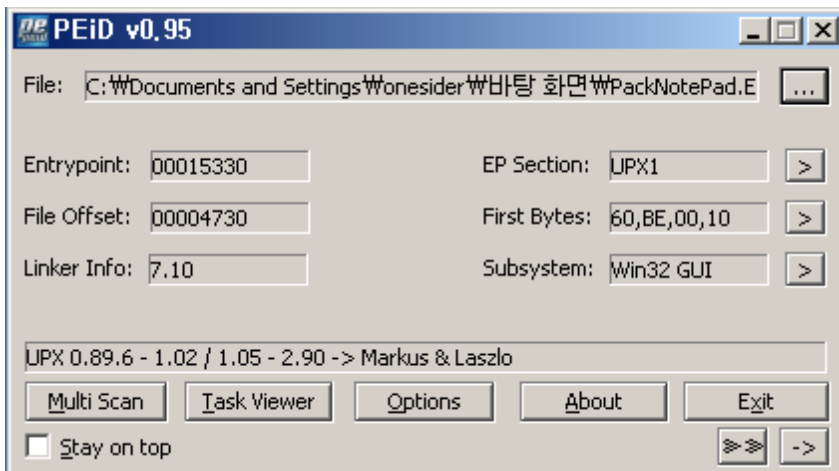


실행압축과 IAT Table 추적

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.

SECTION HEADERS							
Name	Virtual Size	Virtual Address	Size of Raw Data	Pointer to Raw Data	Characteristics	Pointing Directories	
UP...	00010000h	01001000h	00000000h	00000400h	E0000080h		
UP...	00005000h	01011000h	00004600h	00000400h	E0000040h	Load Configuration Table	
.rsrc	00008000h	01016000h	00007200h	00004A00h	C0000040h	Import Table; Resource Table	

섹션은 3개 (두 번째 섹션에 Import Table이 포함되어 있음)

SectionAlignment = 0x1000

FileAlignment = 0x200

ImageBase = 0x01000000

Address Of Entry Point = 0x01015330(ImageBase 를 더해준 값)

DATA DIRECTORIES		
Export Table	00000000	00000000
Directory Name	Virtual Address	Size
Export Table		
Import Table	0101CE14h	0000024Ch
Resource Table	01016000h	00006E14h
Exception Table		
Certificate Table		
Relocation Table		
Debug Data		
Architecture-specific data		
Machine Value (MIPS GP)		
TLS Table		
Load Configuration Table	010154C0h	00000048h
Bound Import Table		
Import Address Table		
Delay Import Descriptor		
COM+ Runtime Header		
(15) Reserved		

임포트 테이블은 0101CE14를 가르키고 있다.
(Pe Explorer는 희안하게 ImageBase값을 알아서 더해준다..)

ImageBase가 없는 원래의 값은 1CE14 이므로
 $400 + (1CE14 - Rva(11000)) = \text{“디스크상의 Import Table위치”}$ 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.

DATA DIRECTORIES		
Export Table	00000000	00000000
Directory Name	Virtual Address	Size
Export Table		
Import Table	01007604h	000000C8h
Resource Table	0100B000h	00008314h
Exception Table		
Certificate Table		
Relocation Table		
Debug Data	01001350h	0000001Ch
Architecture-specific data		
Machine Value (MIPS GP)		
TLS Table		
Load Configuration Table	010018A8h	00000040h
Bound Import Table	01000250h	000000D0h
Import Address Table	01001000h	00000348h
Delay Import Descriptor		
COM+ Runtime Header		
(15) Reserved		

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	..iiy..yyyyyyyy
00006A10	AC 7A 00 00 C4 12 00 00	40 78 00 00 FF FF FF FF	~z..A...@x..yyyy
00006A20	FF FF FF FF FA 7A 00 00	74 11 00 00 80 79 00 00	yyyyuz..t...ly..
00006A30	FF FF FF FF FF FF FF FF	3A 7B 00 00 B4 12 00 00	yyyyyyyyy{...^...
00006A40	EC 76 00 00 FF FF FF FF	FF FF FF FF 5E 7B 00 00	iv..yyyyyyyyy^{...
00006A50	20 10 00 00 B8 79 00 00	FF FF FF FF FF FF FF FF	...y..yyyyyyyyy
00006A60	76 7C 00 00 EC 12 00 00	CC 76 00 00 FF FF FF FF	v ..i...Iv..yyyy
00006A70	FF FF FF FF 08 7D 00 00	00 10 00 00 58 77 00 00	yyyy..}.....Xw..

오리지널은 이처럼 존재합니다..

음..upx를 풀어보도록 하겠습니다.

