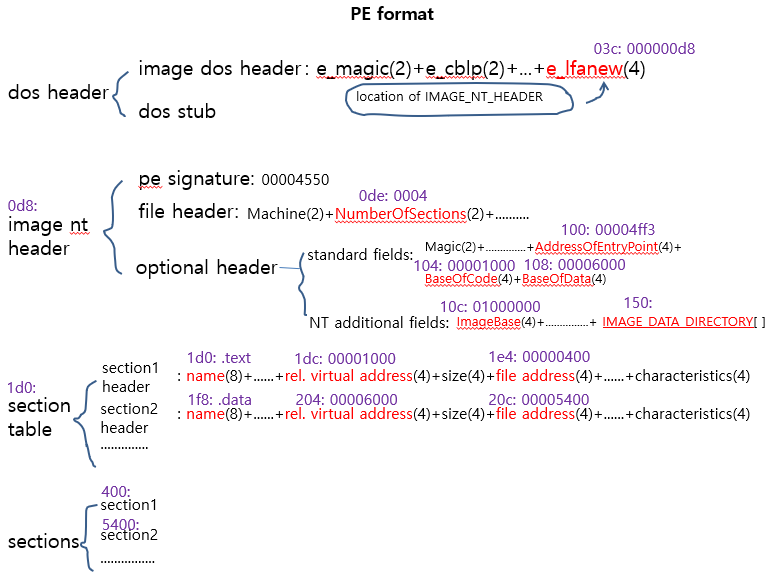
**1) Open cacls.exe with HxD and extract following informtion: e\_lfanew, NumberOfSections, AddressOfEntryPoint, BaseOfCode, BaseOfData, and ImageBase. Check your answer with PEview. (If you are using MacOS, use xxd instead of HxD and skip PEview part.)**



e\_lfanew, [3C,3F]



: offset of IMAGE\_NT\_HEADER



NumberOfSections, [DE,DF]



: Section의 개수는 00 04 = 4

Section으로는 .text .data .rsrc .reloc 4가지를 가지고 있습니다

AddressOfEntryPoint, [100,103]



00 00 4f f3. program starting address(the address of the first instruction). rva(entry point). eip=AddressOfEntryPoint+ImageBase.

File\_offset of AddressOfEntryPoint= rva(AddressOfEntryPoint)-rva(code)+file\_offset(code)

= 4ff3-1000+400=43f3.

BaseOfCode, [104,107]



1000. rva(code). code segment starts at ImageBase+BaseOfCode

BaseOfData, [108,10B]



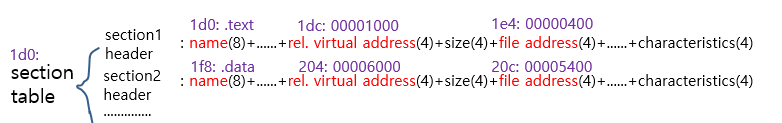
6000. rva(data). data segment starts at ImageBase+BaseOfData

and ImageBase. [10C,10F]



1000000. Image Base is the starting address of a process in its virtual address space. This is the only address which is not rva.

**2) How many sections are there in cacls.exe? Show the name, rva, and file offset of all section.**



000001D0 위치를 확인해봐야 합니다.





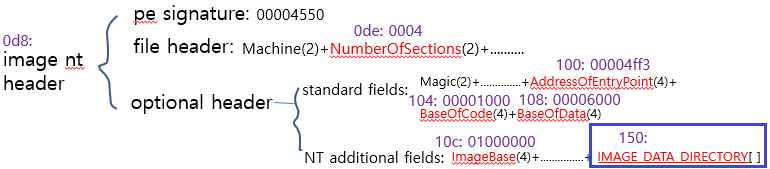


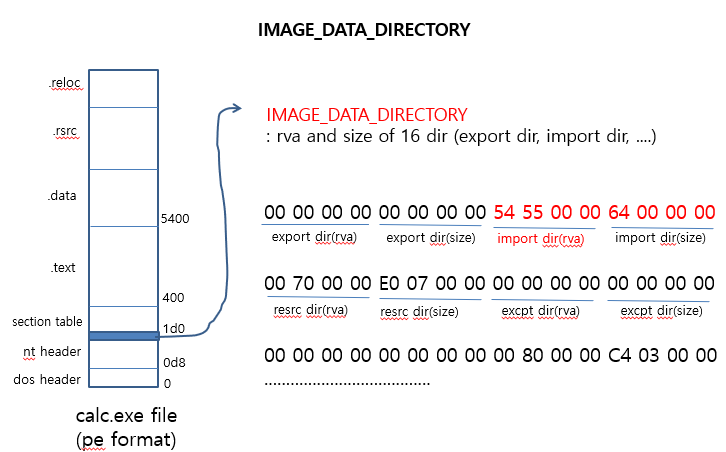


각 부분의 rva / size / file 정보를 표로 정리하면 다음과 같습니다.

|  |  |  |  |
| --- | --- | --- | --- |
|  | rva | Size of Raw data | File offset |
| .text | **1000** | 5000 | **400** |
| .data | **6000** | 200 | **5400** |
| .rsrc | **7000** | 800 | **5600** |
| .reloc | **8000** | 600 | **5E00** |

**3) What is the file offset for IMAGE\_DATA\_DIRECTORY array? What is the file offset of the import directory? How many DLLs do you see in the import directory?**





00000150 부터 16 [byte] 만큼의 위치를 확인해 봐야합니다.



