

软件安全—恶意代码机理与防护

C3 PE文件格式

彭国军 教授

武汉大学国家网络安全学院

guojpeng@whu.edu.cn

本讲的内容提纲

3.1 PE文件及其表现形式

3.2 PE文件格式与恶意软件的关系

3.3 PE文件格式总体结构

3.4 代码节与数据节

3.5 引入函数节：PE文件的引入函数机制

3.6 引出函数节：DLL文件的函数引出机制

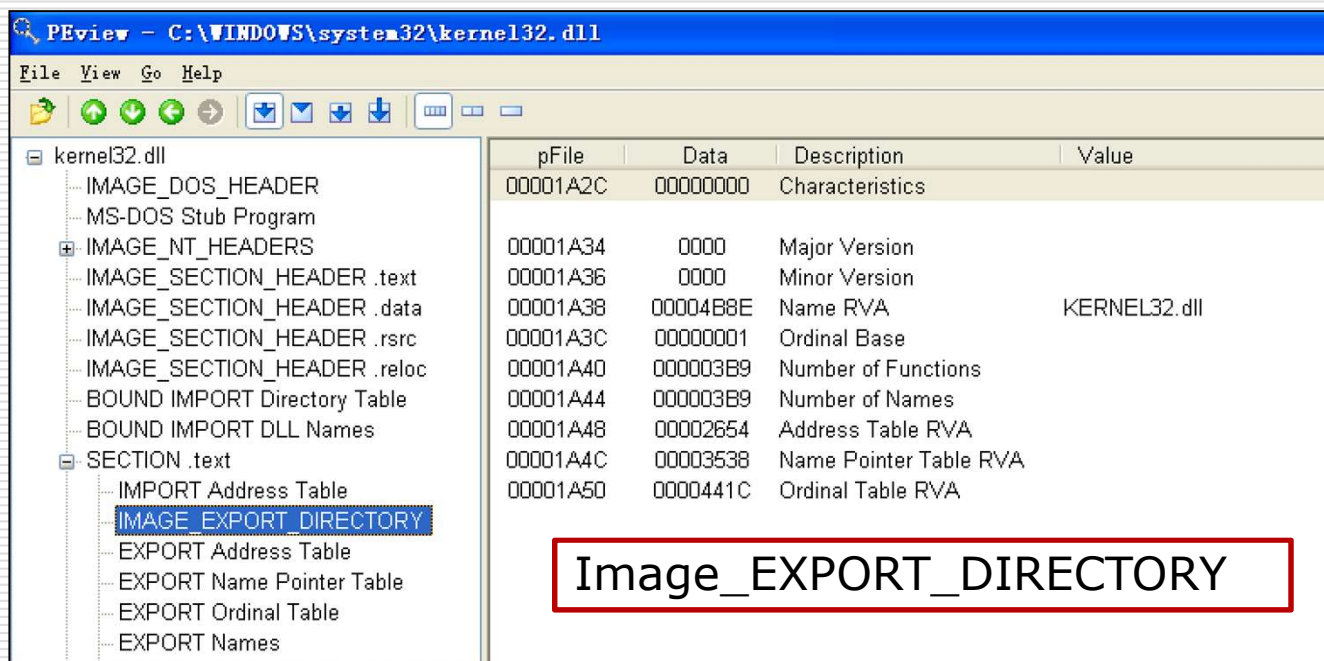
3.7 资源节：文件资源索引、定位与修改

3.8 重定位节：镜像地址改变后的地址自动修正

3.6 引出函数节

- 引出函数节一般名为**.edata**，这是本文件向其他程序提供调用函数的列表、函数所在的地址及具体代码实现。
 - 关键结构：引出目录表（导出表、输出表）
-

