IAT

SCP 20학번 박준형

목차 table of contents

IAT와 DLL의 개념

IAT(Import Address Table)

IAT에는 Windows 운영체제의 핵심 개념인 process, memory, DLL 구조 등에 대한 내용이 함축되어 있습니다. IAT란 프로그램이 어떤 라이브러리에서 어떤 함수를 사용하고 있는지를 기술하는 테이블입니다.

DLL(Dynamic Linked Li-brary)

DLL은 프로그램에 라이브러리를 포함시키지 않고 별도의 파일(DLL)로 구성하여 필요할 때마다 불러 쓰는 방식으로, 한 번 로딩된 DLL은 다른 process들이 공유해서 사용함으로써 memory를 효율적으로 사용할 수 있는 방법입니다.

Explicit Linking 방법 (명시적 호출)

프로그램 내부에서 사용되는 순간에 로딩하고 끝나면 해제

Implicit Linking 방법(암시적 호출)

프로그램이 시작할 때 같이 로딩되어 종료

→IAT는 Implicit Linking에 대한 메커니즘을 제공함

OllyDbg에서 IAT 알아보기

Part 2 Ollydbg에서 IAT 알아보기

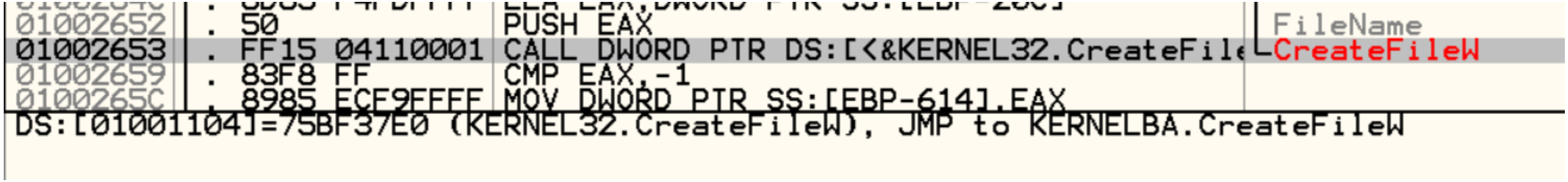
```
01002645  . 6A 01          PUSH 1                                   ShareMode = FILE_SHARE_R
01002647  . 68 00000080    PUSH 80000000                            Access = GENERIC_READ
0100264C  . 8D85 F4FDFFFF  LEA EAX,DWORD PTR SS:[EBP-20C]
01002652  . 50             PUSH EAX                                 FileName
01002653  . FF15 04110001  CALL DWORD PTR DS:[<&KERNEL32.CreateFileW>]   CreateFileW
01002659  . 83F8 FF        CMP EAX,-1
0100265C  . 8985 ECF9FFFF  MOV DWORD PTR SS:[EBP-614],EAX
01002662  .v0F84 91000000  JE notepad.010026F9
01002668  . 57             PUSH EDI                                 pOverlapped => NULL
01002669  . 8D8D F0F9FFFF  LEA ECX,DWORD PTR SS:[EBP-610]           pBytesRead
0100266F  . 51             PUSH ECX                                 BytesToRead = 400 (1024.
01002670  . 68 00040000    PUSH 400
01002675  . 8D8D F4F9FFFF  LEA ECX,DWORD PTR SS:[EBP-60C]           Buffer
0100267B  . 51             PUSH ECX                                 hFile
0100267C  . 50             PUSH EAX
0100267D  . FF15 00110001  CALL DWORD PTR DS:[<&KERNEL32.ReadFile>]   ReadFile
01002683  . 85C0           TEST EAX,EAX
01002685  .v7E 66          JLE SHORT notepad.010026ED
01002687  . 39BD F0F9FFFF  CMP DWORD PTR SS:[EBP-610],EDI
0100268D  .v74 5E          JE SHORT notepad.010026ED                Arg2
0100268F  . FFB5 F0F9FFFF  PUSH DWORD PTR SS:[EBP-610]              Arg1
01002695  . 8D85 F4F9FFFF  LEA EAX,DWORD PTR SS:[EBP-60C]           notepad.0100239D
0100269B  . 50             PUSH EAX
0100269C  . E8 FCFCFFFF    CALL notepad.0100239D                    Switch (cases 1..3)
010026A1  . 8B0D B4900001  MOV ECX,DWORD PTR DS:[10090B4]
010026A7  . A3 30900001    MOV DWORD PTR DS:[1009030],EAX
010026AC  . 48             DEC EAX
010026AD  .v74 16          JE SHORT notepad.010026C5
010026AF  . 48             DEC EAX
DS:[01001104]=765137E0 (KERNEL32.CreateFileW), JMP to KERNELBA.CreateFileW
```

