汇编语言与逆向技术实验报告

Lab4 读取PE文件的输入表和输出表

姓名：丁彦添 学号：1911406

一、描述PE文件的输入表的作用和数据结构

作用

输入表保存函数名和其驻留的 DLL 名等动态链接所需的信息， 记录了程序正在使用哪些库中的哪些函数。

数据结构

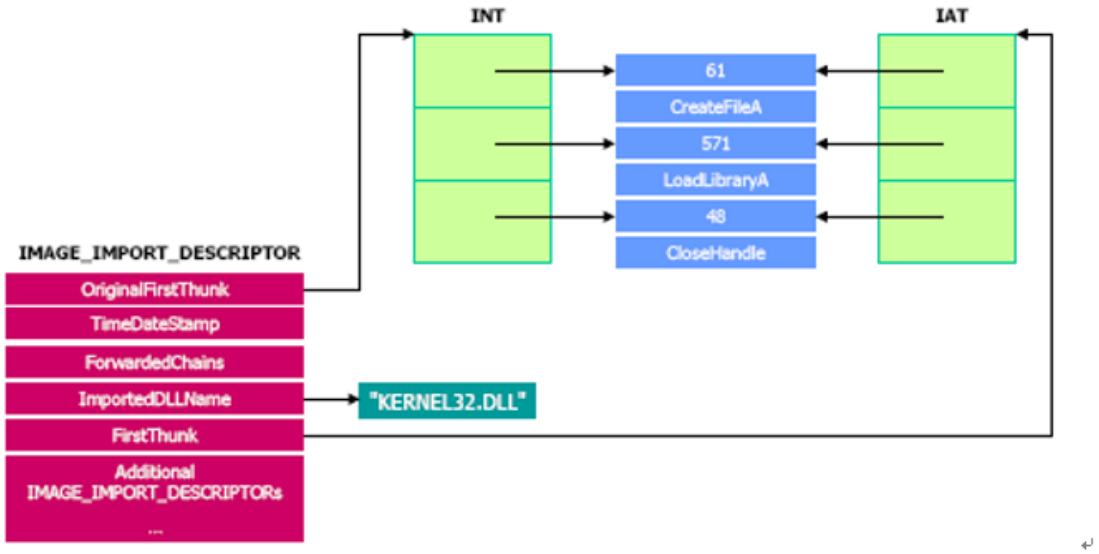

图 1 输入表结构

二、描述PE文件的输出表的作用和数据结构

作用

作用： DLL 文件通过输出表向系统提供输出函数名、序号和入口地址等信息。

数据结构

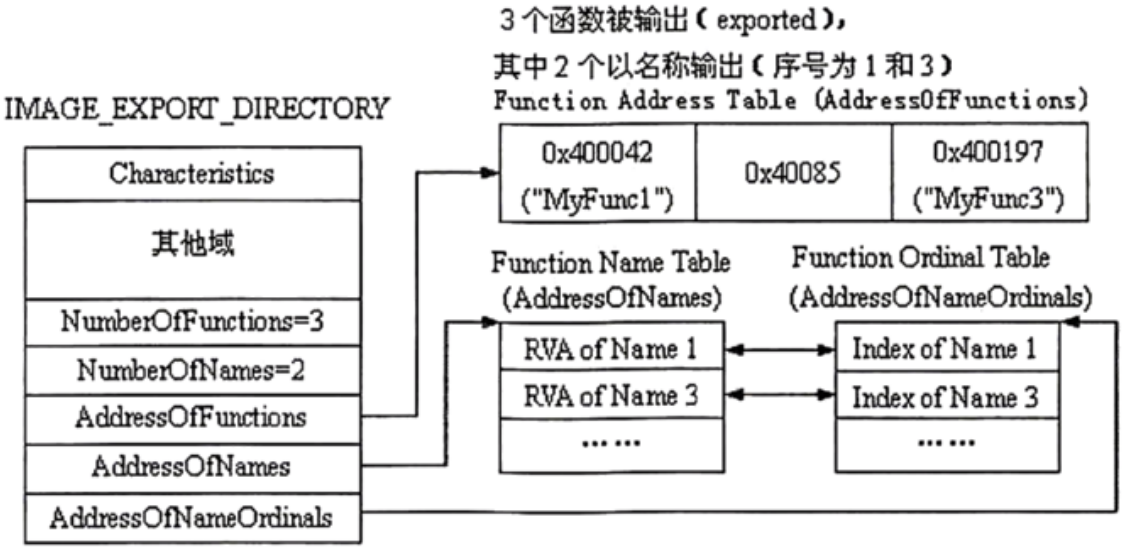

图 2 输出表的数据结构

三、程序的汇编语言源代码和注释

```
.386
.MODEL flat , stdcall
OPTION casemap :none
INCLUDE .\masm32\include\windows.inc
INCLUDE .\masm32\include\kernel32.inc
INCLUDE .\masm32\include\masm32.inc
INCLUDELIB .\masm32\lib\kernel32.lib
INCLUDELIB .\masm32\lib\masm32.lib
.DATA
    txt_input BYTE "Please input a PE file :", 0
    txt_iat BYTE "Import table : " , 0
    txt_eat BYTE "Export table : " , 0
    txt_tab BYTE "  ", 0
    txt_nl BYTE 0Dh, 0Ah, 0
    filename BYTE 10h DUP(0) , 0
    hfile DWORD 0
    raw_e DWORD 3Ch ; e_lfanew 的地址
    RVA_ini DWORD 0
    rva_iat DWORD 80h
    rva_eat DWORD 78h
    rva_proc DWORD 0
    tmp DWORD 0
    tmp2 DWORD 0
    cnt DWORD 0
    buf  DWORD 20000h DUP(0); 测试用文件较大
    buf1 DWORD 20000h DUP(0)
    buf2 DWORD 20000h DUP(0)
    buf3 DWORD 20000h DUP(0)
    buf4 DWORD 20000h DUP(0)
.CODE
main PROC
    INVOKE StdOut , ADDR txt_input
    INVOKE StdIn , ADDR filename , 10h
    INVOKE CreateFile , ADDR filename , GENERIC_READ, FILE_SHARE_READ,\
    0 , OPEN_EXISTING, FILE_ATTRIBUTE_ARCHIVE, 0
```

```
    MOV hfile , EAX
    INVOKE SetFilePointer ,hfile, 0, 0, FILE_BEGIN
    INVOKE ReadFile , hfile, ADDR buf, 0A0000h, 0, 0
    ; 读取文件
    MOV EAX, raw_e
    MOV EBX, buf[EAX]
    MOV RVA_ini, EBX