Kernel32.dll的引出函数节



PEView - C:\WINDOWS\system32\kernel32.dll

File View Go Help

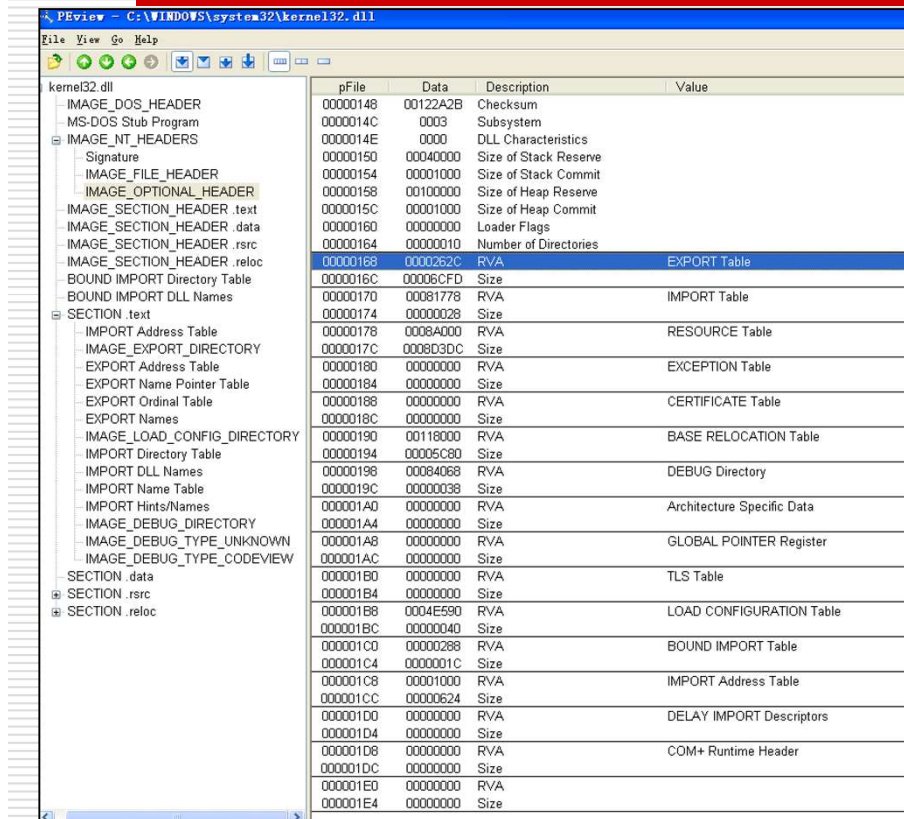
kernel32.dll

- ... IMAGE_DOS_HEADER
- ... MS-DOS Stub Program
- + IMAGE_NT_HEADERS
 - ... IMAGE_SECTION_HEADER .text
 - ... IMAGE_SECTION_HEADER .data
 - ... IMAGE_SECTION_HEADER .rsrc
 - ... IMAGE_SECTION_HEADER .reloc
 - ... BOUND_IMPORT Directory Table
 - ... BOUND_IMPORT DLL Names
 - + SECTION .text
 - ... IMPORT Address Table
 - IMAGE_EXPORT_DIRECTORY**
 - ... EXPORT Address Table
 - ... EXPORT Name Pointer Table
 - ... EXPORT Ordinal Table
 - ... EXPORT Names

pFile	Data	Description	Value
00001A2C	00000000	Characteristics	
00001A34	0000	Major Version	
00001A36	0000	Minor Version	
00001A38	00004B8E	Name RVA	KERNEL32.dll
00001A3C	00000001	Ordinal Base	
00001A40	000003B9	Number of Functions	
00001A44	000003B9	Number of Names	
00001A48	00002654	Address Table RVA	
00001A4C	00003538	Name Pointer Table RVA	
00001A50	0000441C	Ordinal Table RVA	

Image_EXPORT_DIRECTORY

如何定位Export Directory Table —引出目录表？



	pFile	Data	Description	Value
kernel32.dll				
- IMAGE_DOS_HEADER	00000148	00122A2B	Checksum	
- MS-DOS Stub Program	0000014C	0003	Subsystem	
IMAGE_NT_HEADERS	0000014E	0000	DLL Characteristics	
- Signature	00000150	00040000	Size of Stack Reserve	
- IMAGE_FILE_HEADER	00000154	00001000	Size of Stack Commit	
- IMAGE_OPTIONAL_HEADER	00000158	00100000	Size of Heap Reserve	
- IMAGE_SECTION_HEADER .text	0000015C	00001000	Size of Heap Commit	
- IMAGE_SECTION_HEADER .data	00000160	00000000	Loader Flags	
- IMAGE_SECTION_HEADER .rsrc	00000164	00000010	Number of Directories	
- IMAGE_SECTION_HEADER .reloc	00000168	0000262C	RVA	EXPORT Table
- BOUND_IMPORT Directory Table	0000016C	00006CFD	Size	
- BOUND_IMPORT DLL Names	00000170	00081778	RVA	IMPORT Table
SECTION .text	00000174	00000028	Size	
- IMPORT Address Table	00000178	0008A000	RVA	RESOURCE Table
- IMAGE_EXPORT_DIRECTORY	0000017C	0008D3DC	Size	
- EXPORT Address Table	00000180	00000000	RVA	EXCEPTION Table
- EXPORT Name Pointer Table	00000184	00000000	Size	
- EXPORT Ordinal Table	00000188	00000000	RVA	CERTIFICATE Table
- EXPORT Names	0000018C	00000000	Size	
- IMAGE_LOAD_CONFIG_DIRECTORY	00000190	00118000	RVA	BASE RELOCATION Table
- IMPORT Directory Table	00000194	00005C80	Size	
- IMPORT DLL Names	00000198	00084068	RVA	DEBUG Directory
- IMPORT Name Table	0000019C	00000038	Size	
- IMPORT Hints/Names	000001A0	00000000	RVA	Architecture Specific Data
- IMAGE_DEBUG_DIRECTORY	000001A4	00000000	Size	
- IMAGE_DEBUG_TYPE_UNKNOWN	000001A8	00000000	RVA	GLOBAL POINTER Register
- IMAGE_DEBUG_TYPE_CODEVIEW	000001AC	00000000	Size	
SECTION .data	000001B0	00000000	RVA	TLS Table
SECTION .rsrc	000001B4	00000000	Size	
SECTION .reloc	000001B8	0004E590	RVA	LOAD CONFIGURATION Table
	000001BC	00000040	Size	
	000001C0	00000288	RVA	BOUND_IMPORT Table
	000001C4	0000001C	Size	
	000001C8	00001000	RVA	IMPORT Address Table
	000001CC	00000624	Size	
	000001D0	00000000	RVA	DELAY_IMPORT Descriptors
	000001D4	00000000	Size	
	000001D8	00000000	RVA	COM+ Runtime Header
	000001DC	00000000	Size	
	000001E0	00000000	RVA	
	000001E4	00000000	Size	

