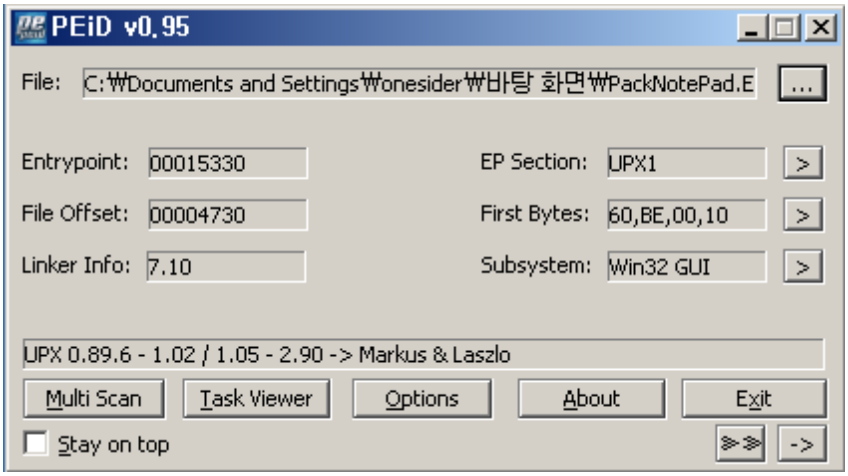
실행압축과
IAT Table 추적

작성일 : 2010.3.30

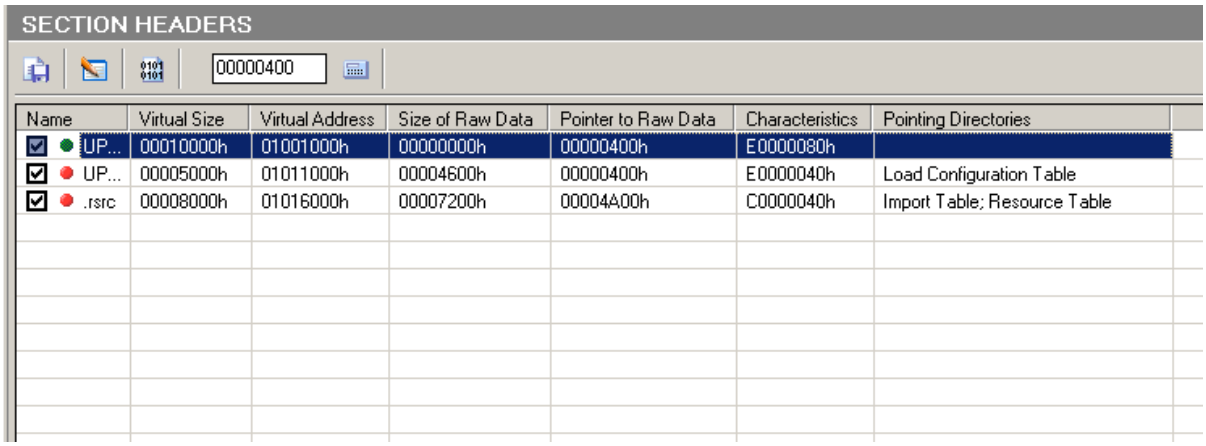
작성자 : onesider@gmail.com

실행 파일 포맷을 공부 하던 중에 IAT Table에 대하여 흥미가 생겼다.
사실 일반 파일에서 IAT Table은 Optional Header의 Data Directories에서 Import Table의 Rva(0x180) 값과 Size(0x184)값을 구할 수 있다.
사실 이 문서를 작성하는 주된 이유는 패킹을 풀면 일반적으로 OEP를 찾고 DUMP를 떠서 IAT Table을 Rebuild 작업을 해준다. (적어도 필자는 그렇게 한다)
이때 IAT Table을 Rebuild 해주는 툴의 역할이 궁금해졌다..
즉 Import REConstructor이 무슨짓(?)을 하는지 알아보기로 하였다.

