



계산기 파일은 window 검색 창에서 cal을 검색해서 파일 경로 보기 후 그 위치로 가서 찾았다.

	pFile	Data	Description	Value
calc.exe				
IMAGE_DOS_HEADER	00000000	5A4D	Signature	IMAGE_DOS_SIGNATURE MZ
MS-DOS Stub Program	00000002	0090	Bytes on Last Page of File	

DOS 헤더에서 PE 파일이기 때문에 e_magic 값으로 MZ를 확인했다.

0000003A	0000	Reserved
0000003C	000000D8	Offset to New EXE Header

000000D0	00 00 00 00 00 00 00 00	50 45 00 00 4C 01 04 00PE..L...
000000E0	9D 97 E7 4C 00 00 00 00	00 00 00 00 E0 00 02 01	...L.....
000000F0	0B 01 09 00 00 2E 05 00	00 A6 06 00 00 00 00 00
00000100	6C 2D 01 00 00 10 00 00	00 20 05 00 00 00 00 01	...I..

DOS 헤더에서 NT 헤더가 시작되는 부분(e_ifanew)을 확인했다.

	pFile	Raw Data	Value
calc.exe			
IMAGE_DOS_HEADER	00000040	0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68!..L!Th
MS-DOS Stub Program	00000050	69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F	is program cannot
IMAGE_NT_HEADERS	00000060	74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20	t be run in DOS
Signature	00000070	6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00	mode.....\$.....
IMAGE_FILE_HEADER	00000080	08 73 A6 53 4C 12 C8 00 4C 12 C8 00 4C 12 C8 00	..s.SL...L...L...
IMAGE_OPTIONAL_HEADER	00000090	45 6A 5D 00 45 12 C8 00 4C 12 C9 00 D8 13 C8 00	Ej].E...L...L...
IMAGE_SECTION_HEADER	000000A0	45 6A 5B 00 6D 12 C8 00 45 6A 4B 00 57 12 C8 00	Ej[.m...EjK.W...
IMAGE_SECTION_HEADER	000000B0	45 6A 4C 00 CF 12 C8 00 45 6A 5C 00 4D 12 C8 00	File File M

DOS_STUB 헤더에서 다음과 같이 윈도우 프로그램을 CMD에서 실행했을 경우 출력해줄 에러 메시지가 저장된 것을 확인했다.

calc.exe	pFile	Raw Data	Value
IMAGE_DOS_HEADER	000000D8	50 45 00 00 4C 01 04 00 9D 97 E7 4C 00 00 00 00	PE .L
MS-DOS Stub Program	000000E8	00 00 00 00 E0 00 02 01 0B 01 09 00 00 2E 05 00
IMAGE_NT_HEADERS	000000F8	00 A6 06 00 00 00 00 00 6C 2D 01 00 00 10 00 00
Signature	00000108	00 20 05 00 00 00 00 01 00 10 00 00 00 02 00 00

NT 헤더에서 PE 파일의 시그니처인 45 50 값을 확인했다. (자료형 DWORD_

calc.exe	pFile	Data	Description	Value
IMAGE_DOS_HEADER	000000DC	014C	Machine	IMAGE_FILE_MACHINE_I386
MS-DOS Stub Program	000000DE	0004	Number of Sections	
IMAGE_NT_HEADERS	000000E0	4CE7979D	Time Date Stamp	2010/11/20 Sat 09:40:45 UTC
Signature	000000E4	00000000	Pointer to Symbol Table	
IMAGE_FILE_HEADER	000000E8	00000000	Number of Symbols	
IMAGE_OPTIONAL_HEADER	000000EC	00E0	Size of Optional Header	
IMAGE_SECTION_HEADER .text	000000EE	0102	Characteristics	
IMAGE_SECTION_HEADER .data		0002		IMAGE_FILE_EXECUTABLE_IMAGE
IMAGE_SECTION_HEADER .rsrc		0100		IMAGE_FILE_32BIT_MACHINE
IMAGE_SECTION_HEADER .reloc				

NT 헤더에 FILE 헤더 구조체에 정보를 확인했다.

컴파일한 기계와 시간 정보인 Timestamp와 machine를 확인하고,

섹션의 개수가 4개이고, characteristics를 통해 0002 - 실행 가능하고 0100 - 32bit machine인 것을 확인했다.

calc.exe	pFile	Data	Description	Value
IMAGE_DOS_HEADER	000000F0	010B	Magic	IMAGE_NT_OPTIONAL_HDR32_
MS-DOS Stub Program	000000F2	09	Major Linker Version	
IMAGE_NT_HEADERS	000000F3	00	Minor Linker Version	
Signature	000000F4	00052E00	Size of Code	
IMAGE_FILE_HEADER	000000F8	0006A600	Size of Initialized Data	
IMAGE_OPTIONAL_HEADER	000000FC	00000000	Size of Uninitialized Data	
IMAGE_SECTION_HEADER .text	00000100	00012D6C	Address of Entry Point	
IMAGE_SECTION_HEADER .data	00000104	00001000	Base of Code	
IMAGE_SECTION_HEADER .rsrc	00000108	00052000	Base of Data	
IMAGE_SECTION_HEADER .reloc	0000010C	01000000	Image Base	
BOUND_IMPORT Directory Table	00000110	00001000	Section Alignment	
BOUND_IMPORT DLL Names	00000114	00000200	File Alignment	