□ DataDirectory
第一项

3.6.1 引出目录表

—Image_EXPORT_DIRECTORY

```
01.  typedef struct _IMAGE_EXPORT_DIRECTORY {
02.      DWORD   Characteristics;
03.      DWORD   TimeDateStamp;
04.      WORD    MajorVersion;
05.      WORD    MinorVersion;
06.      DWORD   Name;
07.      DWORD   Base;
08.      DWORD   NumberOfFunctions;
09.      DWORD   NumberOfNames;
10.      DWORD   AddressOfFunctions;    // RVA from base of image
11.      DWORD   AddressOfNames;        // RVA from base of image
12.      DWORD   AddressOfNameOrdinals; // RVA from base of image
13.  } IMAGE_EXPORT_DIRECTORY, *PIMAGE_EXPORT_DIRECTORY;
```

引出目录表结构解析

1	(00H)	Characteristics	4	一般为 0
2	(04H)	TimeDateStamp	4	文件生成时间
3	(08H)	MajorVersion	2	主版本号
4	(0AH)	MinorVersion	2	次版本号
5	(0CH)	<i>Name</i> 	4	指向 DLL 的名字
6	(10H)	<i>Base</i>	4	开始的序列号
7	(14H)	<i>NumberOfFunctions</i>	4	<i>AddressOfFunctions</i> 数组的项数
8	(18H)	<i>NumberOfNames</i>	4	<i>AddressOfNames</i> 数组的项数
9	(1CH)	<i>AddressOfFunctions</i>	4	指向“函数地址”数组—导出地址表
10	(20H)	<i>AddressOfNames</i>	4	指向“函数名所在地址”数组—函数名地址表
11	(24H)	<i>AddressOfNameOrdinals</i>	4	指向“函数索引序列号”数组—函数序号表

3.6.2 导出地址表

— EXPORT ADDRESS Table

```
01. // 导出表信息
02. typedef struct _IMAGE_Export_Address_Table_
03. {
04.     union {
05.         DWORD dwExportRVA;
06.         DWORD dwForwarderRVA;
07.     };
08. } IMAGE_Export_Address_Table, *pIMAGE_Export_Address_Table;
```

PEview - C:\WINDOWS\system32\kernel32.dll

File View Go Help

kernel32.dll

	RVA	Data	Description	Value
IMAGE_DOS_HEADER	00002654	0000A6D4	Function RVA	0001 ActivateActCtx
MS-DOS Stub Program	00002658	00035505	Function RVA	0002 AddAtomA
IMAGE_NT_HEADERS	0000265C	000326D9	Function RVA	0003 AddAtomW
Signature	00002660	00071CDF	Function RVA	0004 AddConsoleAliasA
IMAGE_FILE_HEADER	00002664	00071CA1	Function RVA	0005 AddConsoleAliasW
IMAGE_OPTIONAL_HEADER	00002668	00059382	Function RVA	0006 AddLocalAlternateComputerNameA
IMAGE_SECTION_HEADER .text	0000266C	00059266	Function RVA	0007 AddLocalAlternateComputerNameW
IMAGE_SECTION_HEADER .data	00002670	0002BEF9	Function RVA	0008 AddRefActCtx
IMAGE_SECTION_HEADER .rsrc	00002674	00008FF5	Forwarded Name RVA	0009 AddVectoredExceptionHandler -> NTDLL.RtlAddVectoredExceptionHandler
IMAGE_SECTION_HEADER .reloc	00002678	00072331	Function RVA	000A AllocConsole
BOUND_IMPORT Directory Table	0000267C	0005F61A	Function RVA	000B AllocateUserPhysicalPages
BOUND_IMPORT DLL Names	00002680	00035967	Function RVA	000C AreFileApisANSI
SECTION .text	00002684	0002E442	Function RVA	000D AssignProcessToJobObject
IMPORT Address Table	00002688	00072519	Function RVA	000E AttachConsole
IMAGE_EXPORT_DIRECTORY	0000268C	000571CA	Function RVA	000F BackupRead
EXPORT Address Table	00002690	000562B0	Function RVA	0010 BackupSeek
EXPORT Name Pointer Table	00002694	00057825	Function RVA	0011 BackupWrite
EXPORT Ordinal Table	00002698	00016867	Function RVA	0012 BaseCheckAppcompatCache
EXPORT Names	0000269C	0006CDE6	Function RVA	0013 BaseCleanupAppcompatCache