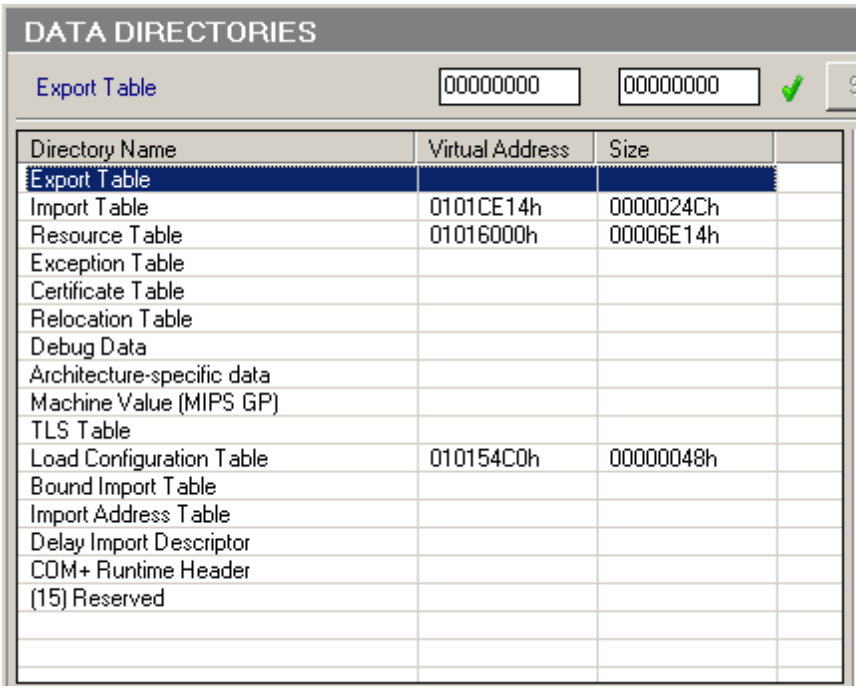


노트 패드를 upx로 패킹하였다.

여기서 확인하고 넘어 갈 것들이 있다.



섹션은 3개 (두 번째 섹션에 Import Table이 포함되어 있음)
SectionAlignment = 0x1000
FileAlignment = 0x200
ImageBase = 0x01000000
Address Of Entry Point = 0x01015330(ImageBase 를 더해준 값)



임포트 테이블은 0101CE14를 가르키고 있다.

(Pe Explorer는 희안하게 ImageBase값을 알아서 더해준다..)

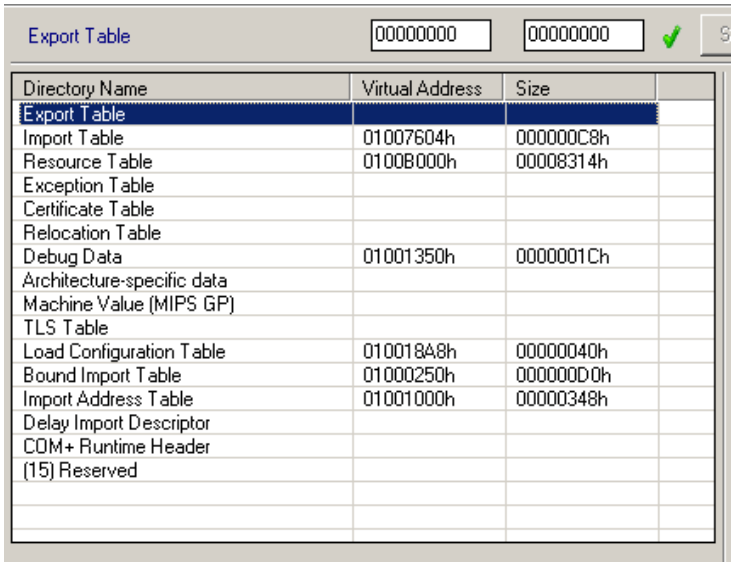
ImageBase가 없는 원래의 값은 1CE14 이므로

400+(1CE14 – Rva(11000)) = "디스크상의 Import Table위치" 가 나옵니다.