Address	Hex dump	ASCII
01009000	00 00 00 00 D4 70 00 01 00 00 00 00 00 00 00 00	....?.☺.....
01009010	00 00 00 00 00 00 00 00 78 00 00 00 01 00 00 00	........x...☺
01009020	4E 00 6F 00 74 00 65 00 70 00 61 00 64 00 00 00	N.o.t.e.p.a.d...
01009030	FF FF FF FF 01 00 00 00 02 00 00 00 03 00 00 00	....☺...☻...
01009040	04 00 00 00 05 00 00 00 06 00 00 00 07 00 00 00	♦...♣...♠...•
01009050	08 00 00 00 09 00 00 00 0A 00 00 00 0B 00 00 00	◘.....○...♂...
01009060	0C 00 00 00 0D 00 00 00 0E 00 00 00 0F 00 00 00	.....♪...♫...☼...
01009070	10 00 00 00 11 00 00 00 12 00 00 00 13 00 00 00	►...◄...↕...‼...
01009080	2D 00 00 00 14 00 00 00 15 00 00 00 16 00 00 00	-...¶...§...▬...

000DFF74	765100C9	
000DFF78	00205000	
000DFF7C	765100B0	
000DFF80	000DFFDC	
000DFF84	77187B1E	
000DFF88	00205000	
000DFF8C	DF26E018	
000DFF90	00000000	
000DFF94	00000000	
000DFF98	00205000	

```
765137DE  CC               INT3
765137DF  CC               INT3
765137E0 -FF25 1C105776    JMP DWORD PTR DS:[<&api-ms-win-core-file-l1- KERNELBA.CreateFileW
765137E6  CC               INT3
765137E7  CC               INT3
DS:[7657101C]=76FE6CB0 (KERNELBA.CreateFileW)
```

Part 2 왜 CALL 명령어로 더 간단하게 표현하지 않을까?



↳CALL 765137E0이 더 간단하게 표현할 수 있지 않나?

1. 프로그램을 컴파일 하는 순간에는 notepad.exe 프로그램이 어떤 환경에서 실행될지 알 수 없음.

2. 기존에 DLL이 이미 로딩된 상태에서 다른 DLL이 로딩될 경우, DLL Relocation이 발생하므로 실제 함수의 주소를 하드 코딩할 수 없음.

IMAGE_IMPORT_DESCRIPTOR 구조체

Part 3 IMAGE_IMPORT_DESCRIPTOR

```
typedef struct _IMAGE_IMPORT_DESCRIPTOR {
        union {
        DWORD Characteristics;
        DWORD OriginalFirstThunk;  INT(Import Name Table)의 주소 (RVA)
        };
    DWORD TimeDateStamp;
    DWORD ForwarderChain;
    DWORD Name;  Library 이름 문자열의 주소(RVA)
    DWORD FirstThunk;   IAT(Import Address Table)의 주소 (RVA)

} IMAGE_IMPORT_DESCRIPTOR;

typedef struct _IMAGE_IMPORT_BY_NAME {
        WORD Hint;      //ordinal
    BYTE Name[1];       //function name string
} IMAGE_IMPORT_BY_NAME, *PIMAGE_IMPORT_BY_NAME;
```

Part 2 PE로더가 IMPORT 함수 주소를 IAT에 입력하는 순서

(1) IID(IMAGE_IMPORT_DESCRIPTOR)의 Name 멤버를 읽어 "kernel32.dll" 문자열을 얻는다.

(2) 해당 라이브러리를 LoadLibrary("kernel32.dll")을 통하여 로딩한다.

(3) IID의 OriginalFirstThunk 멤버를 읽어서 INT 주소를 얻는다.

(4) INT에서 배열의 값을 하나씩 읽어 해당 IMAGE_IMPORT_BY_NAME(RVA)를 얻는다.

(5) IMAGE_IMPORT_BY_NAME의 Hint(ordinal) 또는 Name 항목을 이용하여 해당 함수의 시작 주소를 얻는다. (GetProcAddresss 이용)

(6) IID의 FirstThunk멤버를 읽어서 IAT 주소를 얻는다.

(7) 해당 IAT 배열 값에 위에서 구한 함수의 주소를 입력한다.

(8) INT가 끝날 때까지(NULL) (4) ~ (7)을 반복한다.