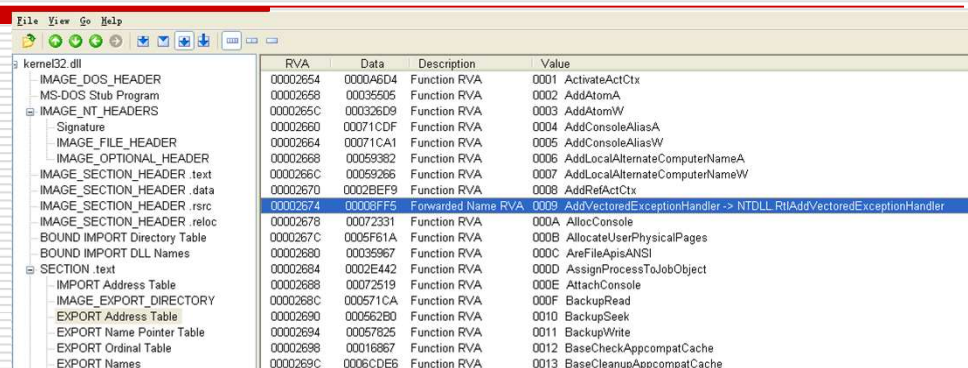
Dword的两种可能含义

□ dwExportRVA

- 指向导出地址

□ dwForwarderRVA

- 指向另外一个DLL中的某个API函数名。
- 举例: [Kernel32.AddVectoredExceptionHandler](#)
 - →NTDLL.RtlAddVectoredExceptionHandler



	RVA	Data	Description	Value
00002654	000046D4	Function RVA	0001 ActivateActCtx	
00002658	00035505	Function RVA	0002 AddAtomA	
0000265C	000326D9	Function RVA	0003 AddAtomW	
00002660	00071CDF	Function RVA	0004 AddConsoleAliasA	
00002664	00071CA1	Function RVA	0005 AddConsoleAliasW	
00002668	00059362	Function RVA	0006 AddLocalAlternateComputerNameA	
0000266C	00059266	Function RVA	0007 AddLocalAlternateComputerNameW	
00002670	00028EF9	Function RVA	0008 AddRefActCtx	
00002674	000358F5	Forwarded Name RVA	0009 AddVectoredExceptionHandler -> NTDLL.RtlAddVectoredExceptionHandler	
00002678	00072331	Function RVA	000A AllocConsole	
0000267C	0005F61A	Function RVA	000B AllocateUserPhysicalPages	
00002680	00035967	Function RVA	000C AreFileApisANSI	
00002684	0002E442	Function RVA	000D AssignProcessToJobObject	
00002688	00072519	Function RVA	000E AttachConsole	
0000268C	000571CA	Function RVA	000F BackupRead	
00002690	000562B0	Function RVA	0010 BackupSeek	
00002694	00057925	Function RVA	0011 BackupWrite	
00002698	00016867	Function RVA	0012 BaseCheckAppcompatCache	
0000269C	0006CDE6	Function RVA	0013 BaseCleanupAppcompatCache	

3.6.3 导出名字表

-EXPORT Name Table

```
01. typedef struct _IMAGE_Export_Name_Pointer_Table_ {  
02.     DWORD dwPointer;  
03. }IMAGE_Export_Name_Pointer_Table,*pIMAGE_Export_Name_Pointer_Table;
```