하지만 디스크상에 존재하지 않죠.. -_-;;(C214)

```
0000BBD0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000BBE0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000BBF0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

요기서 끝... -_-;; 음..같은 방법으로 패킹되지 않은 원본파일의 Import Table을 찾아보도록 하겠습니다.



패킹 안된 메모장은 7604에 위치한다고 하는군요..

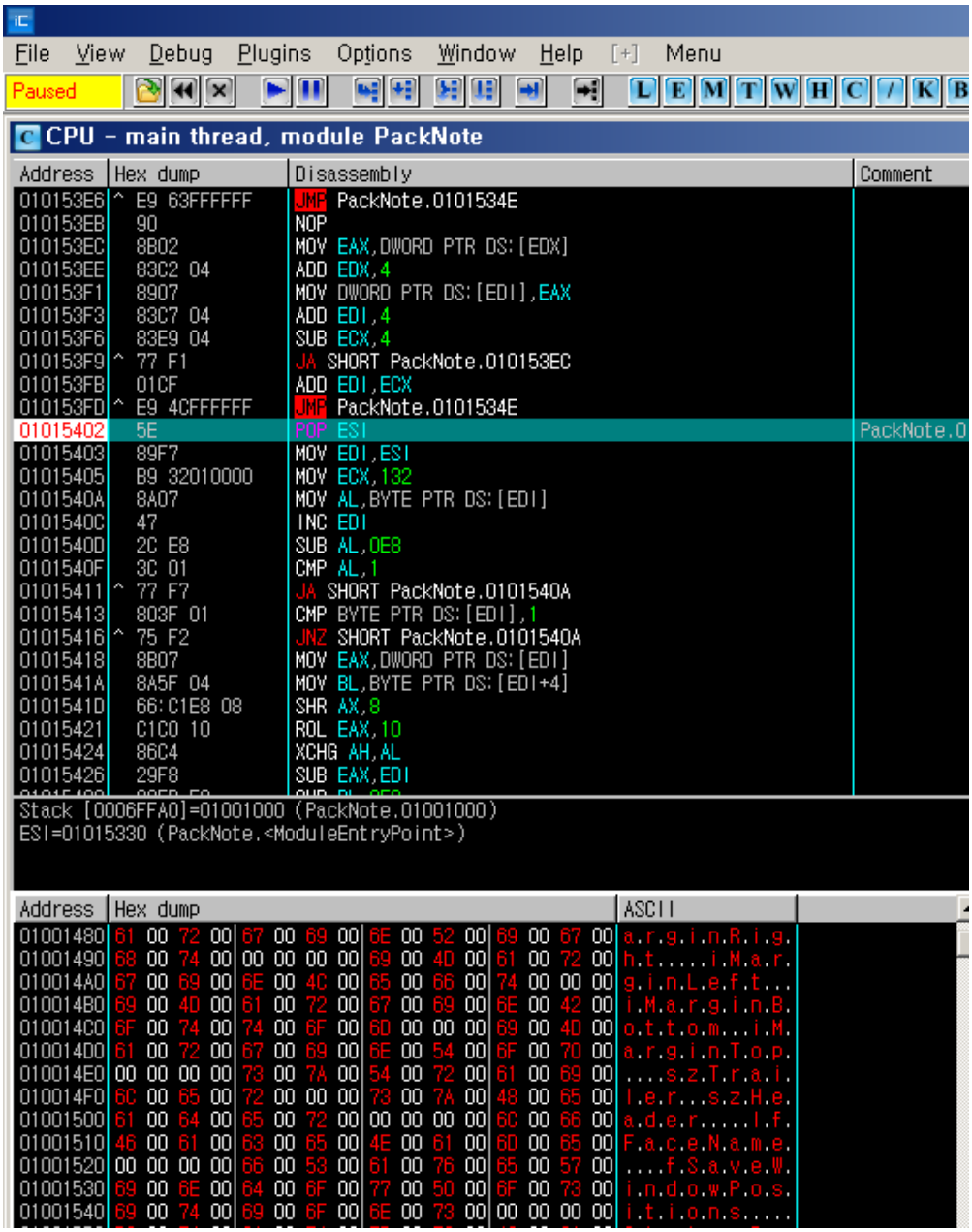
1번(.text) 섹션입니다. 같은 방법으로 계산해보죠..

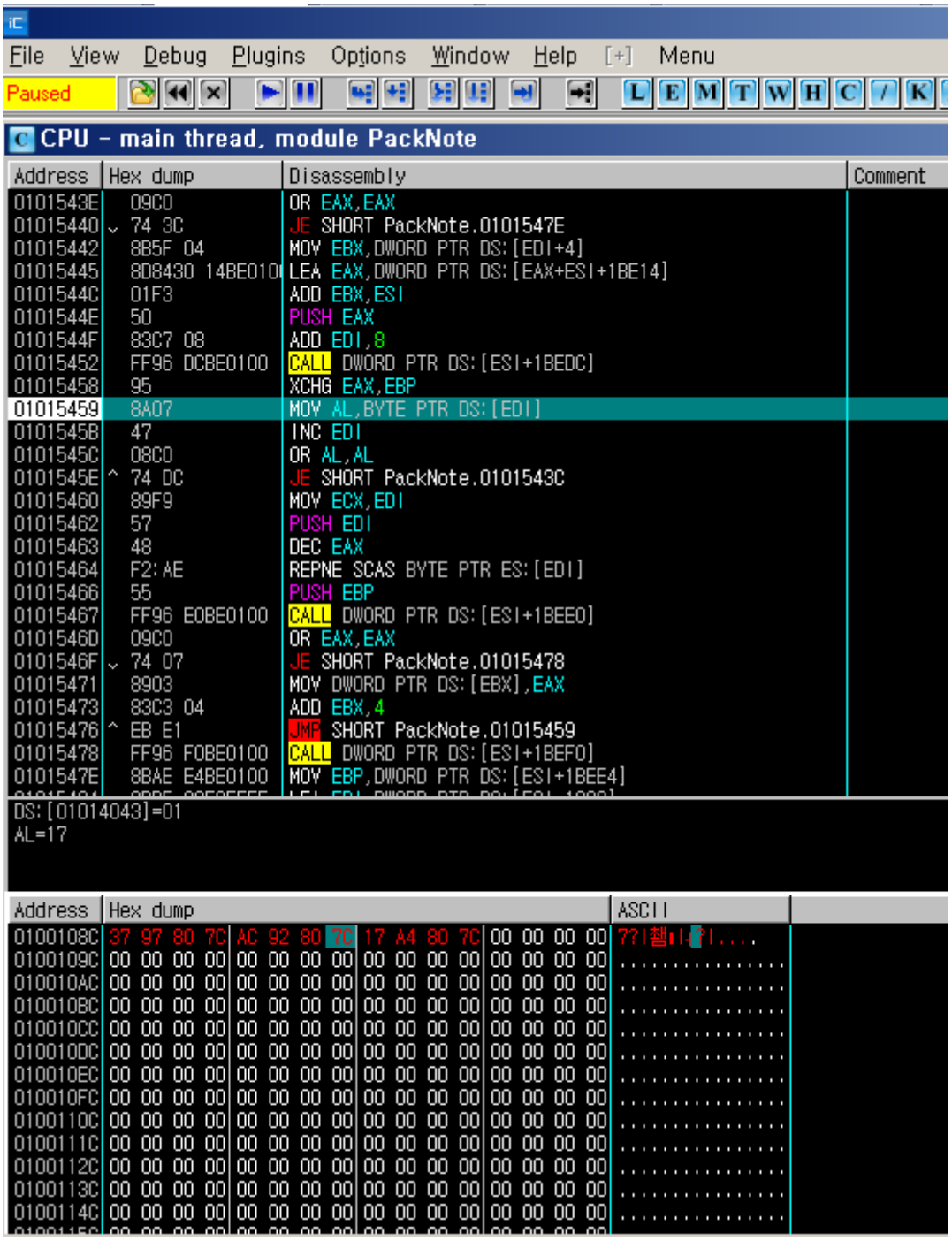
7604 – 1000 + 400 = 6A04 입니다.

