




PE Format

: 2005. 3. 18
: 2005. 4. 05
(A.K.A. Anesra)

- 
1. PE 가?
 2. PE 가?
 3. Overview of PE File Format
 4. Detecting a Valid PE File
 5. File Header
 6. Optional Header
 7. Section Table
 8. Import Table
 9. Export Table

1. PE 가?

1. 가 .
2. 가
3. PE .
4. Exploit Code .

2. PE 가?

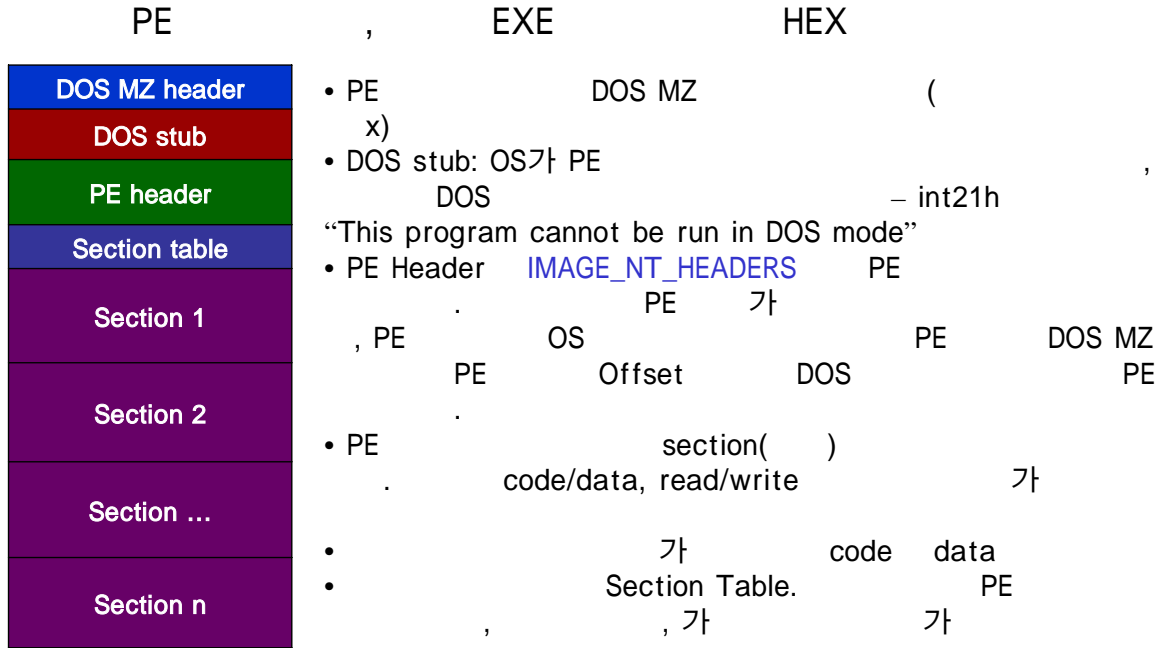
1. Overview Of PE file format

- PE (Portable Executable – 가)
- Win32
- PE UNIX Coff (Common object file format)
- “Portable Executable” Win32 가

2. PE

가?

1. Overview Of PE file format

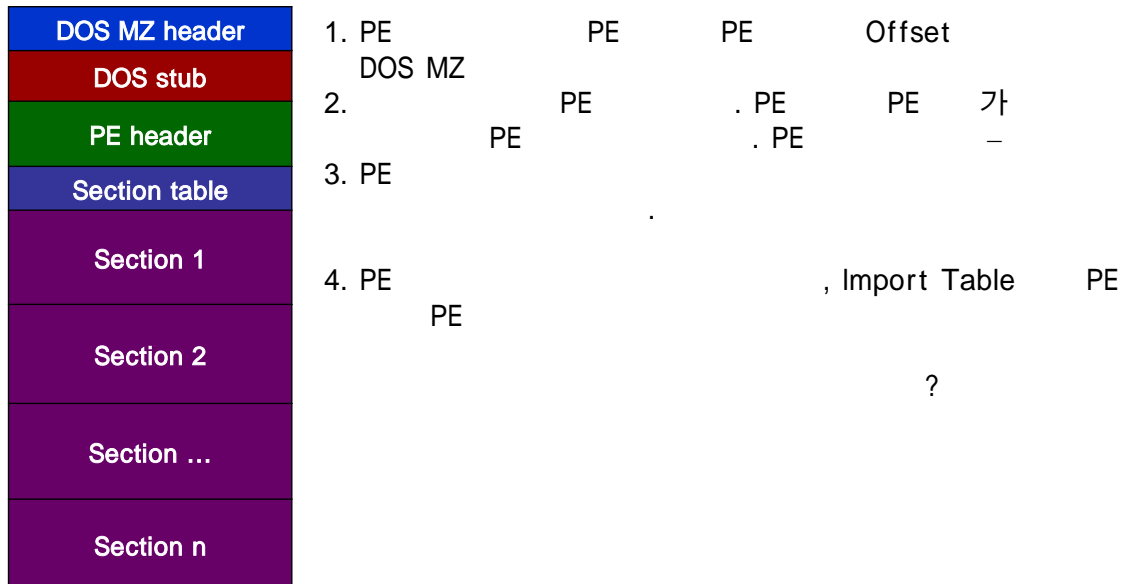


2. PE

가?

1. Overview Of PE file format

PE



2. PE

가?

