汇编语言与逆向技术实验报告

Lab4-peviewer

学号：2112514　姓名：辛浩然　专业：信息安全、法学

一、 实验目的

1.熟悉 PE 文件结构；

2.使用 Windows API 函数读取文件内容。

二、 实验内容

1.输入 PE 文件的文件名，peviewer 程序调用 Windows API 函数，打开指定的 PE 文件；

2.从文件的头部开始，读取 IMAGE_DOS_HEADER 结构中的 e_magic 和 e_lfanew 字段的值，按照实验演示的方式输出到命令行窗口；

3.继续读取 PE 文件的 IMAGE_NT_HEADER 结构中的 Signature 字段的值，按照实验演示的方式输出到命令行窗口；

4.继续读取 IMAGE_NT_HEADER 结构中的 IMAGE_FILE_HEADER 结构，从中读取出字段 NumberOfSections、TimeDateStamp、Characteristics 的值，按照实验演示的方式输出到命令行窗口；

5.继续读取 IMAGE_NT_HEADER 结构中的 IMAGE_OPTIONAL_HEADER 结构，从中读取字段 AddressOfEntryPoint、ImageBase、SectionAlignment、FileAlignment 的值，按照实验演示的方式输出到命令行窗口；

三、 设计思路与控制流图

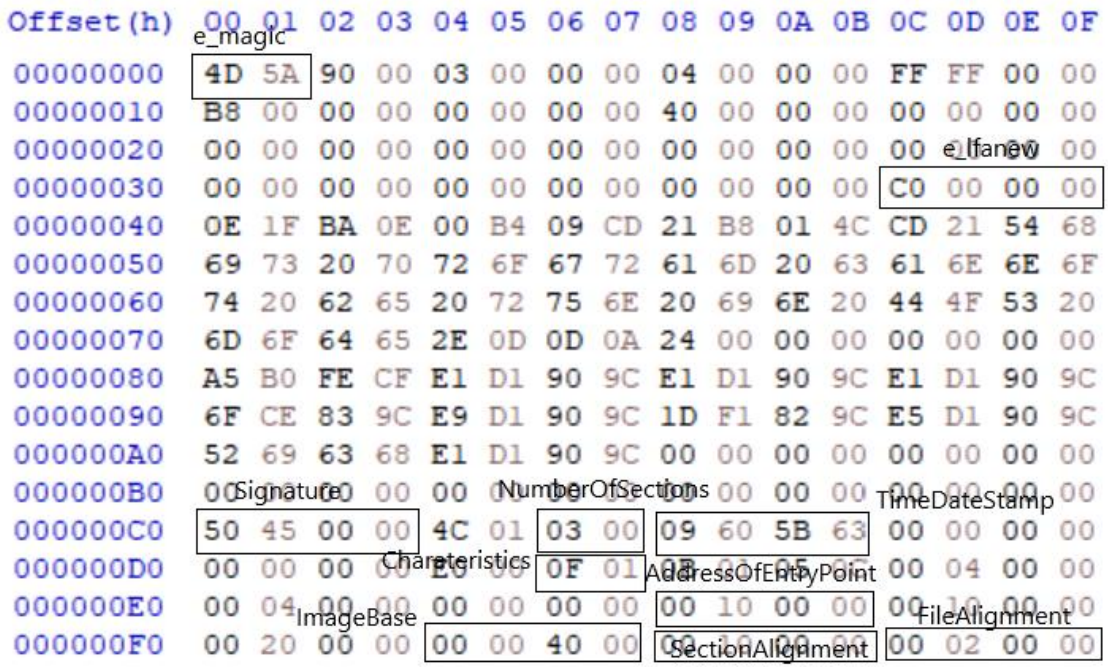
1.主要思路：

使用 CreateFile 函数打开指定文件，返回文件句柄。

使用 SetFilePointer 函数设置句柄的位置，可以从文件头设置偏移量，也可以从当前位置设置偏移量。

使用 ReadFile 函数向后读取指定的字节数，同时句柄也向后移动。

2.具体实现：



读取 e_magic：句柄位于文件头，向后读取 2 字节；

读取 e_lfanew：句柄从当前位置向后偏移 3Ah 字节，然后向后读取 4 字节；

读取 Signature：句柄从文件头开始向后偏移 e_lfanew，然后向后读取 4 字节；

读取

读取 NumberOfSections：句柄从当前位置向后偏移 2 字节，然后向后读取 2 字节；

读取 TimeDateStamp：句柄从当前位置向后偏移 0 字节，然后向后读取 4 字节；

读取 Characteristics：句柄从当前位置向后偏移 10 字节，然后向后读取 2 字节；

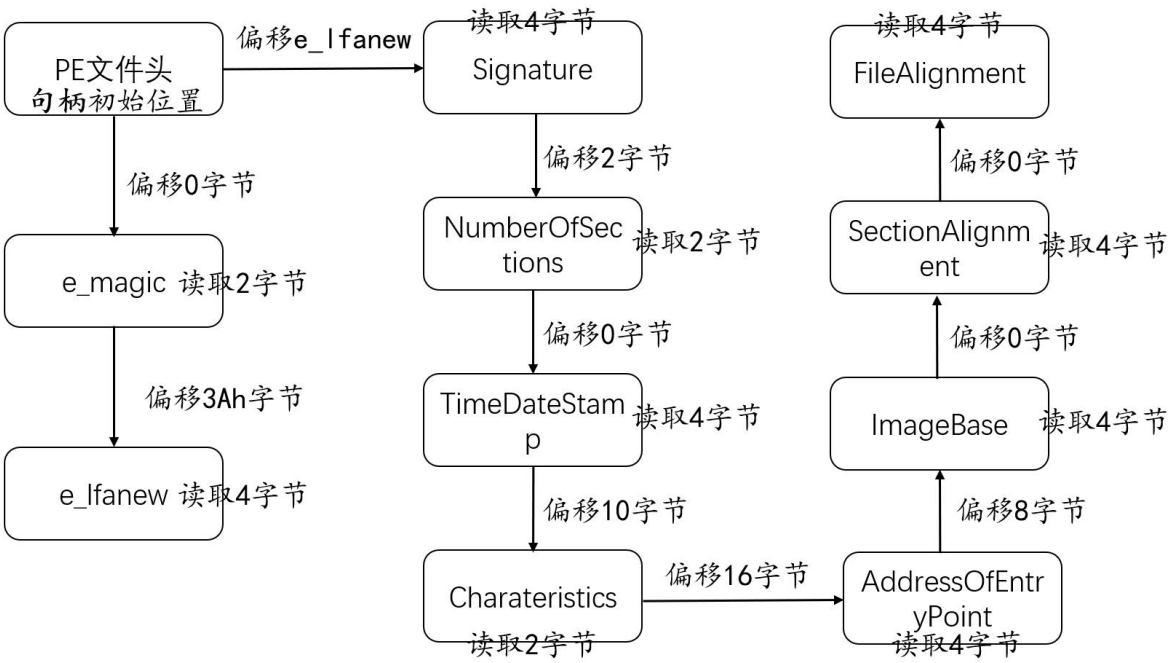
读取 AddressOfEntryPoint：句柄从当前位置向后偏移 16 字节，然后向后读取 4 字节；