PEview - C:\WINDOWS\system32\kernel32.dll

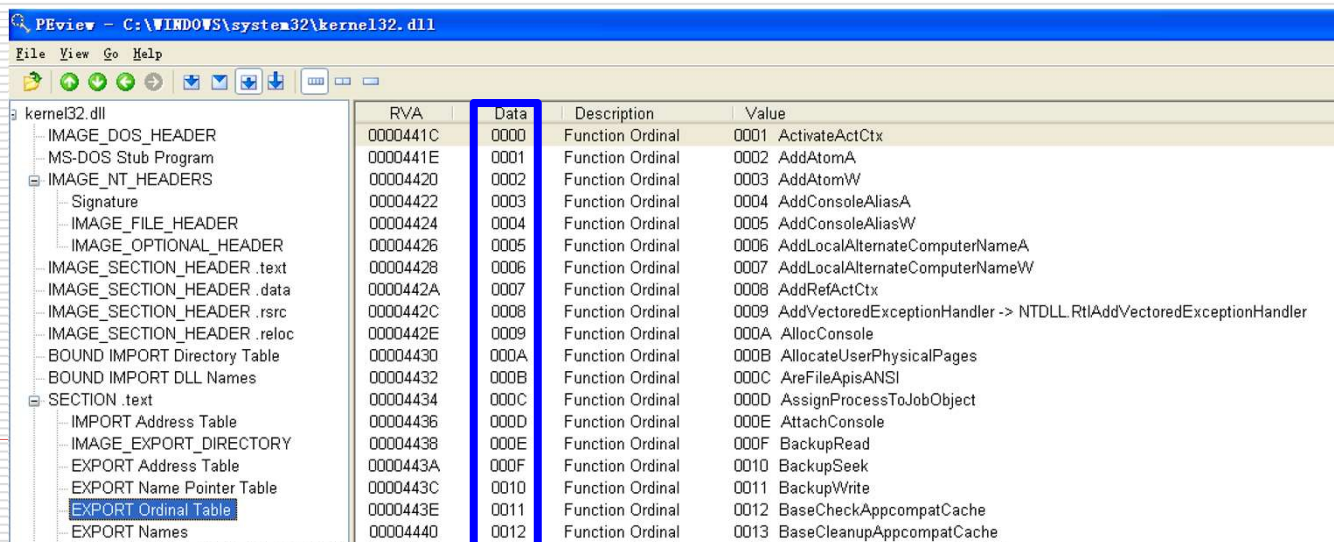
File View Go Help

	RVA	Data	Description	Value
kernel32.dll				
- IMAGE_DOS_HEADER				
- MS-DOS Stub Program	0000353C	00004BAA	Function Name RVA	0001 ActivateActCtx
- IMAGE_NT_HEADERS	00003540	00004BB3	Function Name RVA	0002 AddAtomA
- Signature	00003544	00004BBC	Function Name RVA	0003 AddAtomW
- IMAGE_FILE_HEADER	00003548	00004BCD	Function Name RVA	0004 AddConsoleAliasA
- IMAGE_OPTIONAL_HEADER	0000354C	00004BDE	Function Name RVA	0005 AddConsoleAliasW
- IMAGE_SECTION_HEADER .text	00003550	00004BFD	Function Name RVA	0006 AddLocalAlternateComputerNameA
- IMAGE_SECTION_HEADER .data	00003554	00004C1C	Function Name RVA	0007 AddLocalAlternateComputerNameW
- IMAGE_SECTION_HEADER .rsrc	00003558	00004C29	Function Name RVA	0008 AddRefActCtx
- IMAGE_SECTION_HEADER .reloc	0000355C	00004C45	Function Name RVA	0009 AddVectoredExceptionHandler -> NTDLL.RtlAddVectoredExceptionHandler
- BOUND_IMPORT Directory Table	00003560	00004C52	Function Name RVA	000A AllocConsole
- BOUND_IMPORT DLL Names	00003564	00004C6C	Function Name RVA	000B AllocateUserPhysicalPages
- SECTION .text	00003568	00004C7C	Function Name RVA	000C AreFileApisANSI
- IMPORT Address Table	0000356C	00004C95	Function Name RVA	000D AssignProcessToJobObject
- IMAGE_EXPORT_DIRECTORY	00003570	00004CA3	Function Name RVA	000E AttachConsole
- EXPORT Address Table	00003574	00004CAE	Function Name RVA	000F BackupRead
- EXPORT Name Pointer Table	00003578	00004CB9	Function Name RVA	0010 BackupSeek
- EXPORT Ordinal Table	0000357C	00004CC5	Function Name RVA	0011 BackupWrite
- EXPORT Names	00003580	00004CDD	Function Name RVA	0012 BaseCheckAppcompatCache
				0013 BaseCleanupAppcompatCache

3.6.4 导出序号表 —EXPORT Ordinal Table

□ 该表保存的是各导出函数的函数地址在导出地址表的序

```
01. typedef struct _IMAGE_Export_Ordinal_Table_ {  
02.     WORD dwOrdinal;  
03. }IMAGE_Export_Ordinal_Table,*pIMAGE_Export_Ordinal_Table;
```



The screenshot shows the PEView application window with the file C:\WINDOWS\system32\kernel32.dll loaded. The left pane displays the file's structure, with the 'EXPORT Ordinal Table' highlighted. The right pane shows a table of export ordinals, with the 'Data' column highlighted. The table lists function ordinals from 0000 to 0013, each corresponding to a specific function name.