Notepad.exe를 이용한 실습

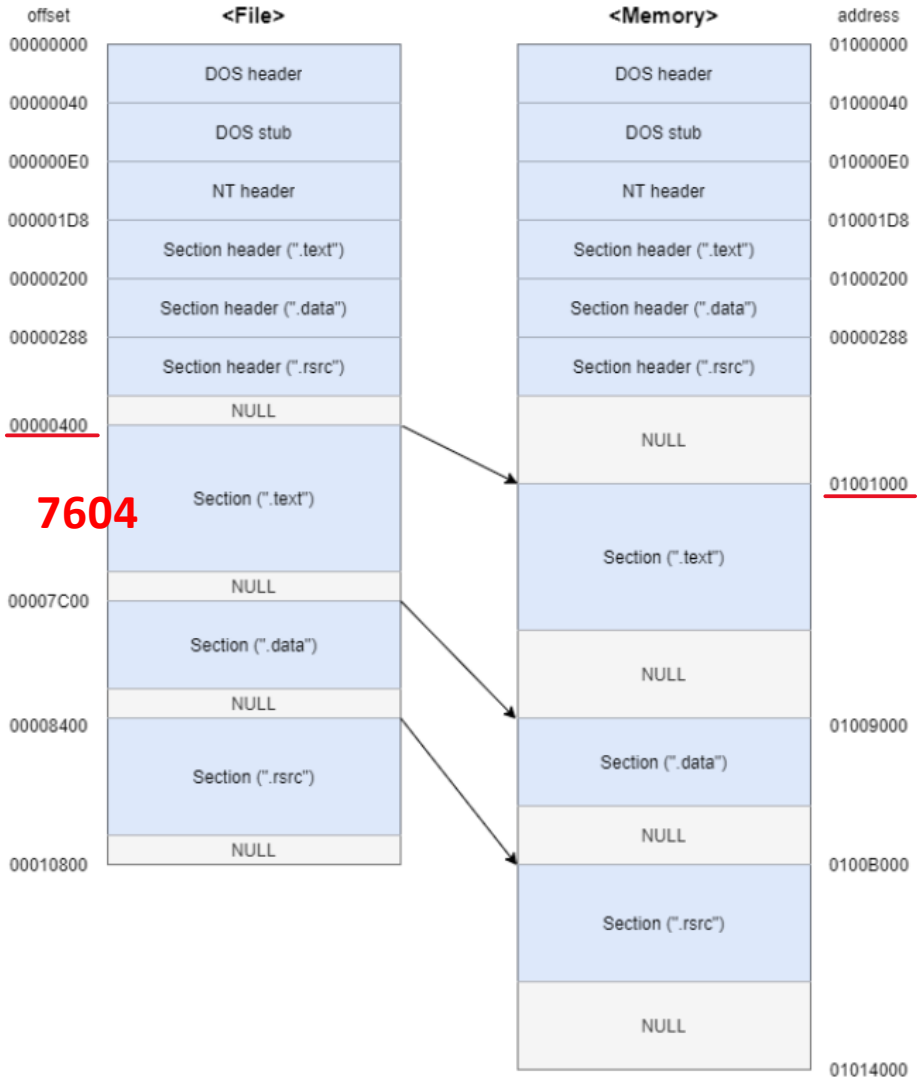
Part 3 Notepad.exe를 이용한 실습

IID가 위치한 곳 = IMAGE_OPTIONAL_HEADER32.DataDirectory[1].VirtualAddress 값

```
00000120  05 00 01 00 05 00 01 00 04 00 00 00 00 00 00 00  ................
00000130  00 40 01 00 00 04 00 00 CE 26 01 00 02 00 00 80  .@......Î&.....€
00000140  00 00 04 00 00 10 01 00 00 00 10 00 00 10 00 00  ................
00000150  00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00  ................
00000160  04 76 00 00 C8 00 00 00 00 B0 00 00 04 83 00 00  .v..È....°...ƒ..
00000170  상대 주소(RVA) / SIZE  0 00 00 00 00 00 00 00 00  ................
00000180  00 00 00 00 00 00 00 00 50 13 00 00 1C 00 00 00  ........P.......
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001A0  00 00 00 00 00 00 00 00 A8 18 00 00 40 00 00 00  ........¨...@...
```

Part 3 RVA = 7604일때, File offset 값은?