    ADD RVA_ini, 104h
    ; ”PE” 的地址加 104h 得到第一个节区头的 RVA 一项
    ADD rva_iat, EBX
    MOV EAX, rva_iat
    MOV EAX, buf[EAX]
    MOV rva_iat, EAX
    ADD rva_eat, EBX
    MOV EAX, rva_eat
    MOV EAX, buf[EAX]
    MOV rva_eat, EAX
    ; 可选头中 IAT 和 EAT 的 RVA 值
    INVOKE StdOut, ADDR txt_iat
    INVOKE StdOut, ADDR txt_nl
    CMP rva_iat, 0
    JE Lf ; 假如没有 IAT 就跳过
    ADD rva_iat, 0Ch ;Name +C
La:
    MOV EAX, rva_iat
    CALL rva2raw ;Name 的地址
    MOV EAX, buf[EAX]
    CALL rva2raw ; dll 名
    MOV tmp, EAX
    INVOKE StdOut, ADDR txt_tab
    MOV EAX, tmp
    INVOKE StdOut, ADDR buf[EAX]
    INVOKE StdOut, ADDR txt_nl
    ; 输出 dll 名
    MOV EAX, rva_iat
    ADD EAX, 4h ;IAT 的地址
    CALL rva2raw
    MOV EAX, buf[EAX]
    CALL rva2raw
    MOV rva_proc, EAX
Lb:
    MOV EAX, rva_proc
    MOV EAX, buf[EAX]
    CALL rva2raw
    MOV tmp, EAX
    INVOKE StdOut, ADDR txt_tab
    INVOKE StdOut, ADDR txt_tab
    MOV EAX, tmp
    INVOKE StdOut, ADDR buf[EAX+2h] ; 跳过 Hint
    INVOKE StdOut, ADDR txt_nl
    ADD rva_proc , 4h
    MOV EAX, rva_proc
    CMP buf[EAX], 0
    JNE Lb
    ; 循环输出函数
    ADD rva_iat, 14h
    MOV EAX, rva_iat
    CALL rva2raw
```

```
    CMP buf[EAX] , 0
    JNE La ; 遍历 IID
    ; 循环输出 dll
Lf:
    INVOKE StdOut, ADDR txt_eat
    INVOKE StdOut, ADDR txt_nl
    CMP rva_eat , 0
    JE Ld ; 假如没有 EAT 就跳过
    MOV EAX, rva_eat
    MOV cnt , EAX
    ADD cnt , 18h ;NoN +18
    MOV EAX, cnt
    CALL rva2raw
    MOV EAX, buf[EAX]
    MOV cnt , EAX
    ADD rva_eat , 20h ;NT +20
    MOV EAX, rva_eat
    CALL rva2raw
