibraryA 的 hash()

Xor edx.edx //edx = 0

Mov ebp,[ecx + 0x8] //ebp = kernel32.dll 的基地址

find_lib_functions:
        Lodsd  //即 move eax,[esi],esi+=4,第一次取 LoadLibraryA 的 hash
        move eax,0x1E380A6A //与 MessageBoxA 的 hash 比较
        Jne find_Functions//如果没有找到 MessageBoxA 函数，继续找
        Xchg eax,ebp
        Call [edi - 0x8]//Load Library A（“user”）
        Xchg eax,ebp //ebp-user32.dll 的基地址，eax-MessageBoxA 的 hash
```

```
        stosd                                   //把找到的函数保
        push    edi

        popad
        cmp     eax,0x1e380a6a      //找到最后一个函
        jne     find_lib_functions

        //------让他做些自己想做的事
function_call:
        xor     ebx,ebx
        push    ebx
        push     0x74736577
        push 0x74736577         //push "westwest"
        mov     eax,esp
        push    ebx
        push    eax
        push    eax
        push    ebx
        call    [edi-0x04]              //MessageBoxA(NU
        push    ebx
        call    [edi-0x08]              //ExitProcess(0)
        nop
        nop
        nop
        nop
    }
    return 0;
```

//导出函数名列表指针
find_Functons:

//找下一个函数名
next_function_loop:

//函数名的 hash 运算
Hash_loop:

//比较找到的当前函数的 hash 是否是自己想找到
compare_hash:

Function_call:
Push”westwest”;
MessageBoxA(NULL,”westwest”,”westwest”)
ExitProcess(0);//程序健壮性，退出

综上，验证成功。

心得体会：

API 函数自搜索技术:编写通用 shellcode,shellcode 自身具备动态的自动搜索所需 API 函数地址的能力
调用 MessageBoxA()函数，使用 LoadLibrary(“user32.dll)装载 user32.dll,
定位 LoadLibrary()函数：
①定位 kernel32.dll
②解析 kernel32.dll 导出表
③搜索定位 LoadLibrary 等目标函数
④基于找到的函数地址，完成 Shellcode 的编写