读取 ImageBase：句柄从当前位置向后偏移 8 字节，然后向后读取 4 字节；

读取 SectionAlignment：句柄从当前位置向后偏移 0 字节，然后向后读取 4 字节；

读取 FileAlignment：句柄从当前位置向后偏移 0 字节，然后向后读取 4 字节；

3.控制流图



注：箭头表示句柄移动轨迹，读取字节的同时句柄也移动相应字节，箭头旁偏移字节数是在完成上步读取后句柄继续偏移的字节数。

四、 实验代码及注释

```
.386
.model flat, stdcall
option casemap:none
include D:\masm32\include\windows.inc
include D:\masm32\include\kernel32.inc
include D:\masm32\include\masm32.inc
includelib D:\masm32\lib\kernel32.lib
includelib D:\masm32\lib\masm32.lib

.data
        file_name byte 50 dup(0)
        str1      byte "Please input a PE file: ",0
        str2      byte "IMAGE_DOS_HEADER",0Ah,0Dh,0
        str3      byte "   e_magic: ",0
        str4      byte "   e_lfanew: ",0
        str5      byte "IMAGE_NT_HEADER",0Ah,0Dh,0
        str6      byte "   Signature: ",0
        str7      byte "IMAGE_FILE_HEADER",0Ah,0Dh,0
        str8      byte "   NumberOfSections: ",0
        str9      byte "   TimeDateStamp: ",0
        str10     byte "   Characteristics: ",0
        str11     byte "IMAGE_OPTIONAL_HEADER",0Ah,0Dh,0
```

```
        str12     byte "   AddressOfEntryPoint: ",0
        str13     byte "   ImageBase: ",0
        str14     byte "   SectionAlignment: ",0
        str15     byte "   FileAlignment: ",0
        str_      byte 0Ah,0Dh,0
.data?
        hfile     dword ?
        buf1      dword ?
        buf2      dword ?
.code
    start:
          invoke StdOut,addr str1
          invoke StdIn,addr file_name,50
          invoke StdOut,addr str2
          invoke StdOut,addr str3

          invoke CreateFile,addr
file_name,GENERIC_READ,FILE_SHARE_READ,0,OPEN_EXISTING,FILE_ATTRIBUTE_ARCHIVE,
0
    ;读文件，这里的参数分别为：指向文件名的指针、允许对设备进行读访问、允许对文件进行
共享访问、不指定指向 SECURITY_ATTRIBUTES 结构的指针、文件必须已经存在标记、归档属性、
不指定文件句柄
          mov    hfile,eax
    ;CreateFile 函数返回一个打开的指定文件的句柄到 eax 中，将句柄存到 hfile 里
          invoke SetFilePointer,hfile,0,0,FILE_BEGIN
    ;SetFilePointer 在打开的文件中设置新的读取位置，四个参数分别为：文件句柄、偏移量
（低位）、偏移量（高位）、基准位置
    ;FILE_BEGIN 是从文档头开始，e_magic 就在文档头，所以句柄偏移 0 字节
          invoke ReadFile,hfile,addr buf1,2,0,0
    ;readfile 从文件指针指向的位置开始将数据读出到一个文件中，五个参数分别为：文件的
句柄、用于保存读入数据的一个缓冲区、要读入的字符数、指向实际读取字节数的指针、OVERLAPPED
结构体指针
    ;因为 e_magic 只有 2 个字节，所以只从句柄处向后移动读取两个字节，将读到的存至 buf1
中。在读取的过程中，句柄也在向后移动。
          mov    ebx,dword ptr buf1
          invoke dw2hex,ebx,addr buf2
    ;de2hex 的参数是 32 位操作数的地址，将 dword 数据转成 16 进制字符串
          invoke StdOut,addr buf2+4
    ;buf2 前四个字节为 0000，后四个字节是 5A4D，所以偏移四个字节输出
          invoke StdOut,addr str_
          invoke StdOut,addr str4

          invoke SetFilePointer,hfile,3Ah,0,FILE_CURRENT
    ;FILE_CURRENT 是从文件现在的位置开始
```

```
    ;e_lfanew 在文档头后 3Ch 处，在文件读完 2 个字节 e_magic 后，句柄位于文件头后 2 字节处，还需偏移 3Ah，所以句柄偏移 3Ah 字节
        invoke ReadFile,hfile,addr buf1,4,0,0
    ;e_lfanew 占 4 个字节，向后读 4 个字节
        mov    ebx,dword ptr buf1
        invoke dw2hex,ebx,addr buf2
    ;将 dword 数据转成 16 进制字符串
        invoke StdOut,addr buf2
    ;输出 buf2
        invoke StdOut,addr str_
        invoke StdOut,addr str5
        invoke StdOut,addr str6

        invoke SetFilePointer,hfile,buf1,0,FILE_BEGIN
    ;IMAGE_DOS_HEADER 结构的 e_lfanew 字段定位 PE Header 的起始偏移量，加上基址，得到 PE 文件头的指针
    ;buf1 中存放的即为 PE Header 的起始偏移量
    ;FILE_BEGIN 是从文档头开始，句柄向后偏移 buf1，这样句柄就到 PE header 处，PE header 的前四个字节即为 Signature
        invoke ReadFile,hfile,addr buf1,4,0,0
    ;Signature 占 4 个字节，向后读 4 个字节
        mov    eax,dword ptr buf1
        invoke dw2hex,eax,addr buf2
    ;将 dword 数据转成 16 进制字符串
        invoke StdOut,addr buf2
    ;输出 buf2
        invoke StdOut,addr str_
        invoke StdOut,addr str7
        invoke StdOut,addr str8

        invoke SetFilePointer,hfile,2,0,FILE_CURRENT
    ;FILE_CURRENT 是从文件现在的位置开始
    ;经过上述操作，句柄位于 signature 后的一个字节处，NumberOfSections 在 Signature 后 2 字节处，所以句柄向后偏移 2 个字节
        invoke ReadFile,hfile,addr buf1,2,0,0