RVA	Data	Description	Value
0000441C	0000	Function Ordinal	0001 ActivateActCtx
0000441E	0001	Function Ordinal	0002 AddAtomA
00004420	0002	Function Ordinal	0003 AddAtomW
00004422	0003	Function Ordinal	0004 AddConsoleAliasA
00004424	0004	Function Ordinal	0005 AddConsoleAliasW
00004426	0005	Function Ordinal	0006 AddLocalAlternateComputerNameA
00004428	0006	Function Ordinal	0007 AddLocalAlternateComputerNameW
0000442A	0007	Function Ordinal	0008 AddRefActCtx
0000442C	0008	Function Ordinal	0009 AddVectoredExceptionHandler -> NTDLL.RtlAddVectoredExceptionHandler
0000442E	0009	Function Ordinal	000A AllocConsole
00004430	000A	Function Ordinal	000B AllocateUserPhysicalPages
00004432	000B	Function Ordinal	000C AreFileApisANSI
00004434	000C	Function Ordinal	000D AssignProcessToJobObject
00004436	000D	Function Ordinal	000E AttachConsole
00004438	000E	Function Ordinal	000F BackupRead
0000443A	000F	Function Ordinal	0010 BackupSeek
0000443C	0010	Function Ordinal	0011 BackupWrite
0000443E	0011	Function Ordinal	0012 BaseCheckAppcompatCache
00004440	0012	Function Ordinal	0013 BaseCleanupAppcompatCache

为何需要导出序号表？

□ 导出函数名字和导出地址表中的地址不是一对一关系。

□ 为什么？

- 一个函数实现可能有多个名字；
 - 某些函数没有名字，仅通过序号导出。
-

shlwapi.dll文件 (13A/35A)

PEview - C:\WINDOWS\system32\shlwapi.dll

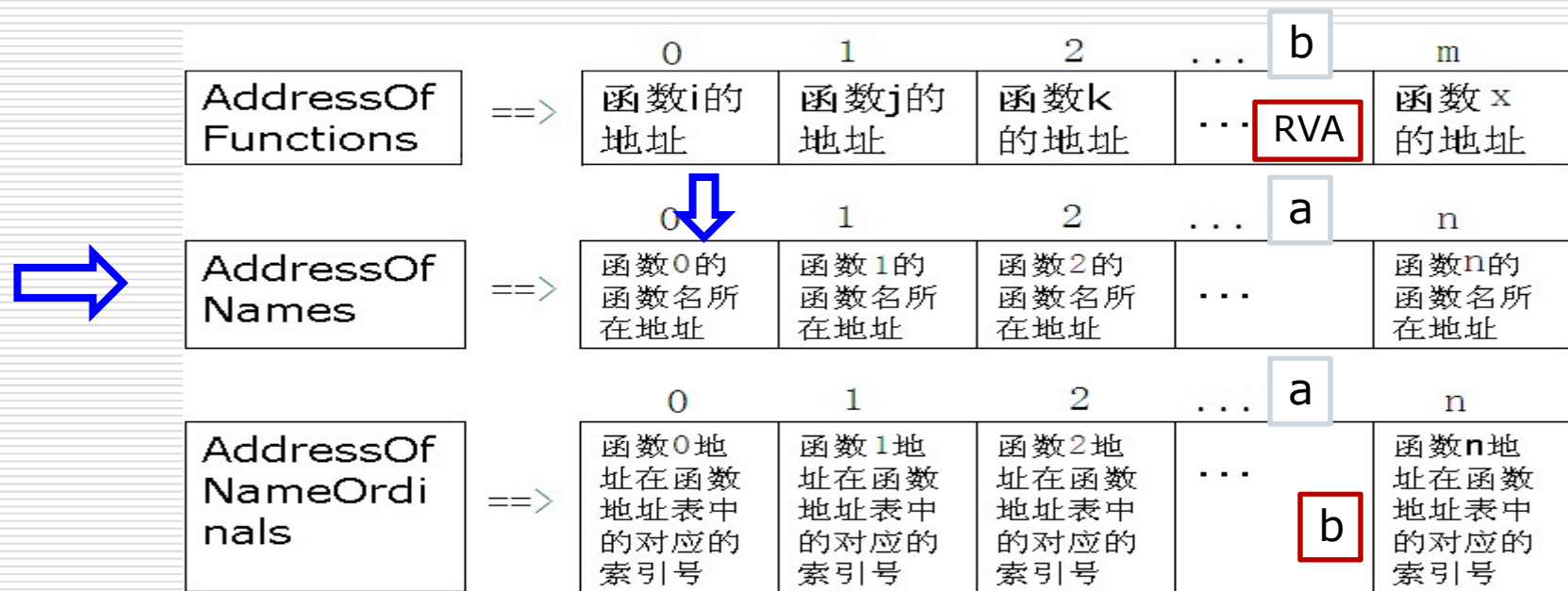
	RVA	Data	Description	Value
shlwapi.dll				
IMAGE_DOS_HEADER	00001844	00021DCE	Function RVA	0001
MS-DOS Stub Program	00001848	0000B802	Function RVA	0002
IMAGE_NT_HEADERS	0000184C	0004688E	Function RVA	0003
IMAGE_SECTION_HEADER .text	00001850	000187DF	Function RVA	0004
IMAGE_SECTION_HEADER .data	00001854	00048C41	Function RVA	0005
IMAGE_SECTION_HEADER .rsrc	00001858	0001531A	Function RVA	0006
IMAGE_SECTION_HEADER .reloc	0000185C	0000B5F1	Function RVA	0007 SHAllocShared
BOUND_IMPORT Directory Table	00001860	0001C287	Function RVA	0008 SHLockShared
BOUND_IMPORT DLL Names	00001864	0000B53E	Function RVA	0009 SHUnlockShared
SECTION .text	00001868	0000B559	Function RVA	000A SHFreeShared
IMPORT Address Table	0000186C	0000B4B0	Function RVA	000B
IMAGE_EXPORT_DIRECTORY	00001870	000184CA	Function RVA	000C
EXPORT Address Table	00001874	00022C34	Function RVA	000D
EXPORT Name Pointer Table	00001878	0005DDF5	Function RVA	000E GetAcceptLanguagesA
EXPORT Ordinal Table	0000187C	0002CAF2	Function RVA	000F GetAcceptLanguagesW
EXPORT Names	00001880	000173A6	Function RVA	0010 SHCreateThread
IMAGE_LOAD_CONFIG_DIRECTORY	00001884	0001BD3B	Function RVA	0011
DELAY_IMPORT Descriptors	00001888	0001A31B	Function RVA	0012
DELAY_IMPORT DLL Names	0000188C	0000B87E	Function RVA	0013
DELAY_IMPORT Name Table	00001890	0000B800	Function RVA	0014
DELAY_IMPORT Hints/Names	00001894	0000B7DF	Function RVA	0015
IMPORT Directory Table	00001898	0001A2E1	Function RVA	0016
IMPORT DLL Names	0000189C	0000B562	Function RVA	0017
IMPORT Name Table	000018A0	00004D01	Function RVA	0018
IMPORT Hints/Names	000018A4	000227A5	Function RVA	0019
IMAGE_DEBUG_DIRECTORY	000018A8	000454C4	Function RVA	001A
IMAGE_DEBUG_TYPE_UNKNOWN	000018AC	00045605	Function RVA	001B
IMAGE_DEBUG_TYPE_CODEVIEW	000018B0	00022955	Function RVA	001C
SECTION .data	000018B4	000158A5	Function RVA	001D IsCharSpaceW
SECTION .rsrc	000018B8	0004548D	Function RVA	001E
SECTION .reloc	000018BC	00045325	Function RVA	001F
	000018C0	00045465	Function RVA	0020
	000018C4	00022686	Function RVA	0021