Optional 헤더 구조체에서 magic 값이 010B, 즉 32비트인 것을 알 수 있고

실제 코드가 올라가는 .TEXT 영역의 크기를 SIZE OF CODE를 통해 알 수 있고

프로그램의 RVA를 ADDRESS OF ENTRY POINT를 통해 알 수 있고

실제 디버거에서 사용되는 주소인 VA를 구하기 위해 필요한 IMAGEBASE의 값을 알 수 있고

SECTION ALIGNMENT를 통해 메모리에서의 섹션의 최소 단위를 알 수 있고

FILE ALIGNMENT를 통해 파일에서의 섹션 최소 단위를 알 수 있다.

	pFile	Data	Description	Value
calc.exe	00000270	4CE7B9DE	Time Date Stamp	2010/11/20 Sat 12:06:54 UTC
IMAGE_DOS_HEADER	00000274	0098	Offset to Module Name	SHELL32.dll
MS-DOS Stub Program	00000276	0000	Number of Module Forwarder Refs	
IMAGE_NT_HEADERS	00000278	4CE7B9E2	Time Date Stamp	2010/11/20 Sat 12:06:58 UTC
Signature	0000027C	00A4	Offset to Module Name	SHLWAPI.dll
IMAGE_FILE_HEADER	0000027E	0000	Number of Module Forwarder Refs	
IMAGE_OPTIONAL_HEADER	00000280	4CE7B715	Time Date Stamp	2010/11/20 Sat 11:55:01 UTC
IMAGE_SECTION_HEADER .text	00000284	00B0	Offset to Module Name	gdiplus.dll
IMAGE_SECTION_HEADER .data	00000286	0000	Number of Module Forwarder Refs	
IMAGE_SECTION_HEADER .rsrc	00000288	4CE7B706	Time Date Stamp	2010/11/20 Sat 11:54:46 UTC
IMAGE_SECTION_HEADER .reloc	0000028C	00BC	Offset to Module Name	ADVAPI32.dll
BOUND_IMPORT Directory Table	0000028E	0001	Number of Module Forwarder Refs	
BOUND_IMPORT DLL Names	00000290	4CE7B96E	Time Date Stamp	2010/11/20 Sat 12:05:02 UTC
SECTION .text	00000294	00C9	Offset to Module Name	ntdll.DLL
SECTION .data	00000296	0000	Reserved	
SECTION .rsrc	00000298	4CE7B972	Time Date Stamp	2010/11/20 Sat 12:05:06 UTC
SECTION .reloc	0000029C	00D3	Offset to Module Name	OLEAUT32.dll
	0000029E	0000	Number of Module Forwarder Refs	
	000002A0	4CE7BA2F	Time Date Stamp	2010/11/20 Sat 12:08:15 UTC
	000002A4	00E0	Offset to Module Name	UxTheme.dll
	000002A6	0000	Number of Module Forwarder Refs	
	000002A8	4CE7B96F	Time Date Stamp	2010/11/20 Sat 12:05:03 UTC
	000002AC	00EC	Offset to Module Name	ole32.dll
	000002AE	0000	Number of Module Forwarder Refs	
	000002B0	4CE7B71C	Time Date Stamp	2010/11/20 Sat 11:55:08 UTC
	000002B4	00F6	Offset to Module Name	COMCTL32.dll

Import table을 통해서 exe 파일이 실행되기 위해 필요한 pe 파일인 dll등과 같은 파일의 정보를 확인할 수 있다.

	pFile	Raw Data	Value
calc.exe	00000308	53 48 45 4C 4C 33 32 2E 64 6C 6C 00 53 48 4C 57	SHELL32.dll.SHLW
IMAGE_DOS_HEADER	00000318	41 50 49 2E 64 6C 6C 00 67 64 69 70 6C 75 73 2E	API.dll.gdiplus.
IMAGE_NT_HEADERS	00000328	64 6C 6C 00 41 44 56 41 50 49 33 32 2E 64 6C 6C	dll.ADVAPI32.dll
Signature	00000338	00 6E 74 64 6C 6C 2E 44 4C 4C 00 4F 4C 45 41 55	.ntdll.DLL.OLEAU
IMAGE_FILE_HEADER	00000348	54 33 32 2E 64 6C 6C 00 55 78 54 68 65 6D 65 2E	T32.dll.UxTheme.
IMAGE_OPTIONAL_HEADER	00000358	64 6C 6C 00 6F 6C 65 33 32 2E 64 6C 6C 00 43 4F	dll.ole32.dll.CO
IMAGE_SECTION_HEADER .text	00000368	4D 43 54 4C 33 32 2E 64 6C 6C 00 4B 45 52 4E 45	MCTL32.dll.KERNE
IMAGE_SECTION_HEADER .data	00000378	4C 33 32 2E 64 6C 6C 00 55 53 45 52 33 32 2E 64	L32.dll.USER32.d
IMAGE_SECTION_HEADER .rsrc	00000388	6C 6C 00 52 50 43 52 54 34 2E 64 6C 6C 00 57 49	II.RPCRT4.dll.WI
IMAGE_SECTION_HEADER .reloc	00000398	4E 4D 4D 2E 64 6C 6C 00 56 45 52 53 49 4F 4E 2E	NMM.dll.VERSION.
BOUND_IMPORT Directory Table	000003A8	64 6C 6C 00 47 44 49 33 32 2E 64 6C 6C 00 6D 73	dll.GDI32.dll.ms
BOUND_IMPORT DLL Names	000003B8	76 63 72 74 2E 64 6C 6C 00 00 00 00	vcrt.dll....
SECTION .text			
SECTION .data			
SECTION .rsrc			
SECTION .reloc			