1. Overview Of PE file format

PE	, EXE	HEX
DOS MZ header	4B 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00	: MZ?..... ..
DOS stub	00 00 00 00 00 00 00 00 40 00 00 00 00 00 00	: ?.....@.....
PE header	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	:
Section table	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	:
Section 1	0E 1F 8A 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68	: ..?.???L?Th
Section 2	69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F	: is program canno
Section ...	74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20	: t be run in DOS
Section n	6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00	: mode....@.....
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	: ?K?H?H?H?H?
	79 54 2B F4 F1 48 25 F4 12 57 2F F4 CC 48 25 F4	: yT+?H?W/?H?
	98 57 36 F4 F9 48 25 F4 FA 48 24 F4 CB 48 25 F4	: ?H?H?H?H?H?
	12 57 2E F4 F9 48 25 F4 52 69 63 68 FA 48 25 F4	: .M.?H?ich?H?
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	:
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	:
	FE 7C E4 41 00 00 00 00 00 00 00 00 00 00 00	:
	08 01 06 00 00 F0 01 00 00 70 00 00 00 00 00	:
	40 11 00 00 00 10 00 00 00 10 00 00 00 40 00	: @.....@.....
	00 10 00 00 00 10 00 00 00 04 00 00 00 00 00	:
	04 00 00 00 00 00 00 00 00 70 02 00 00 10 00	:
	00 00 00 00 03 00 00 00 00 10 00 00 10 00 00	:
	00 00 10 00 00 10 00 00 00 00 00 00 10 00 00	:
	00 00 00 00 00 00 00 00 00 50 02 00 28 00 00	:
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	:
	00 00 00 00 00 00 00 00 60 02 00 E0 09 00 00	:

2. PE

가?

2. Detecting a valid PE File

PE Header (IMAGE_NT_HEADERS)

```

IMAGE_NT_HEADERS STRUCT
    Signature dd ?
    FileHeader IMAGE_FILE_HEADER <>
    OptionalHeader IMAGE_OPTIONAL_HEADER32 <>
IMAGE_NT_HEADERS ENDS
    
```

- Signature dd ? : 50h,45h,00h,00h dword, (=“PE/0/0”)-PE
- FileHeader : , machine(CPU) physical 가
- OptionalHeader : Logical 가

• IMAGE_NT_HEADERS Signature ‘PE/0/0’ PE

• 2: Where is PE Header?

• => DOS MZ가 PE Offset 가

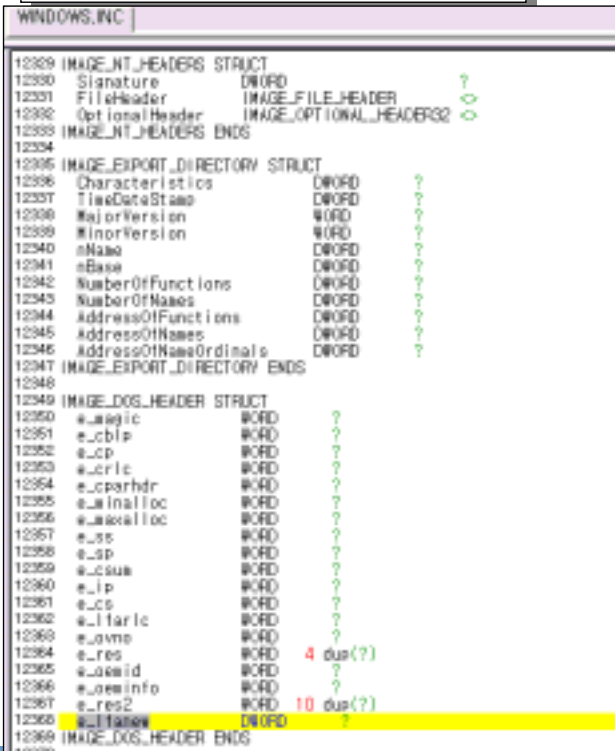
• DOS MZ IMAGE_DOS_HEADER (windows.inc)

• IMAGE_DOS_HEADER e_lfanew PE 가

• windows.inc !