The screenshot shows a debugger window with the following content:

CPU - main thread, module PackNote

Address	Hex dump	Disassembly	Comment
010153E6	E9 63FFFFFF	JMP PackNote.0101534E	
010153EB	90	NOP	
010153EC	8B02	MOV EAX,DWORD PTR DS:[EDX]	
010153EE	83C2 04	ADD EDX,4	
010153F1	8907	MOV DWORD PTR DS:[EDI],EAX	
010153F3	83C7 04	ADD EDI,4	
010153F6	83E9 04	SUB ECX,4	
010153F9	77 F1	JA SHORT PackNote.010153EC	
010153FB	01CF	ADD EDI,ECX	
010153FD	E9 4CFFFFFF	JMP PackNote.0101534E	
01015402	5E	POP ESI	PackNote.0
01015403	89F7	MOV EDI,ESI	
01015405	B9 32010000	MOV ECX,132	
0101540A	8A07	MOV AL,BYTE PTR DS:[EDI]	
0101540C	47	INC EDI	
0101540D	2C E8	SUB AL,0E8	
0101540F	3C 01	CMP AL,1	
01015411	77 F7	JA SHORT PackNote.0101540A	
01015413	803F 01	CMP BYTE PTR DS:[EDI],1	
01015416	75 F2	JNZ SHORT PackNote.0101540A	
01015418	8B07	MOV EAX,DWORD PTR DS:[EDI]	
0101541A	8A5F 04	MOV BL,BYTE PTR DS:[EDI+4]	
0101541D	66 C1E8 08	SHR AX,8	
01015421	C1C0 10	ROL EAX,10	
01015424	86C4	XCHG AH,AL	
01015426	29F8	SUB EAX,EDI	
01015428	89FB 00	MOV EBX,0	

Stack [0006FFA0]=01001000 (PackNote.01001000)
ESI=01015330 (PackNote.<ModuleEntryPoint>)

Address	Hex dump	ASCII
01001480	61 00 72 00 67 00 69 00 6E 00 52 00 69 00 67 00	a.r.g.i.n.R.i.g.
01001490	68 00 74 00 00 00 00 00 69 00 40 00 61 00 72 00	h.t.....i.M.a.r.
010014A0	67 00 69 00 6E 00 4C 00 65 00 66 00 74 00 00 00	g.i.n.L.e.f.t...
010014B0	69 00 40 00 61 00 72 00 67 00 69 00 6E 00 42 00	i.M.a.r.g.i.n.B.
010014C0	6F 00 74 00 74 00 6F 00 6D 00 00 00 69 00 40 00	o.t.t.o.m...i.M.
010014D0	61 00 72 00 67 00 69 00 6E 00 54 00 6F 00 70 00	a.r.g.i.n.T.o.p.
010014E0	00 00 00 00 73 00 7A 00 54 00 72 00 61 00 69 00	...s.z.T.r.a.i.
010014F0	6C 00 65 00 72 00 00 00 73 00 7A 00 48 00 65 00	l.e.r...s.z.H.e.
01001500	61 00 64 00 65 00 72 00 00 00 00 00 6C 00 66 00	a.d.e.r.....l.f.
01001510	46 00 61 00 63 00 65 00 4E 00 61 00 6D 00 65 00	F.a.c.e.N.a.m.e.
01001520	00 00 00 00 66 00 53 00 61 00 76 00 65 00 57 00	...f.S.a.v.e.W.
01001530	69 00 6E 00 64 00 6F 00 77 00 50 00 6F 00 73 00	i.n.d.o.w.P.o.s.
01001540	69 00 74 00 69 00 6F 00 6E 00 73 00 00 00 00 00	i.t.i.o.n.s.....

Address	Hex dump	ASCII
0101CE14	00 00 00 00 00 00 00 00 00 00 00 00 38 CF 01 008?.
0101CE24	DC CE 01 00 00 00 00 00 00 00 00 00 00 00 00 00	禿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!?.
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	...켈r.?.
0101CEA4	00 00 00 00 00 00 00 00 80 CF 01 00 28 CF 01 00뵁r.(?
0101CEB4	00 00 00 00 00 00 00 00 00 00 00 00 98 CF 01 00샐r.
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 10 80 7C 28 AC 80 7C뵁!(?

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	DA	DB 6A	CHAR 'j'
0100739E	73	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	0B	DB 0B	
010073AB	53	DB 53	CHAR 'S'
010073AC	98	DB 98	
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	07	DB 07	
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	330B	XOR EBX,EBX	
010073AB	53	PUSH EBX	
010073AC	8B3D CC1000	MOV EDI,DWORD PTR DS:[10010CC]	pModule => NULL
010073B2	FFD7	CALL EDI	kernel32.GetModuleHandleA
010073B4	66: 8138 4D5A	CMP WORD PTR DS:[EAX],5A4D	GetModuleHandleA
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 GetModuleHandleA 함수가 보이는군요..

```

010073AB 53          PUSH EBX
010073AC 8B3D CC1000 MOV EDI,DWORD PTR DS:[10010CC]
010073B2 FF07       CALL EDI

```

pModule => NULL
kernel32.GetModuleHandleA
GetModuleHandleA

아까 복원한 IAT Table을 가르키죠?