```
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..
```

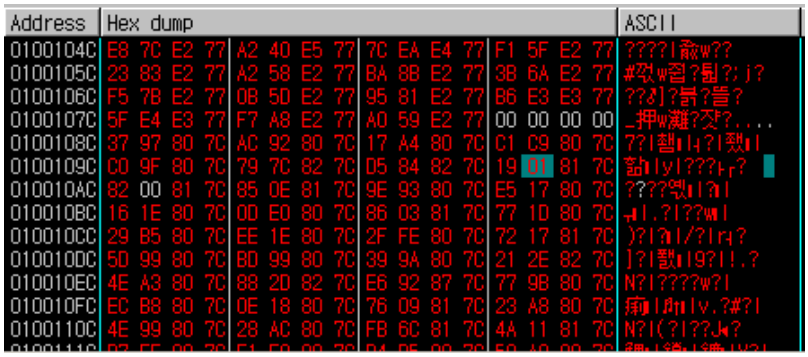
오리지널은 이처럼 존재합니다..

음..upx를 풀어보도록 하겠습니다.





여기서 IAT Table을 새로 생성해주고 있다

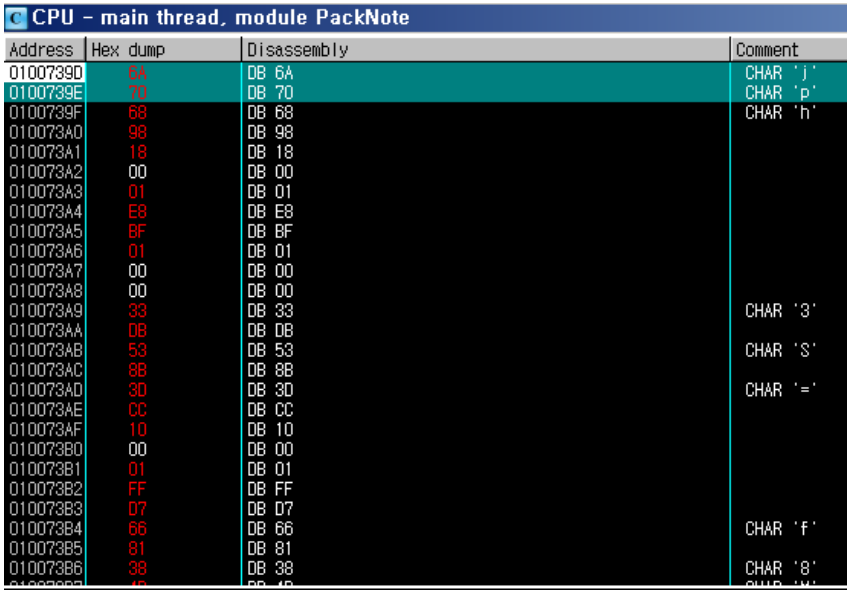


Address	Hex dump	ASCII
0101CE14	00 00 00 00 00 00 00 00 00 00 00 00 38 CF 01 00	,.....8?.
0101CE24	DC CE 01 00 00 00 00 00 00 00 00 00 00 00 00 00	裨r.....
0101CE34	45 CF 01 00 F8 CE 01 00 00 00 00 00 00 00 00 00	E?.陛r.....
0101CE44	00 00 00 00 52 CF 01 00 00 CF 01 00 00 00 00 00	....R?..?.....
0101CE54	00 00 00 00 00 00 00 00 5F CF 01 00 08 CF 01 00	........_?.?.
0101CE64	00 00 00 00 00 00 00 00 00 00 00 00 6C CF 01 00	............l?.
0101CE74	10 CF 01 00 00 00 00 00 00 00 00 00 00 00 00 00	+?.....
0101CE84	76 CF 01 00 18 CF 01 00 00 00 00 00 00 00 00 00	v?.↑?.....
0101CE94	00 00 00 00 81 CF 01 00 20 CF 01 00 00 00 00 00	....곋r. ?.....
0101CEA4	00 00 00 00 00 00 00 00 8D CF 01 00 28 CF 01 00	........뗑r.(?.