2. PE 가?

2. Detecting a valid PE File

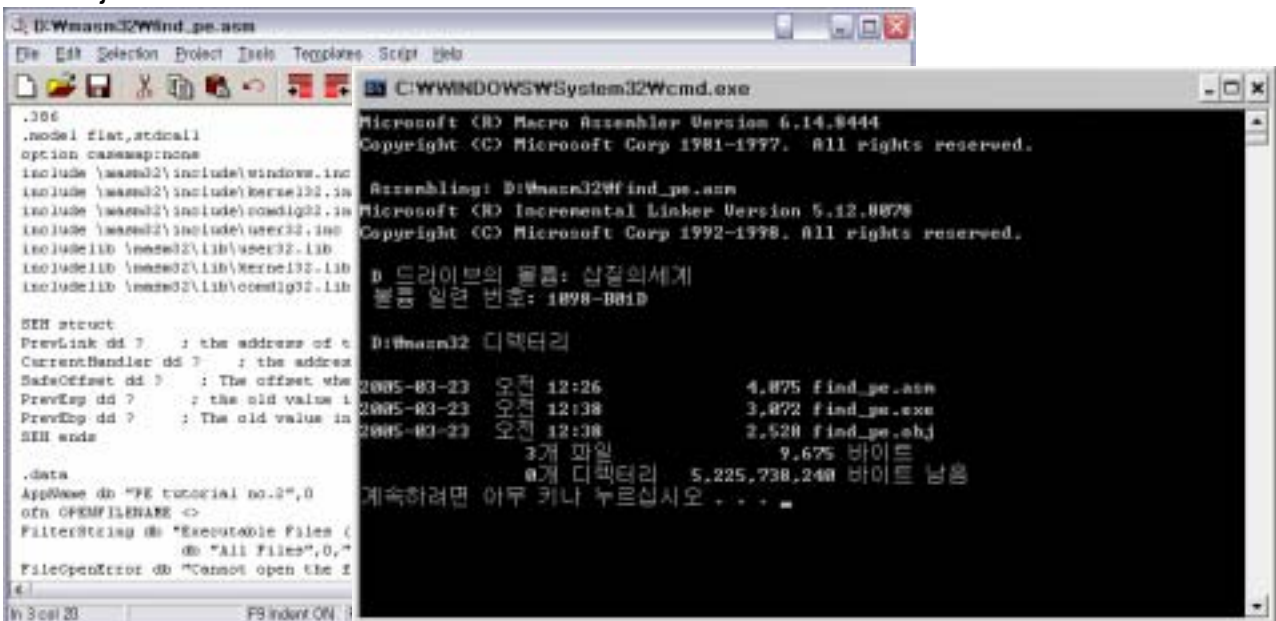


- e_lfanew
- <PE >
- IMAGE_DOS_HEADER (=DOS MZ)
- 1) 가
IMAGE_DOS_SIGNATURE(='MZ')
(e_magic)
 - 2) , PE
e_lfanew
 - 3) PE 가
IMAGE_NT_HEADER (='PE')

2. PE 가?

2. Detecting a valid PE File

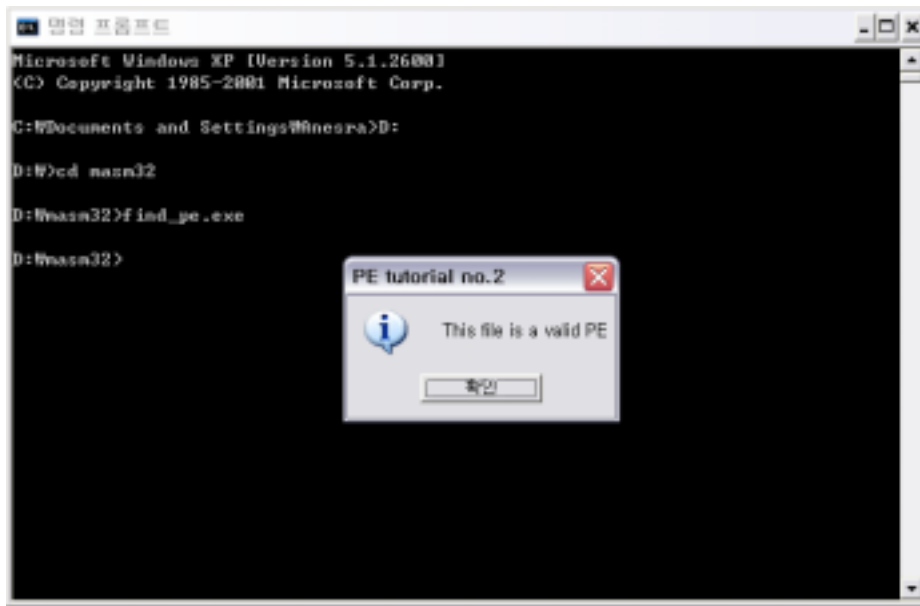
- masm32 Qeditor.exe PE asm project asm->obj, obj->exe



2. PE 가?

2. Detecting a valid PE File

- Find_PE.exe exe dll



2. PE 가?

3. File Header

```
PE Header
IMAGE_NT_HEADERS STRUCT
Signature dd ?
FileHeader IMAGE_FILE_HEADER <>
OptionalHeader IMAGE_OPTIONAL_HEADER32
<>
IMAGE_NT_HEADERS ENDS
```

```
FileHeader
IMAGE_FILE_HEADER STRUCT
Machine WORD ?
NumberOfSections WORD ?
TimeDateStamp dd ?
PointerToSymbolTable dd ?
NumberOfSymbols dd ?
SizeOfOptionalHeader WORD ?
Characteristics WORD ?
IMAGE_FILE_HEADER ENDS
```

- Machine : CPU , i386 14Ch
- NumberOfSection : 가/
- TimeDateStamp :
- PointerToSymbolTable :
- NumberOfSymbol :
- SizeOfOptionalHeader : OptionalHeade ,
- Characteristics : exe dll 가

2. PE

가?

4. Optional Header

- PE 가
- Logical Layout , 31
- PE 가 가 RVA .
- RVA (Relative Virtual Address: 가)-
- RVA - 가 , 가
- 가 400000h 가 401000h
- , RVA 1000h

2. PE

가?