    MOV EAX, buf[EAX]
    CALL rva2raw
    MOV tmp, EAX
Lc:
    MOV EAX, tmp
    MOV EAX, buf[EAX]
    CALL rva2raw
    MOV tmp2, EAX
    INVOKE StdOut, ADDR txt_tab
    MOV EAX, tmp2
    INVOKE StdOut, ADDR buf[EAX]
    INVOKE StdOut, ADDR txt_nl
    ADD tmp, 4h
    DEC cnt
    CMP cnt , 0
    JNE Lc
    ; 循环输出函数
Ld:
    INVOKE ExitProcess , 0
    main ENDP
    rva2raw PROC
    MOV EBX, RVA_ini
L1:
    ADD EBX, 28h
    CMP buf [EBX] , 0
    JE L2 ; 判断是否是最后一个节区头
    CMP buf [EBX] , EAX
    JB L1 ; 判断下一个节区头
L2:
    SUB EBX, 28h
    SUB EAX, buf[EBX]
    ADD EBX, 8h
    ADD EAX, buf[EBX]
    RET
rva2raw ENDP
; rva 转 raw , 借助 EAX 传参
END main
```

四、程序测试时，输出结果的截图

使用命令对import_export.asm进行编译成import_export.obj。

```
D:\dyt\Studie\5Junior\D_Assemblersprache\Zuordnung\lab4>.\masm32\bin\ml /c /coff import_export.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997.  All rights reserved.

 Assembling: import_export.asm

**********
ASCII build
**********
```

再将其链接成import_export.exe。

```
D:\dyt\Studie\5Junior\D_Assemblersprache\Zuordnung\lab4>.\masm32\bin\link /SUBSYSTEM:CONSOLE import_export.obj
Microsoft (R) Incremental Linker Version 5.12.8078
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.
```

在命令行打开，输入测试样例hello.exe，得到结果如下：

```
D:\dyt\Studie\5Junior\D_Assemblersprache\Zuordnung\lab4>import_export.exe
Please input a PE file :hello.exe
Import table :
    kernel32.dll
        GetStdHandle
        WriteFile
        ExitProcess
Export table :
    start
```

五、讨论输入表的安全问题（选做）

输入表容易被强行篡改使得程序不能按照原本的代码正常执行。