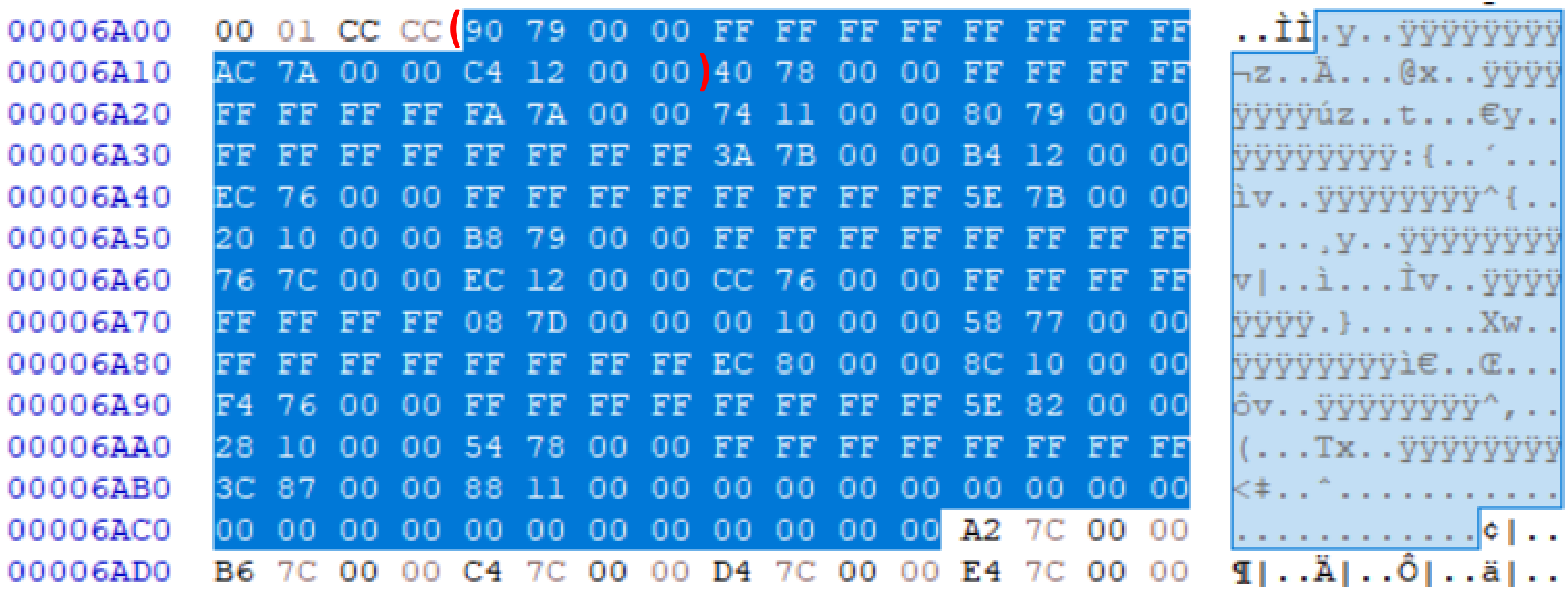
RAW(offset) = RVA - VirtualAddress + PointerToRawData
(Offset 값 = 상대 주소 – 메모리에서 섹션의 시작 주소 + 파일섹션에서의 시작 위치)



RVA = 7604일때, File offset?

RAW = 7604(RVA) - 1000(Virutal Address) + 400(PointerToRawData) = 6A04

Part 3 Notepad.exe를 이용한 실습



```
00006A00  00 01 CC CC (90 79 00 00 FF FF FF FF FF FF FF FF  ..ÌÌ.y..ÿÿÿÿÿÿÿÿ
00006A10  AC 7A 00 00 C4 12 00 00 )40 78 00 00 FF FF FF FF  ¬z..Ä...@x..ÿÿÿÿ
00006A20  FF FF FF FF FA 7A 00 00 74 11 00 00 80 79 00 00  ÿÿÿÿúz..t...€y..
00006A30  FF FF FF FF FF FF FF FF 3A 7B 00 00 B4 12 00 00  ÿÿÿÿÿÿÿÿ:{..´...
00006A40  EC 76 00 00 FF FF FF FF FF FF FF FF 5E 7B 00 00  ìv..ÿÿÿÿÿÿÿÿ^{..
00006A50  20 10 00 00 B8 79 00 00 FF FF FF FF FF FF FF FF   ...¸y..ÿÿÿÿÿÿÿÿ
00006A60  76 7C 00 00 EC 12 00 00 CC 76 00 00 FF FF FF FF  v|..ì...Ìv..ÿÿÿÿ
00006A70  FF FF FF FF 08 7D 00 00 00 10 00 00 58 77 00 00  ÿÿÿÿ.}......Xw..
00006A80  FF FF FF FF FF FF FF FF EC 80 00 00 8C 10 00 00  ÿÿÿÿÿÿÿÿì€..Œ...
00006A90  F4 76 00 00 FF FF FF FF FF FF FF FF 5E 82 00 00  ôv..ÿÿÿÿÿÿÿÿ^‚..
00006AA0  28 10 00 00 54 78 00 00 FF FF FF FF FF FF FF FF  (...Tx..ÿÿÿÿÿÿÿÿ
00006AB0  3C 87 00 00 88 11 00 00 00 00 00 00 00 00 00 00  <‡..ˆ............
00006AC0  00 00 00 00 00 00 00 00 00 00 00 00 A2 7C 00 00  ............¢|..
00006AD0  B6 7C 00 00 C4 7C 00 00 D4 7C 00 00 E4 7C 00 00  ¶|..Ä|..Ô|..ä|..
```