0101CEB4	00 00 00 00 00 00 00 00 00 00 00 00 98 CF 01 00	..........샫r.
0101CEC4	30 CF 01 00 00 00 00 00 00 00 00 00 00 00 00 00	0?.....
0101CED4	00 00 00 00 00 00 00 00 77 1D 80 7C 28 AC 80 7C	........w↔(?l

요건 Import Table 이고..(아까 안보이던....)

패킹이 풀리면서 나타난듯하다. (이건 추적을 못했네요...)

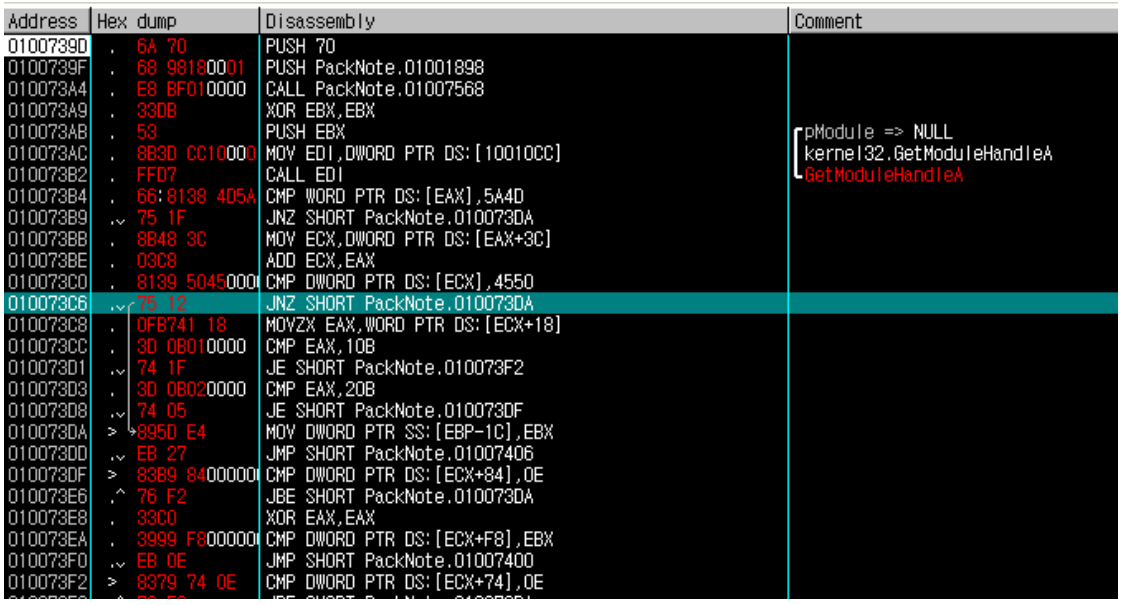
아래는 OEP



CPU - main thread, module PackNote

Address	Hex dump	Disassembly	Comment
0100739D	6A	DB 6A	CHAR 'j'
0100739E	70	DB 70	CHAR 'p'
0100739F	68	DB 68	CHAR 'h'
010073A0	98	DB 98	
010073A1	18	DB 18	
010073A2	00	DB 00	
010073A3	01	DB 01	
010073A4	E8	DB E8	
010073A5	BF	DB BF	
010073A6	01	DB 01	
010073A7	00	DB 00	
010073A8	00	DB 00	
010073A9	33	DB 33	CHAR '3'
010073AA	DB	DB DB	
010073AB	53	DB 53	CHAR 'S'
010073AC	8B	DB 8B	
010073AD	3D	DB 3D	CHAR '='
010073AE	CC	DB CC	
010073AF	10	DB 10	
010073B0	00	DB 00	
010073B1	01	DB 01	
010073B2	FF	DB FF	
010073B3	D7	DB D7	
010073B4	66	DB 66	CHAR 'f'
010073B5	81	DB 81	