따라가보도록 하겠습니다.

Address	Hex dump	ASCII
010010CC	29 B5 80 7C EE 1E 80 7C 2F FE 80 7C 72 17 81 7C)?i?/?i?
010010DC	5D 99 80 7C 8D 99 80 7C 39 9A 80 7C 21 2E 82 7C]?i?i9?!!?
010010EC	4E A3 80 7C 88 2D 82 7C E6 92 87 7C 77 98 80 7C	N?i????w?
010010FC	EC B8 80 7C 0E 18 80 7C 76 09 81 7C 23 A8 80 7C	痲i?nly.?#?

예상대로군요...그럼 7C80B529 로 가볼까요?

Address	Hex dump	Disassembly	Comment
7C80B529	8BFF	MOV EDI,EDI	kernel32.GetModuleHandleA

음..예상대로입니다.

그럼 이제 원래 목적대로 Import ReConstructor 의 역할을 알아보도록하죠..

일단 OEP에서 덤프를 뜨도록 하겠습니다.

저는 OEP가 739D군요.. 그 후 Import ReConstructor로 리빌딩 하겠습니다.

First 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	
00000090	6B	EB	3A	F2	A9	E4	35	F2	6B	EB	55	F2	A9	E4	35	F2	kë:ò@ä5òkëUò@ä5ò
000000A0	6B	EB	68	F2	BB	E4	35	F2	A8	E4	34	F2	63	E4	35	F2	këhò»ä5ò"ä4òcä5ò
000000B0	6B	EB	6B	F2	A9	E4	35	F2	6B	EB	6A	F2	BF	E4	35	F2	këkò@ä5òkëjòçä5ò
000000C0	6B	EB	6F	F2	A9	E4	35	F2	52	69	63	68	A8	E4	35	F2	këoò@ä5òRich"ä5ò
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000E0	50	45	00	00	4C	01	04	00	C3	7C	10	41	00	00	00	00	PE..I..Ä .A...
000000F0	00	00	00	00	E0	00	0F	01	0B	01	07	0A	00	50	00	00	..à.....P..
00000100	00	80	00	00	00	00	01	00	9D	73	00	00	00	10	01	00 s.....
00000110	00	60	01	00	00	00	00	01	00	10	00	00	00	02	00	00
00000120	05	00	01	00	05	00	01	00	04	00	00	00	00	00	00	00
00000130	00	F0	01	00	00	10	00	00	00	00	00	00	02	00	00	80	..ä.....
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	00	E0	01	00	B4	00	00	00	00	60	01	00	14	6E	00	00	..ä.. .n..
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
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	C0	54	01	00	48	00	00	00ÄT..H...
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

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	
00000090	6B	EB	3A	F2	A9	E4	35	F2	6B	EB	55	F2	A9	E4	35	F2	kë:ò@ä5òkëUò@ä5ò
000000A0	6B	EB	68	F2	BB	E4	35	F2	A8	E4	34	F2	63	E4	35	F2	këhò»ä5ò"ä4òcä5ò
000000B0	6B	EB	6B	F2	A9	E4	35	F2	6B	EB	6A	F2	BF	E4	35	F2	këkò@ä5òkëjòçä5ò
000000C0	6B	EB	6F	F2	A9	E4	35	F2	52	69	63	68	A8	E4	35	F2	këoò@ä5òRich"ä5ò
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000E0	50	45	00	00	4C	01	03	00	C3	7C	10	41	00	00	00	00	PE..I..Ä .A...
000000F0	00	00	00	00	E0	00	0F	01	0B	01	07	0A	00	50	00	00	..à.....P..
00000100	00	80	00	00	00	00	01	00	9D	73	00	00	00	10	01	00 s.....
00000110	00	60	01	00	00	00	00	01	00	10	00	00	00	02	00	00
00000120	05	00	01	00	05	00	01	00	04	00	00	00	00	00	00	00
00000130	00	E0	01	00	00	10	00	00	00	00	00	00	02	00	00	80	..ä.....
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	14	CE	01	00	4C	02	00	00	00	60	01	00	14	6E	00	00	..I..n..

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 00Èà..
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 00rá..(.....
0001E040	00 00 00 00 00 00 00 00 E0 E2 01 00 8C 10 00 00àà..!.....
0001E050	00 00 00 00 00 00 00 00 00 00 00 00 C4 E6 01 00Åæ..
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..!.....
0001E080	00 00 00 00 F2 EB 01 00 B4 12 00 00 00 00 00 00òè.....
0001E090	00 00 00 00 00 00 00 00 34 EC 01 00 C4 12 00 004i..Å.....

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 : 틀을 안쓰고 손으로 hexa값 고쳐서 리빌딩은 아까 만들어진 섹션을
 밑에 붙이고 헤더값을 수정하면 될듯하지만 그냥 보기에다 귀찮은 작업으로
 보이는군요..