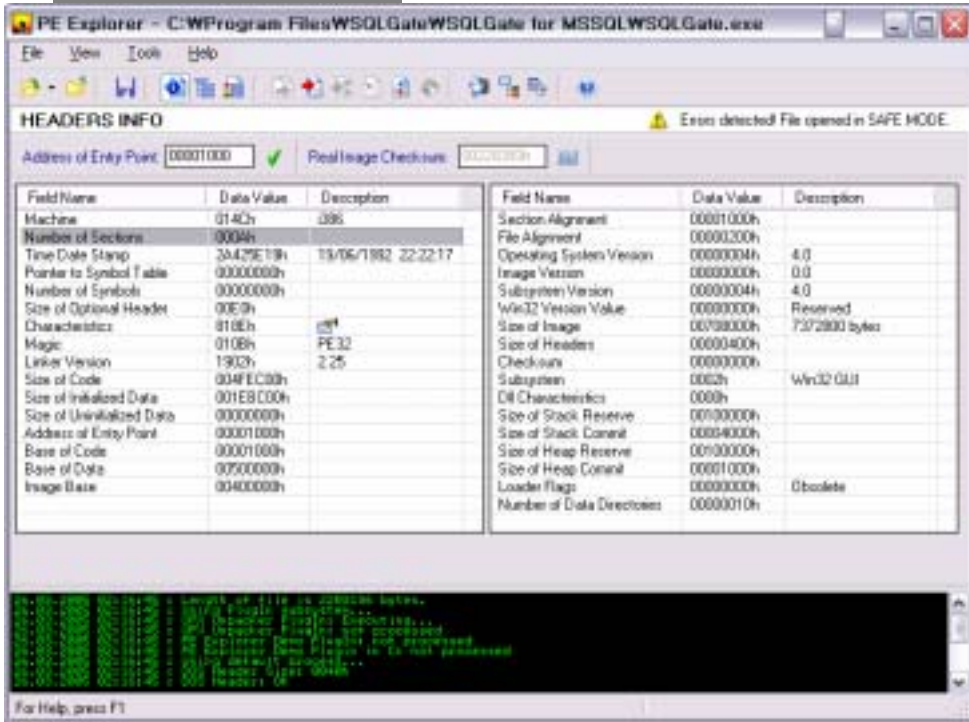
4. Optional Header

AddressOfEntryPoint	PE RVA
ImageBase	PE preferred.
SectionAlignment	granularity 4096(1000h) 4096 가 가10 402000h 401000h
FileAlignment	granularity.
MajorSubsystemVersion MinorSubsystemVersion	
SizeOfImage	PE
SizeOfHeaders	
Subsystem	
DataDirectory	IMAGE_DATA_DIRECTORY address table RVA import 가

2. PE

가?

4. Optional Header

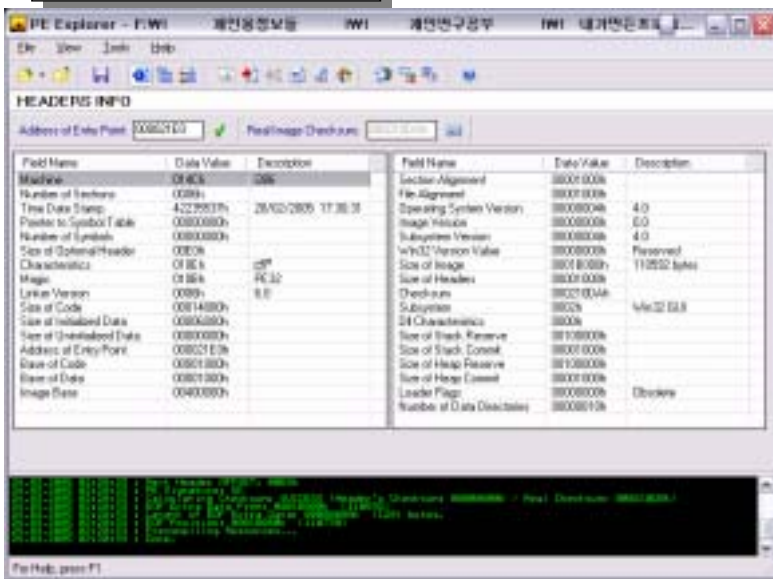


• PE Explorer

2. PE

가?

4. Optional Header

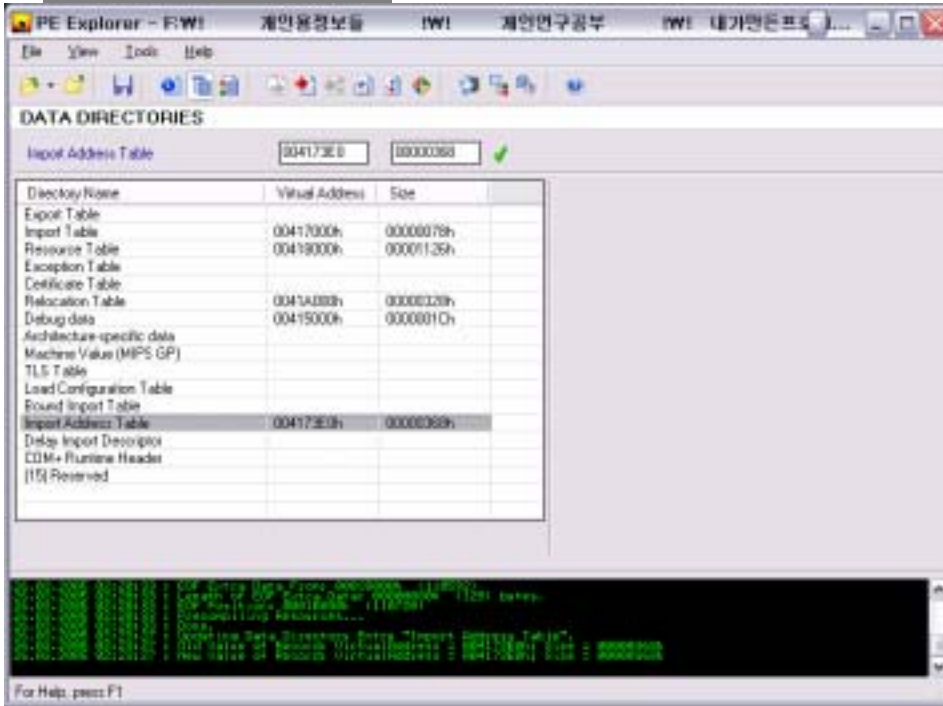


• PE Explorer

• PE Explorer

•

4. Optional Header



5. Section Table

DOS MZ header
DOS stub
PE header
Section table
Section 1
Section 2
Section ...
Section n