PEview - C:\WINDOWS\system32\shlwapi.dll

	RVA	Data	Description	Value
shlwapi.dll				
IMAGE_DOS_HEADER	0000181C	00000000	Characteristics	
MS-DOS Stub Program				
IMAGE_NT_HEADERS	00001824	0000	Major Version	
IMAGE_SECTION_HEADER .text	00001826	0000	Minor Version	
IMAGE_SECTION_HEADER .data	00001828	00002D08	Name RVA	SHLWAPI.dll
IMAGE_SECTION_HEADER .rsrc	0000182C	00000001	Ordinal Base	
IMAGE_SECTION_HEADER .reloc	00001830	0000035A	Number of Functions	
BOUND_IMPORT Directory Table	00001834	0000013A	Number of Names	
BOUND_IMPORT DLL Names				
SECTION .text				
IMPORT Address Table				
IMAGE_EXPORT_DIRECTORY				
EXPORT Address Table				
EXPORT Name Pointer Table				
EXPORT Ordinal Table				
EXPORT Names				
IMAGE_LOAD_CONFIG_DIRECTORY				
DELAY_IMPORT Descriptors				
DELAY_IMPORT DLL Names				
DELAY_IMPORT Name Table				
DELAY_IMPORT Hints/Names				
IMPORT Directory Table				
IMPORT DLL Names				
IMPORT Name Table				
IMPORT Hints/Names				
IMAGE_DEBUG_DIRECTORY				
IMAGE_DEBUG_TYPE_UNKNOWN				
IMAGE_DEBUG_TYPE_CODEVIEW				
SECTION .data				
SECTION .rsrc				
SECTION .reloc				