010073B6	38	DB 38	CHAR '8'

OEP로 왔습니다.

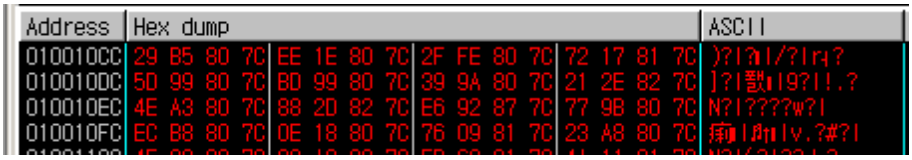


Address	Hex dump	Disassembly	Comment
0100739D	6A 70	PUSH 70	
0100739F	68 98180001	PUSH PackNote.01001898	
010073A4	E8 BF010000	CALL PackNote.01007568	
010073A9	33DB	XOR EBX,EBX	
010073AB	53	PUSH EBX	pModule => NULL
010073AC	8B3D CC100001	MOV EDI,DWORD PTR DS:[1001CC]	kernel32.GetModuleHandleA
010073B2	FFD7	CALL EDI	GetModuleHandleA
010073B4	66:8138 4D5A	CMP WORD PTR DS:[EAX],5A4D	
010073B9	75 1F	JNZ SHORT PackNote.010073DA	
010073BB	8B48 3C	MOV ECX,DWORD PTR DS:[EAX+3C]	
010073BE	03C8	ADD ECX,EAX	
010073C0	8139 50450000	CMP DWORD PTR DS:[ECX],4550	
010073C6	75 12	JNZ SHORT PackNote.010073DA	
010073C8	0FB741 18	MOVZX EAX,WORD PTR DS:[ECX+18]	
010073CC	3D 0B010000	CMP EAX,10B	
010073D1	74 1F	JE SHORT PackNote.010073F2	
010073D3	3D 0B020000	CMP EAX,20B	
010073D8	74 05	JE SHORT PackNote.010073DF	
010073DA	895D E4	MOV DWORD PTR SS:[EBP-1C],EBX	