- Section Table PE Header
- file header (IMAGE_FILE_HEADER)
- NumberOfSections
- IMAGE_SECTION_HEADER

```

IMAGE_SIZEOF_SHORT_NAME equ 8
IMAGE_SECTION_HEADER STRUCT
    Name1 db IMAGE_SIZEOF_SHORT_NAME dup(?)
    union Misc
        PhysicalAddress dd ?
        VirtualSize dd ?
    ends
    VirtualAddress dd ?
    SizeOfRawData dd ?
    PointerToRawData dd ?
    PointerToRelocations dd ?
    PointerToLinenumbers dd ?
    NumberOfRelocations dw ?
    NumberOfLinenumbers dw ?
    Characteristics dd ?
IMAGE_SECTION_HEADER ENDS
    
```

2. PE

가?

5. Section Table

Name1	가 . 가 8
VirtualAddress	RVA, RVA가 1000h PE 400000h 401000h
SizeOfRawData	File alignment PE 가
PointerToRawData	가 , PE
Characteristics	Flag() 가 ,

2. PE

가?

5. Section Table

- IMAGE_SECTION_HEADER

- PE

1) IMAGE_FILE_HEADER NumberOfSections

2) SizeOfHeaders ,

3) 가 (?) --)?

4) PointerToRawData

, SizeOfRawData

가

VirtualAddress

ImageBase

Characteristics

Flag

5)

-
-

--;

;;

2. PE

가?

6. Import Table

- import function : , 가
- import function DLL
- PE Header OptionalHeader data directory

IMAGE_OPTIONAL_HEADER32 STRUCT

....

LoaderFlags dd ?

NumberOfRvaAndSizes dd ?

DataDirectory IMAGE_DATA_DIRECTORY 16 dup(<>)

IMAGE_OPTIONAL_HEADER32 ENDS

- DataDirectory 16 가
- DataDirectory PE 가

2. PE

가?

6. Import Table

Member	Info inside
0	Export symbols
1	Import symbols
2	Resources
3	Exception
4	Security
5	Base relocation
6	Debug
7	Copyright string
8	Unknown
9	Thread local storage (TLS)
10	Load configuration
11	Bound Import
12	Import Address Table
13	Delay Import
14	COM descriptor

- DataDirectory 16 가
- DataDirectory PE 가
- data directory IMAGE_DATA_DIRECTORY

IMAGE_DATA_DIRECTORY STRUCT

VirtualAddress dd ?

isize dd ?

IMAGE_DATA_DIRECTORY ENDS

- VirtualAddress : RVA
- isize : VirtualAddress

2. PE

가?

6. Import Table

- PE (?)
- 1) DOS PE . (IMAGE_DOS_HEADER->IMAGE_NT_HEADERS)..
- 2) optional header data directory
- 3) IMAGE_DATA_DIRECTORY , import symbols , IMAGE_DATA_DIRECTORY(8byte)
1
- 4) data directory 가 가
IMAGE_DATA_DIRECTORY 가

2. PE

가?

6. Import Table

- Import Table
- 2 data directory VirtualAddress
IMAGE_DATA_DIRECTORY STRUCT
VirtualAddress dd ?
isize dd ?
IMAGE_DATA_DIRECTORY ENDS
- Import Table IMAGE_IMPORT_DESCRIPTOR
- PE import
DLL 가 , 10
DLL import 10 가

Member	Info inside
0	Export symbols
1	Import symbols
2	Resources
3	Exception
4	Security
5	Base relocation
6	Debug
7	Copyright string
8	Unknown
9	Thread local storage (TLS)
10	Load configuration
11	Bound Import
12	Import Address Table
13	Delay Import
14	COM descriptor

2. PE

가?

6. Import Table

```

IMAGE_IMPORT_DESCRIPTOR STRUCT
union
  Characteristics dd ?
  OriginalFirstThunk dd ?
ends
TimeDataStamp dd ?
ForwarderChain dd ?
Name1 dd ?
FirstThunk dd ?
IMAGE_IMPORT_DESCRIPTOR ENDS
    
```

```

OriginalFirstThunk alias
IMAGE_THUNK_DATA
IMAGE_THUNK_DATA
IMAGE_IMPORT_BY_NAME
    
```

```

IMAGE_IMPORT_BY_NAME STRUCT
Hint dw ?
Name1 db ?
IMAGE_IMPORT_BY_NAME ENDS
    
```

```

DLL RVA
: "kernel32.dll" RVA
    
```

```

Hint : 가 DLL
export table index
Name1 : Import Function
    
```

```

OriginalFirstThunk
IMAGE_THUNK_DATA RVA
    
```