3.6.5通过函数名 定位函数导出地址

7	(14H)	NumberOfFunctions	4	AddressOfFunctions 数组的项数
8	(18H)	NumberOfNames	4	AddressOfNames 数组的项数
9	(1CH)	AddressOfFunctions	4	指向函数地址数组
10	(20H)	AddressOfNames	4	指向“函数名所在地址”数组
11	(24H)	AddressOfNameOrdinals	4	指向“函数索引序列号”数组



m=NumberOfFunctions

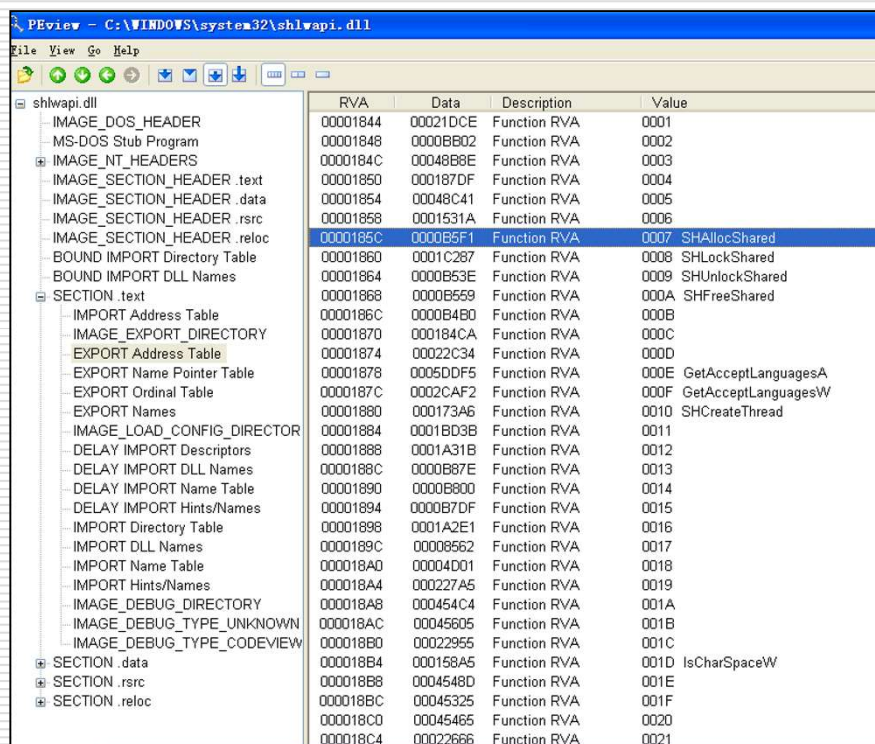
n=NumberOfNames

9	(1CH)	AddressOfFunctions	4	指向函数地址数组
10	(20H)	AddressOfNames	4	函数名字的指针的地址
11	(24H)	AddressOfNameOrdinals	4	指向输入序列号数组

定位shlwapi.dll中HashData函数地址

1. 首先从AddressOfNames指向的指针数组中找到“HashData”字符串，并记下该数组序号a=_____
2. 然后从AddressOfNameOrdinals指向的数组中，定位第a项成员，得到一个序号b=_____
3. 从AddressOfFunction指向的数组中定位第b项，获得DLL的RVA函数地址=_____
4. 该函数在内存中的真实地址为：_____。
(ImageBase: 77F40000)

定位shlwapi.dll中HashData函数地址



RVA	Data	Description	Value
00001844	00021DCE	Function RVA	0001
00001848	0000BB02	Function RVA	0002
0000184C	00048B8E	Function RVA	0003
00001850	000187DF	Function RVA	0004
00001854	00048C41	Function RVA	0005
00001858	0001531A	Function RVA	0006
0000185C	0000B5F1	Function RVA	0007 SHAllocShared
00001860	0001C287	Function RVA	0008 SHLockShared
00001864	0000B53E	Function RVA	0009 SHUnlockShared
00001868	0000B559	Function RVA	000A SHFreeShared
0000186C	0000B4B0	Function RVA	000B
00001870	000184CA	Function RVA	000C
00001874	00022C34	Function RVA	000D
00001878	0005DDF5	Function RVA	000E GetAcceptLanguagesA
0000187C	0002CAF2	Function RVA	000F GetAcceptLanguagesW
00001880	000173A6	Function RVA	0010 SHCreateThread
00001884	0001BD3B	Function RVA	0011
00001888	0001A31B	Function RVA	0012
0000188C	0000B87E	Function RVA	0013
00001890	0000B800	Function RVA	0014
00001894	0000B7DF	Function RVA	0015
00001898	0001A2E1	Function RVA	0016
0000189C	00008562	Function RVA	0017
000018A0	00004D01	Function RVA	0018
000018A4	000227A5	Function RVA	0019
000018A8	000454C4	Function RVA	001A
000018AC	00045605	Function RVA	001B
000018B0	00022955	Function RVA	001C
000018B4	000158A5	Function RVA	001D IsCharSpaceW
000018B8	0004548D	Function RVA	001E
000018BC	00045325	Function RVA	001F
000018C0	00045465	Function RVA	0020
000018C4	00022666	Function RVA	0021

- ☐ a=
- ☐ b=
- ☐ Func RVA=
- ☐ Func VA=

通过函数序号导出（引入序号）：base+b

练习

- 在介绍函数引入机制时，提到也可以通过函数序号导入。
 - MessageBoxA的函数序号是多少？
 - Base+ordinal
 - 如何验证？
-