010073DD	EB 27	JMP SHORT PackNote.01007406	
010073DF	83B9 84000000	CMP DWORD PTR DS:[ECX+84],0E	
010073E6	76 F2	JBE SHORT PackNote.010073DA	
010073E8	33C0	XOR EAX,EAX	
010073EA	3999 F8000000	CMP DWORD PTR DS:[ECX+F8],EBX	
010073F0	EB 0E	JMP SHORT PackNote.01007400	
010073F2	8379 74 0E	CMP DWORD PTR DS:[ECX+74],0E	

위에 보니 GetModuleHandleA함수가 보이는군요..

아까 복원한 IAT Table을 가르키죠?

따라가보도록 하겠습니다.



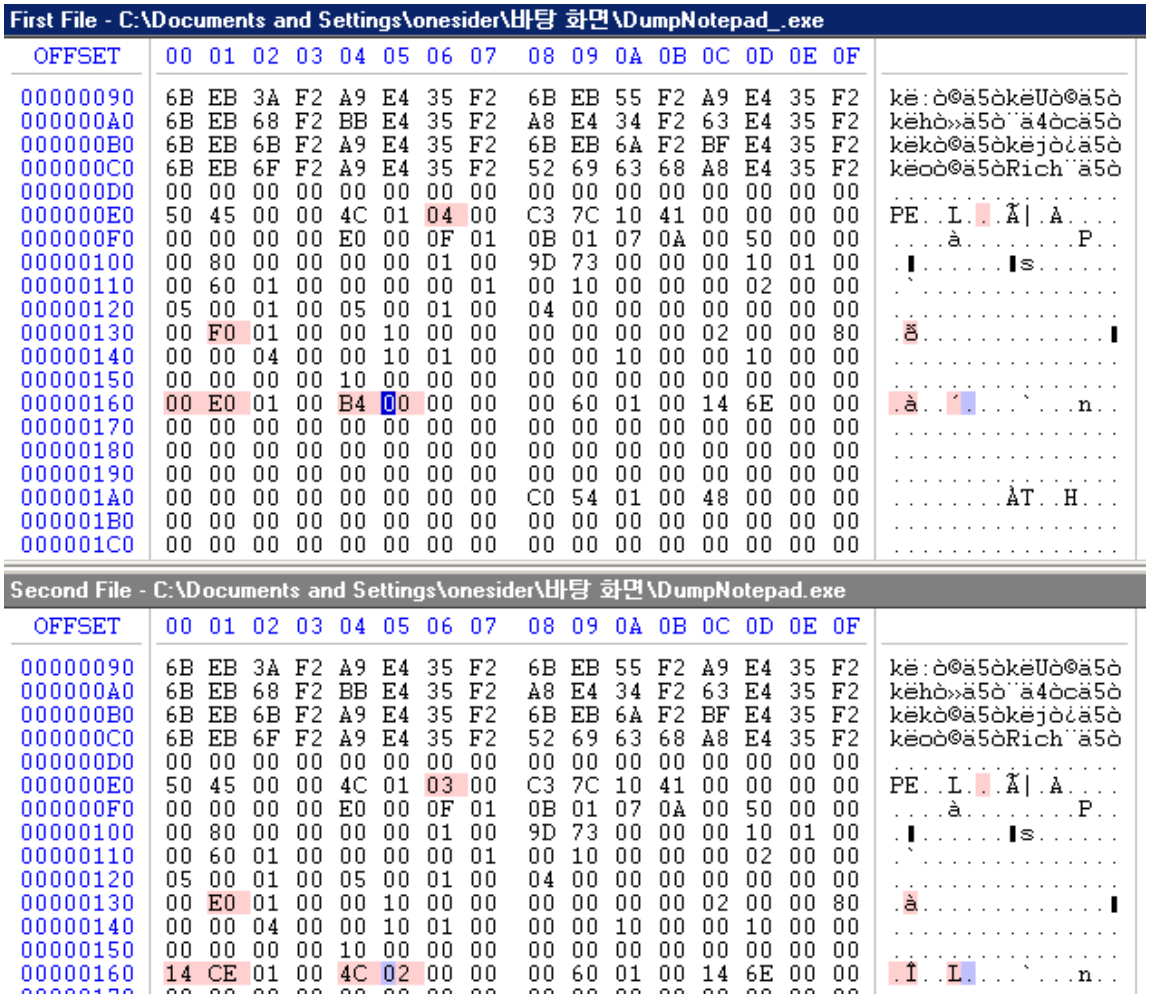
예상대로군요...그럼 7C80B529 로 가볼까요?

음..예상대로입니다.

그럼 이제 원래 목적대로 Import ReConstructor 의 역할을 알아보도록하죠..

일단 OEP에서 덤프를 뜨도록 하겠습니다.

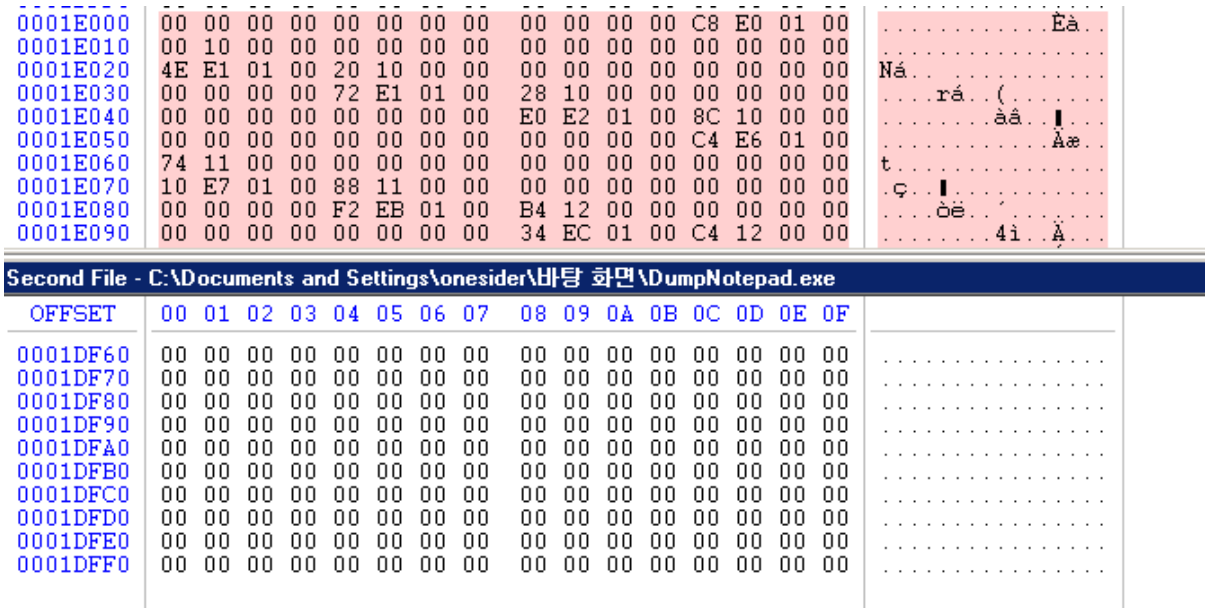
저는 OEP가 739D 군요.. 그 후 Import ReConstructor로 리빌딩 하겠습니다.



좀 수정 되었는데요.. 일단.. 아래쪽이 그냥 덤프 파일입니다. 위쪽은 덤프 뜬 것을 리빌딩 했습니다.

NumberOfSection 3 -> 4
SizeOfImage 01E000 -> 01F000
ImportTableRva 01CE14 -> 01E000
ImportTableSize 024C -> 00b4

섹션 추가 0x250
Name : .mackt -> 2E 6D 61 63 6B 74
VirtualSize : 0x1000
RVA : 1E000
SizeOfRawData : 1000
PointerToRawData : 1E000