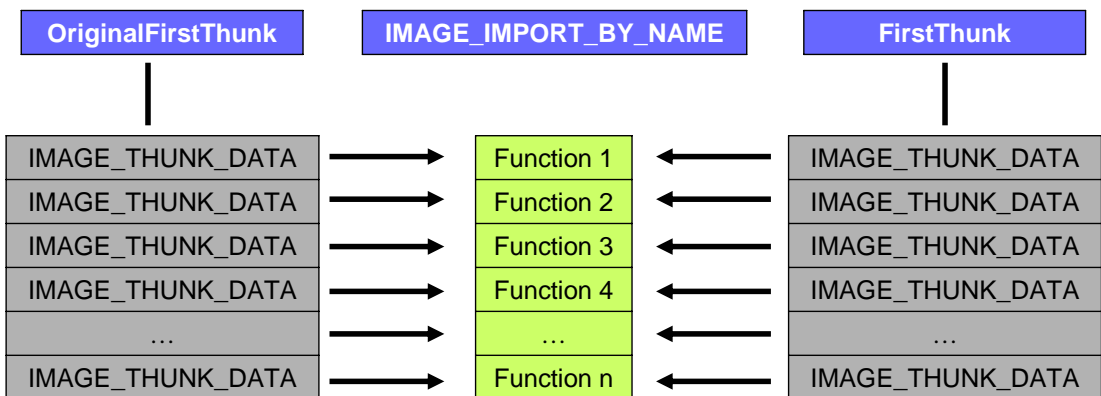
• Import Table IMAGE_IMPORT_DESCRIPTOR

• PE import
 DLL 가 , 10
 DLL import 10 가

2. PE

가?

6. Import Table



• PE Kernel32.dll 10 Import , RVA
 IMAGE_IMPORT_DESCRIPTOR Name1 "kernel32.dll"
 10 IMPORT_THUNK_DATA가 .

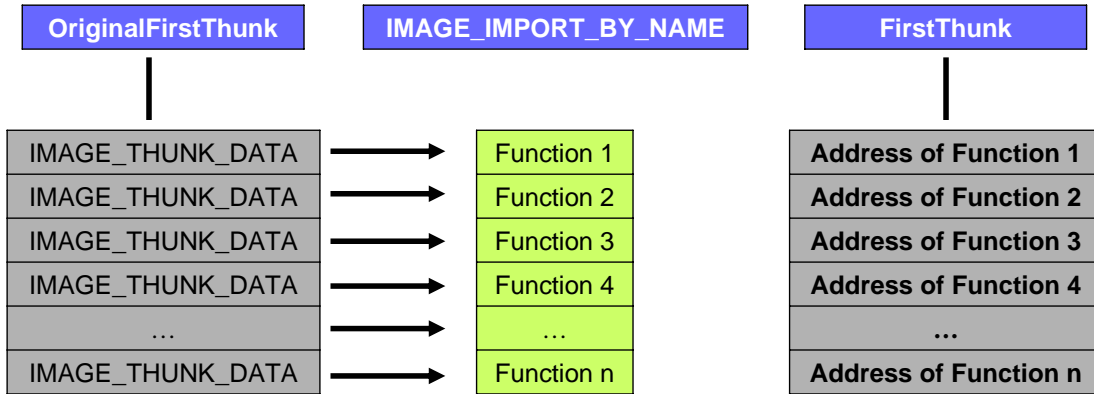
• ?

2. PE

가?

6. Import Table

- PE IMAGE_IMPORT_BY_NAME PE IMAGE_THUNK_DATA import function
- FirstThunk pointing IMAGE_THUNK_DATA
- PE 가 . .



2. PE

가?