Import dll names에서는 이름과 같이 위에서 본 dll 파일들의 이름들만을 확인할 수 있다.

pFile	Data	Description	Value
000001D0	2E 74 65 78	Name	.text
000001D4	74 00 00 00		
000001D8	00052CA1	Virtual Size	
000001DC	00001000	RVA	
000001E0	00052E00	Size of Raw Data	
000001E4	00000400	Pointer to Raw Data	
000001E8	00000000	Pointer to Relocations	
000001EC	00000000	Pointer to Line Numbers	
000001F0	0000	Number of Relocations	
000001F2	0000	Number of Line Numbers	
000001F4	60000020	Characteristics	
		00000020	IMAGE_SCN_CNT_CODE
		20000000	IMAGE_SCN_MEM_EXECUTE
		40000000	IMAGE_SCN_MEM_READ

Section 헤더의 .text 부분에서 다음과 같이 .text 영역이라는 name 값을 확인할 수 있고

.text 섹션의 크기를 virtual size로 확인할 수 있고

Size of raw data값을 통해 파일에서의 섹션 크기를 알 수 있고,

(이 값은 optional 헤더에서 확인한 file alignment의 배수임)

Pointer to raw data를 통해 파일에서 .text의 시작 위치를 알 수 있고

Characteristics를 통해 실행, 읽기가 가능하고 코드 섹션(.text)이라는 것을 확인할 수 있었다.

000001C0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	...
000001D0	2E 74 65 78 74 00 00 00	A1 2C 05 00 00 10 00 00	.text
000001E0	00 2E 05 00 00 04 00 00	00 00 00 00 00 00 00 00	
000001F0	00 00 00 00 20 00 00 60	2E 64 61 74 61 00 00 00	.data
00000200	C0 40 00 00 00 40 05 00	00 42 00 00 00 32 05 00	@ . @ . B . 2 .
00000210	00 00 00 00 00 00 00 00	00 00 00 00 40 00 00 C0	@ .
00000220	2E 72 73 72 63 00 00 00	98 27 06 00 00 90 05 00	.rsrc.
00000230	00 28 06 00 00 74 05 00	00 00 00 00 00 00 00 00	(. . t .
00000240	00 00 00 00 40 00 00 40	2E 72 65 6C 6F 63 00 00	@ . @ . reloc
00000250	3C 3B 00 00 00 C0 0B 00	00 3C 00 00 00 9C 0B 00	< ; <

.text 영역의 Pointer to raw data 값 위치로 가보니 다음과 같이 여러 섹션들의 시작 위치들이 모여있는 것을 확인할 수 있었다.

IMAGE_SECTION_HEADER .data	00000214	00000000	Pointer to Line Numbers	
IMAGE_SECTION_HEADER .rsrc	00000218	0000	Number of Relocations	
IMAGE_SECTION_HEADER .reloc	0000021A	0000	Number of Line Numbers	
BOUND_IMPORT Directory Table	0000021C	C0000040	Characteristics	
BOUND_IMPORT DLL Names				IMAGE_SCN_CNT_INITIALIZED_DATA
SECTION .text		40000000		IMAGE_SCN_MEM_READ
SECTION .data		80000000		IMAGE_SCN_MEM_WRITE
SECTION .rsrc				

.data 섹션의 정보

... IMAGE_SECTION_HEADER .data	0000023C	00000000	Number of Line Numbers	
... IMAGE_SECTION_HEADER .rsrc	00000240	0000	Number of Relocations	
... IMAGE_SECTION_HEADER .reloc	00000242	0000	Number of Line Numbers	
... BOUND_IMPORT Directory Table	00000244	40000040	Characteristics	
... BOUND_IMPORT DLL Names			00000040	IMAGE_SCN_CNT_INITIALIZED_DATA
... SECTION .text			40000000	IMAGE_SCN_MEM_READ
... SECTION .data				

.rsrc 섹션의 정보이다.

섹션들은 Characteristics 값을 통해서 구분이 가능하다.