각 부분 순서대로,

150-153: export dir virtual addr. 00

154-157: export dir size. 00

158-15b: import dir. virtual addr. 55 54

15c-15f: import dir. size. 64

160-167: resource dir. virt addr 70 00, size 7e0

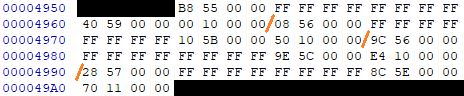
168-16f: exception dir. 00

를 의미합니다.

**What is the file offset of the import directory? : 54 55 00 00**

file offset for 5554: 5554 (rva of imp dir) -1000 (rva of .text) + 400 (file offset of .text) = 4954

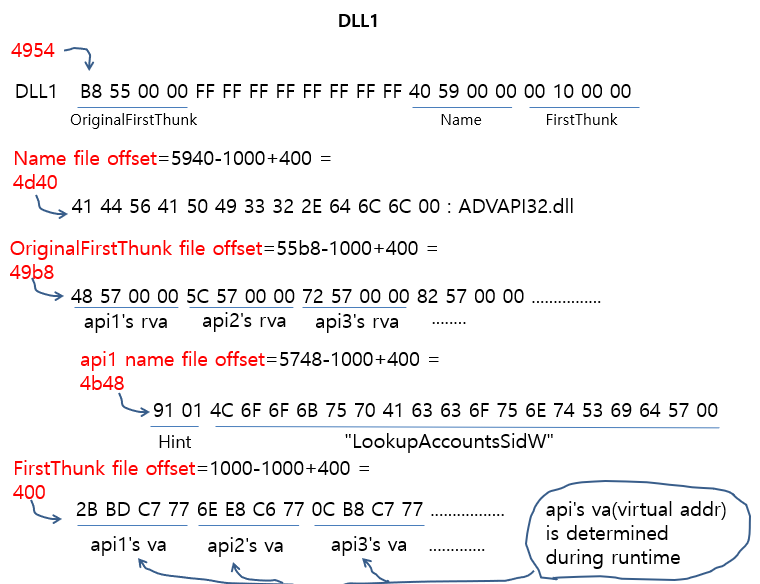
= 00004954 부터 64 [byte] 만큼 읽어줍니다.



4개의 DLL 이 존재합니다.

∵ 각 DLL entry는 0x14 [byte] 이므로, 64/14=5 개의DLL을 import하고있는 것처럼 보이나, 마지막 하나는 NULL entry이므로 결국 4개의 DLL을 import하는 것 입니다.

**4) Show the file offset of OriginalFirstThunk, Name, and FirstThunk for all DLLs in cacls.exe.**

****

위 과정을 각 DLL 4개 모두에서 진행해줍니다.

**DLL 1)** B8 55 00 00 / FF FF FF FF FF FF FF FF / 40 59 00 00 / 00 10 00 00

각 위치에서 -1000 + 400 을 해줍니다

결과는 위와 같습니다.

**DLL 2)** 08 56 00 00 / FF FF FF FF FF FF FF FF / 10 5B 00 00 / 50 10 00 00

각 위치에서 -6000 + 5400 을 해줍니다.

Original First Thunk: 5608 – 6000 + 5400 = 22024 – 24576 + 21504 = 18952 => 4A08

Name: 5B10 – 6000 + 5400 = 23312 – 24576 + 21504 = 20240 => 4F10

First Thunk: 1050 – 6000 + 5400 = 4176– 24576 + 21504 = 654 => 28E

**DLL 3)** 9C 56 00 00 / FF FF FF FF FF FF FF FF / 9E 5C 00 00 / E4 10 00 00

각 위치에서 -7000 + 5600 을 해줍니다.

Original First Thunk: 569C -7000 + 5600 = 22172 – 28672 + 22016 = 15516 => 3C9C

Name: 5C9E -7000 + 5600 = 23710 – 28672 + 22016 = 17054 => 429E

First Thunk: 10E4 -7000 + 5600 = 4323 – 28672 + 22016 = -4240 => -1090 (?)

**DLL 4)** 28 57 00 00 / FF FF FF FF FF FF FF FF / 8C 5E 00 00 / 70 11 00 00

각 위치에서 -8000 + 5E00 을 해줍니다.

Original First Thunk: 5728 -8000 + 5E00 = 22312 – 32768 + 24064 = 13608 => 3528

Name: 5E8C -8000 + 5E00 = 24204 – 32768 + 24064 = 15500 => 3C8C

First Thunk: 1170 -8000 + 5E00 = 4464 – 32768 + 24064 = -4240 => -1090 (?)

**5) Go to the file offset you found in 4) and show names, first 5 API names, and first 5 API function addresses (which are yet garbage addresses) for all DLLs in cacls.exe. Do not use PEview for this problem.**

2)

4A08:



4F10:



28E:



3)

3C9C:



429E:



-1090: (?) 계산에 문제가 있는 것 같습니다.

4)

3528:



3C8C:



-1090: (?) 계산에 문제가 있는 것 같습니다.

마지막 값이 이상하므로, 계산이 잘못된 것 같습니다.

다시 offset 값을 구한 후 재 제출 하도록 하겠습니다!

감사합니다.