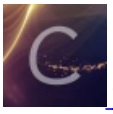
获取PE文件的区段表

原创

sysprogram 于 2011-05-08 14:32:00 发布 3841 收藏 1
分类专栏： C/C++/MFC PE文件格式 文章标签： image null header file delete dos



获取PE文件的区段表，用的方法是，首先打开CreateFile，然后读取这个文件的DosHeader，

从DosHeader中取e_lfanew这个成员的值，这样就能知道“PE00”的偏移，然后SetFilePointer文件的指针

到e_lfanew + sizeof(IMAGE_NT_SIGNATURE)这个地址，读取FileHeader结构

，从文件头结构中获取NumberOfSections区段的个数，根据区段个数分配够用的缓冲区，

最后SetFilePointer将文件指针从当前位置偏移FileHeader.SizeOfOptionalHeader，进行读取IMAGE_SECTION_HEADER。

OK，搞定。

代码如下：

```
//清空列表控件
m_ListCtrlSection.DeleteAllItems();
```

```
IMAGE_DOS_HEADER DosHeader = {0};
IMAGE_FILE_HEADER FileHeader = {0};
HANDLE  hFile = INVALID_HANDLE_VALUE;
DWORD  dwReadLen = 0;
WORD  NumOfSec = 0; //区段表个数
IMAGE_SECTION_HEADER *pSecHeader = NULL;
//打开文件
hFile =
CreateFile("C://windows//system32//cmd.exe",GENERIC_READ,FILE_SHARE_READ,NULL,OPEN_EXISTING,
if (hFile == INVALID_HANDLE_VALUE)
{
 MessageBox(_T("无法打开文件，分析失败！"));
 return;
}

//读取数据
if (!ReadFile(hFile,&DosHeader,sizeof(DosHeader),&dwReadLen,NULL))
{
 MessageBox(_T("无法读取文件，分析失败！"));
 return;
}
SetFilePointer(hFile,DosHeader.e_lfanew + sizeof(IMAGE_NT_SIGNATURE),NULL,FILE_BEGIN);
ReadFile(hFile,&FileHeader,sizeof(FileHeader),&dwReadLen,NULL);
//得出区段个数
NumOfSec = FileHeader.NumberOfSections;
//分配区段内存
DWORD SectionSize = NumOfSec * IMAGE_SIZEOF_SECTION_HEADER;
char *pSecBuff = new char[SectionSize + 1];
//移动指针到区段表
SetFilePointer(hFile,FileHeader.SizeOfOptionalHeader,NULL,FILE_CURRENT);
ReadFile(hFile,pSecBuff,SectionSize,&dwReadLen,NULL);
```

```
//读取完毕，关闭句柄
CloseHandle(hFile);
```

```
int iIndexItem = 0;
TCHAR tzBuff[10] = {0};
char szSecName[IMAGE_SIZEOF_SHORT_NAME] = {0};
//内存数据指针转换
pSecHeader = (PIMAGE_SECTION_HEADER)pSecBuff;
for (int i = 0; i < NumOfSec; i++)
{
 memcpy(szSecName,pSecHeader->Name,8);
 iIndexItem = m_ListCtrlSection.InsertItem(i,(LPCTSTR)szSecName);

 //显示VirtualAddress和VirtualSize
 wsprintf(tzBuff,_T("%08X"),pSecHeader->VirtualAddress);
 m_ListCtrlSection.SetItemText(iIndexItem,1,tzBuff);
 wsprintf(tzBuff,_T("%08X"),pSecHeader->Misc.VirtualSize);
 m_ListCtrlSection.SetItemText(iIndexItem,2,tzBuff);

 //显示RawAddress和RawData
 wsprintf(tzBuff,_T("%08X"),pSecHeader->PointerToRawData);
 m_ListCtrlSection.SetItemText(iIndexItem,3,tzBuff);
 wsprintf(tzBuff,_T("%08X"),pSecHeader->SizeOfRawData);
 m_ListCtrlSection.SetItemText(iIndexItem,4,tzBuff);

 wsprintf(tzBuff,_T("%08X"),pSecHeader->Characteristics);
 m_ListCtrlSection.SetItemText(iIndexItem,5,tzBuff);

 //指针后移
 pSecHeader++;
}
//释放内存
if (pSecBuff)
{
 delete []pSecBuff;
}
```