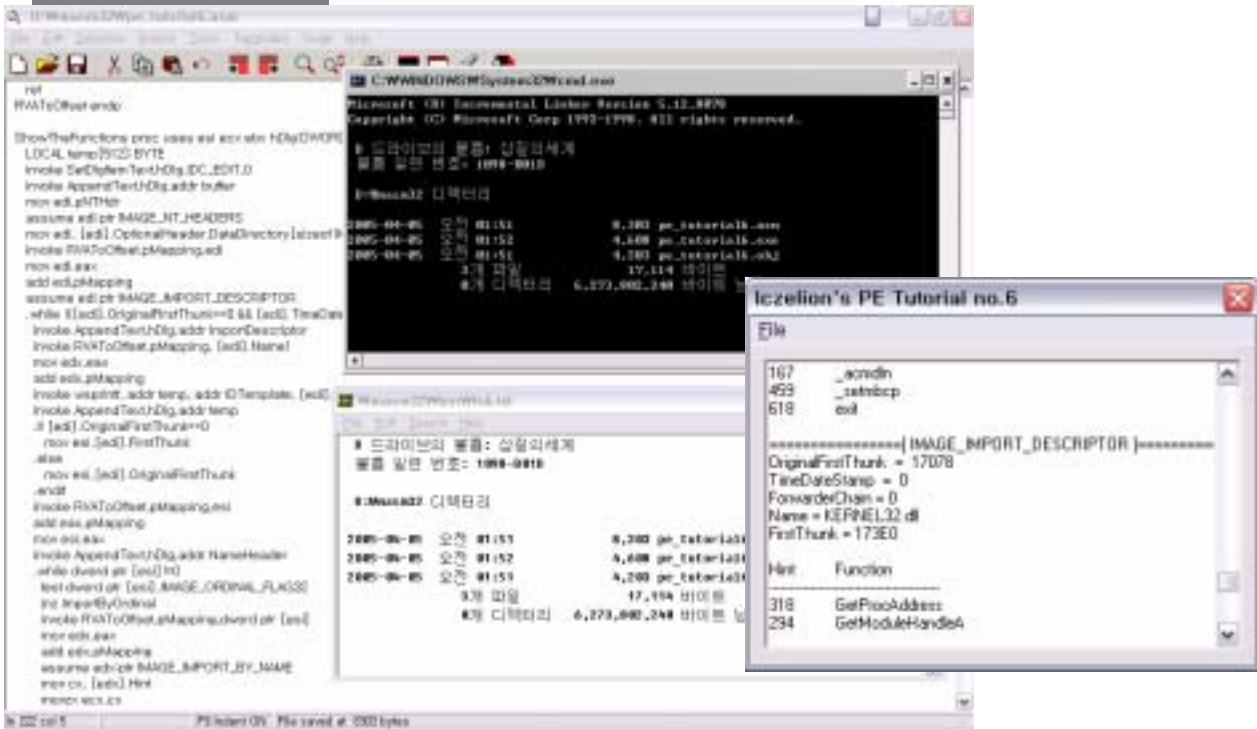
6. Import Table

- PE Import Function
- 1. PE .
- 2. DOS PE .
- 3. OptionalHeader data directory
- 4. data directory (import symbols) 가 VirtualAddress(Import Table RVA)
- 5. IMAGE_IMPORT_DESCRIPTOR 가
- 6. OriginalFirstThunk . If not Zero, RVA 가 OriginalFirstThunk RVA . If OrinigalFirstThunk == Zero, use FirstThunk. OriginalFirstThunk
- 7. IMAGE_ORDINAL_FLAG32 .
- 8. IF MSB == 1 THEN ordinal export , ordinal
- 8. IF MSB == 0 THEN RVA(FirstThunk or OriginalFirstThunk) IMAGE_IMPORT_BY_NAME . Hint ,
- 9. Null byte .
- 10. DLL import . DLL (IMAGE_IMPORT_DESCRIPTOR 가 , 가 NULL .)

2. PE

가?

6. Import Table



2. PE

가?

7. Export Table

- 1. PE 가 DLLs
- 2. Import Function
- 3. DLLs
- 4. PE 가 DLLs Export Table.
- DLL/EXE가 DLL/EXE Export 가
 1. name
 2. ordinal Export
- 1. name : DLL "GetSysConfig" 가 , DLL/EXE "GetSysConfig"
- 2. Ordinal Export
 - : ordinal - DLL 16-bit DLL
 - DLL ordinal 가
 - DLL ordinal

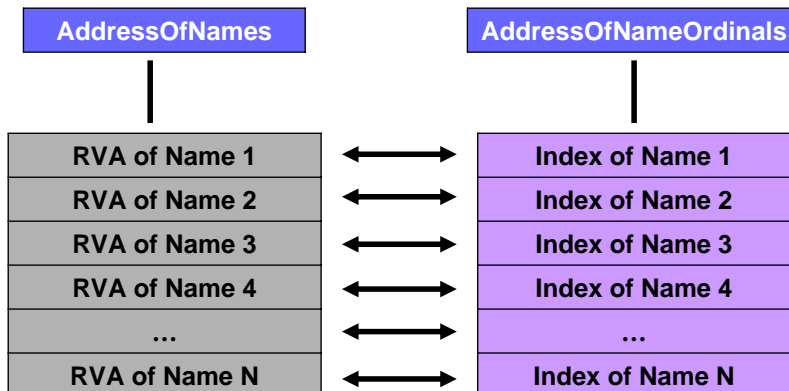
7. Export Table

- (Export Structure)

nName	.
nBase	Address-of-function index ordinal
NumberOfFunctions	Export
NumberOfNames	Export
AddressOfFunctions	RVA. , AddressOfFunctions RVAs가 가
AddressOfNames	RVA RVAs 가
AddressOfNameOrdinals	AddressOfNames ordinal 16-bit 가 RVA

7. Export Table

- Index RVA



7. Export Table

- Export

1. PE
2. data directory Export Table 가
3. Export Table 가 (NumberOfNames)
4. AddressOfName AddressOfNameOrdinals가 가
 , AddressOfNameOrdinals
 AddressOfName 77 RVA
 AddressOfNameOrdinal 77
 NumberOfNames
5. AddressOfNameOrdinal AddressOfFunctions index
 5 AddressOfFunctions 5
 RVA.

7. Export Table

- ordinal

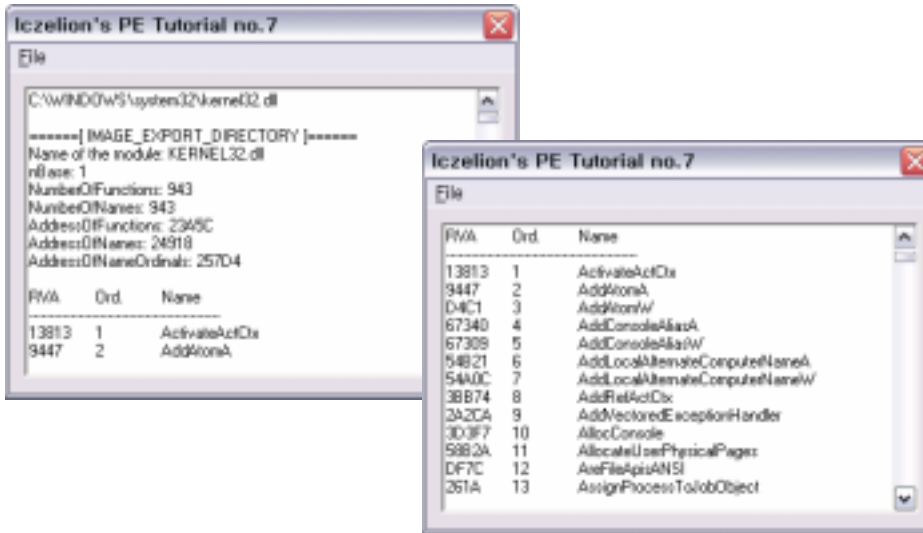
1. PE
2. data directory Export Table RVA
3. Export Table 가 nBase
4. ordinal nBase AddressOfFunctions index
5. Index NumberOfFunctions . If index >= NumberOfFunctions then ordinal
6. AddressOfFunctions index