Part 3 Notepad.exe를 이용한 실습

Member	RVA	RAW(File offset 값)
OriginalFirstThunk(INT) 00006A00 00 01 CC CC 90 79 00 00	00007990	00006D90
TimeDateStamp 00006A00 00 01 CC CC 90 79 00 00 FF FF FF FF	FFFFFFFF	-
ForwarderChain 00006A00 00 01 CC CC 90 79 00 00 FF FF FF FF FF FF FF FF	FFFFFFFF	-
Name 00006A10 AC 7A 00 00	00007AAC	00006EAC
FirstThunk(IAT) 00006A10 AC 7A 00 00 C4 12 00 00	000012C4	000006C4

Part 3 Notepad.exe를 이용한 실습 - 1. 라이브러리 이름(Name)

Name	00007AAC	00006EAC
00006A10 AC 7A 00 00		

```
00006EA0  50 72 69 6E 74 44 6C 67 45 78 57 00 63 6F 6D 64  PrintDlgExW.comd
00006EB0  6C 67 33 32 2E 64 6C 6C 00 00 03 01 53 68 65 6C  lg32.dll.....Shel
00006EC0  6C 41 62 6F 75 74 57 00 1F 00 44 72 61 67 46 69  lAboutW...DragFi
```

Part 3 Notepad.exe를 이용한 실습–2.OriginalFirstThunk - INT(Import Name Table)

OriginalFirstThunk(INT)		
00006A00 00 01 CC CC 90 79 00 00	00007990	00006D90

```
00006D90  7A 7A 00 00 5E 7A 00 00 9E 7A 00 00 50 7A 00 00   zz..^z..žz..Pz..
00006DA0  40 7A 00 00 8A 7A 00 00 6A 7A 00 00 14 7A 00 00   @z..Šz..jz...z..
00006DB0  2C 7A 00 00 00 00 00 00 DC 7B 00 00 D4 7B 00 00   ,z......Ü{..Ô{..
00006DC0  CA 7B 00 00 C2 7B 00 00 B6 7B 00 00 EA 7B 00 00   Ê{..Â{..¶{..ê{..
```

Part 3 Notepad.exe를 이용한 실습-IMAGE_IMPORT_BY_NAME

7A7A(RVA) -> 6E7A(RAW)

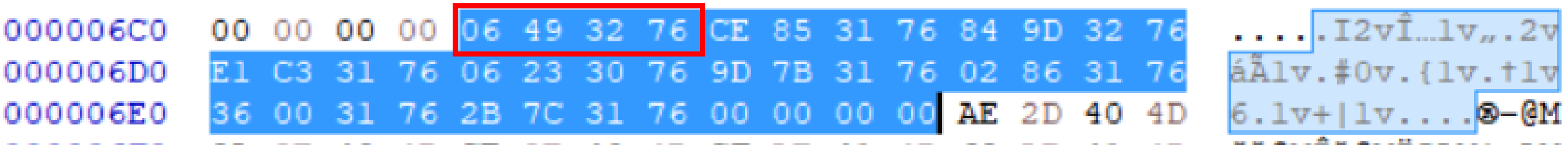
고유번호

```
00006E70   61 63 65 54 65 78 74 57 00 00 0F 00 50 61 67 65   aceTextW....Page
00006E80   53 65 74 75 70 44 6C 67 57 00 0A 00 47 65 74 4F   SetupDlgW...GetO
00006E90   70 65 6E 46 69 6C 65 4E 61 6D 65 57 00 00 12 00   penFileNameW....
```

Part 3 Notepad.exe를 이용한 실습-FirstThunk - IAT(Import Address Table)

FirstThunk(IAT)

00006A10 AC 7A 00 00 C4 12 00 00 000012C4 000006C4



Q&A