    ;NumberOfSections 占 2 个字节，向后读取 2 字节
        mov    eax,dword ptr buf1
        invoke dw2hex,eax,addr buf2
    ;将 dword 数据转成 16 进制字符串
        invoke StdOut,addr buf2+4
    ;输出 buf2 后四个字节
        invoke StdOut,addr str_
        invoke StdOut,addr str9
```

```
        invoke SetFilePointer,hfile,0,0,FILE_CURRENT
    ;FILE_CURRENT 是从文件现在的位置开始
    ;TimeDateStamp 在 NumberOfSections 之后，所以无需偏移，从现在的位置往后读就可以
        invoke ReadFile,hfile,addr buf1,4,0,0
    ;TimeDateStamp 占 4 个字节，向后读取 4 个字节
        mov    eax,dword ptr buf1
        invoke dw2hex,eax,addr buf2
    ;将 dword 数据转成 16 进制字符串
        invoke StdOut,addr buf2
    ;输出 buf2
        invoke StdOut,addr str_
        invoke StdOut,addr str10

        invoke SetFilePointer,hfile,10,0,FILE_CURRENT
    ;FILE_CURRENT 是从文件现在的位置开始
    ;经过上述操作，句柄位于 TimeDateStamp 后的一个字节处，需要向后偏移 10 个字节，到达
Characteristics 位置
        invoke ReadFile,hfile,addr buf1,2,0,0
    ;Characteristics 占 2 个字节，向后读取 2 字节
        mov    eax,dword ptr buf1
        invoke dw2hex,eax,addr buf2
    ;将 dword 数据转成 16 进制字符串
        invoke StdOut,addr buf2+4
    ;输出 buf2 后四个字节
        invoke StdOut,addr str_
        invoke StdOut,addr str11
        invoke StdOut,addr str12

        invoke SetFilePointer,hfile,16,0,FILE_CURRENT
    ;FILE_CURRENT 是从文件现在的位置开始
    ;经过上述操作，句柄位于 Characteristics 后的一个字节处，需要向后偏移 16 个字节，到
达 AddressOfEntryPoint 位置
        invoke ReadFile,hfile,addr buf1,4,0,0
    ;AddressOfEntryPoint 占 4 个字节，向后读取 4 个字节
        mov    eax,dword ptr buf1
        invoke dw2hex,eax,addr buf2
    ;将 dword 数据转成 16 进制字符串
        invoke StdOut,addr buf2
    ;输出 buf2
        invoke StdOut,addr str_
        invoke StdOut,addr str13

        invoke SetFilePointer,hfile,8,0,FILE_CURRENT
```

```
    ;FILE_CURRENT 是从文件现在的位置开始
    ;经过上述操作，句柄位于 AddressOfEntryPoint 后的一个字节处，需要向后偏移 8 个字节，
到达 ImageBase 位置
          invoke ReadFile,hfile,addr buf1,4,0,0
    ;ImageBase 占 4 个字节，向后读取 4 个字节

          mov    eax,dword ptr buf1
          invoke dw2hex,eax,addr buf2
    ;将 dword 数据转成 16 进制字符串
          invoke StdOut,addr buf2
    ;输出 buf2
          invoke StdOut,addr str_
          invoke StdOut,addr str14

          invoke SetFilePointer,hfile,0,0,FILE_CURRENT
    ;FILE_CURRENT 是从文件现在的位置开始
    ;SectionAlignment 紧跟在 ImageBase 之后，所以句柄无需偏移
          invoke ReadFile,hfile,addr buf1,4,0,0
    ;SectionAlignment 占 4 个字节，向后读取 4 个字节
          mov    eax,dword ptr buf1
          invoke dw2hex,eax,addr buf2
    ;将 dword 数据转成 16 进制字符串
          invoke StdOut,addr buf2
    ;输出 buf2
          invoke StdOut,addr str_
          invoke StdOut,addr str15

          invoke SetFilePointer,hfile,0,0,FILE_CURRENT
    ;FILE_CURRENT 是从文件现在的位置开始
    ;FileAlignment 紧跟在 SectionAlignment 之后，所以句柄无需偏移
          invoke ReadFile,hfile,addr buf1,4,0,0
    ;FileAlignment 占 4 个字节，向后读取 4 个字节
          mov    eax,dword ptr buf1
          invoke dw2hex,eax,addr buf2
    ;将 dword 数据转成 16 进制字符串
          invoke StdOut,addr buf2
    ;输出 buf2 后四个字节
          invoke CloseHandle,hfile
    ;关闭打开的对象句柄
 end start
```

五、 实验截图

```
D:\汇编与逆向>\masm32\bin\ml /c /Zd /coff peviewer.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997.  All rights reserved.

 Assembling: peviewer.asm

**********
ASCII build
**********


D:\汇编与逆向>\masm32\bin\link /SUBSYSTEM:CONSOLE peviewer.obj
Microsoft (R) Incremental Linker Version 5.12.8078
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.


D:\汇编与逆向>.\peviewer.exe
Please input a PE file: bubble.exe
IMAGE_DOS_HEADER
   e_magic: 5A4D
   e_lfanew: 000000C0
IMAGE_NT_HEADER
   Signature: 00004550
IMAGE_FILE_HEADER
   NumberOfSections: 0003
   TimeDateStamp: 635B6009
   Characteristics: 010F
IMAGE_OPTIONAL_HEADER
   AddressOfEntryPoint: 00001000
   ImageBase: 00400000
   SectionAlignment: 00001000
   FileAlignment: 00000200
```

附上使用 HxD 对该文件的读取：