- ordinal 가

2. PE 가?

7. Export Table

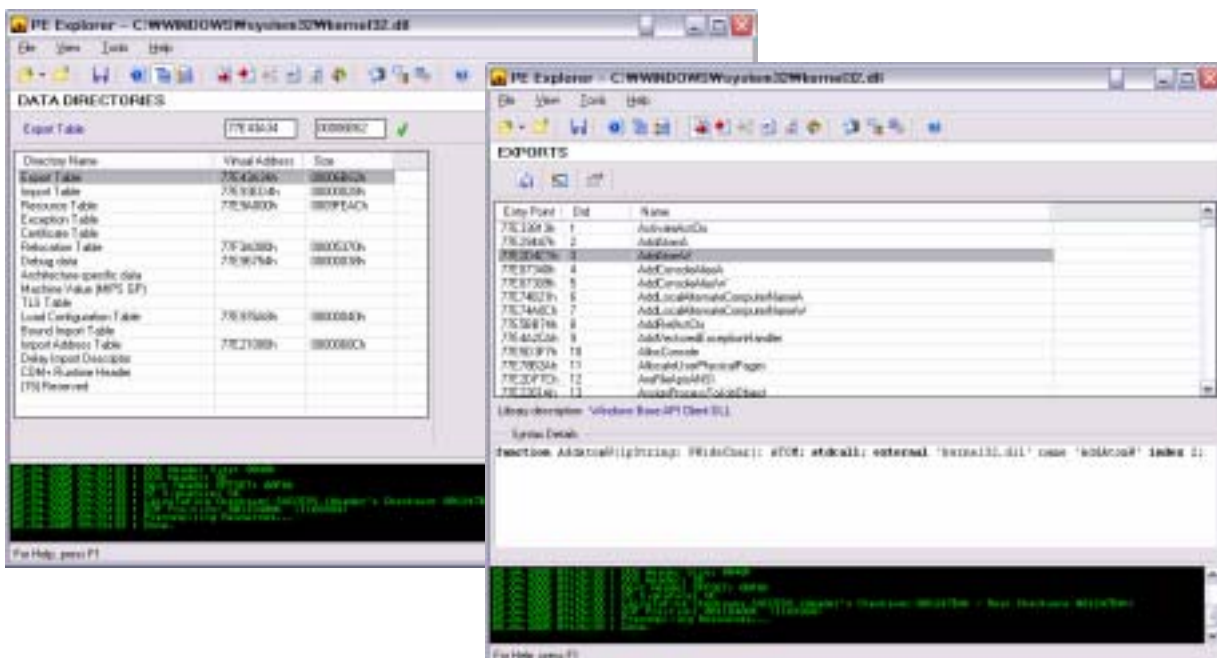
- Export Table . lczelion.

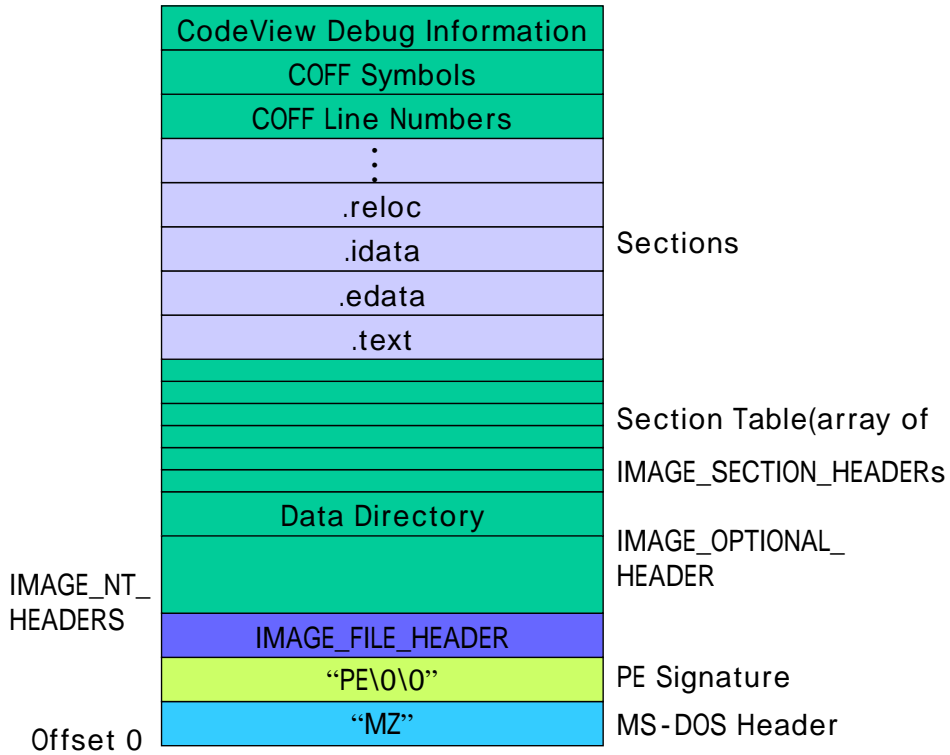


2. PE 가?

7. Export Table

- PE Explorer kernel32.dll Export Table Functions .





- Once Upon A Time In PE – Icezlion
- <http://win32assembly.online.fr/>



Discussion