```
0001E000  00 00 00 00 00 00 00 00  00 00 00 00 C8 E0 01 00  ............Èà..
0001E010  00 10 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0001E020  4E E1 01 00 20 10 00 00  00 00 00 00 00 00 00 00  Ná.. ...........
0001E030  00 00 00 00 72 E1 01 00  28 10 00 00 00 00 00 00  ....rá..( ......
0001E040  00 00 00 00 00 00 00 00  E0 E2 01 00 8C 10 00 00  ........àâ..Œ...
0001E050  00 00 00 00 00 00 00 00  00 00 00 00 C4 E6 01 00  ............Äæ..
0001E060  74 11 00 00 00 00 00 00  00 00 00 00 00 00 00 00  t...............
0001E070  10 E7 01 00 88 11 00 00  00 00 00 00 00 00 00 00  .ç..ˆ...........
0001E080  00 00 00 00 F2 EB 01 00  B4 12 00 00 00 00 00 00  ....òë..´.......
0001E090  00 00 00 00 00 00 00 00  34 EC 01 00 C4 12 00 00  ........4ì..Ä...
```

Second File - C:\Documents and Settings\onesider\바탕 화면\DumpNotepad.exe

```
OFFSET    00 01 02 03 04 05 06 07  08 09 0A 0B 0C 0D 0E 0F
0001DF60  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0001DF70  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0001DF80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0001DF90  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0001DFA0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0001DFB0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0001DFC0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0001DFD0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0001DFE0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0001DFF0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

덤프에는 없지만 언팩한 파일에는 뒤에 섹션이 추가되어 있음
이곳에 IAT Table이 복원되어 있음.

결론 : 툴을 안쓰고 손으로 헥사값 고쳐서 리빌딩은 아까 만들어진 섹션을 밑에 붙이고 헤더값을 수정하면 될듯하지만 그냥 보기에도 귀찮은 작업으로 보이는군요..