BASIC RCE Level 2

CodeEngn
ReverseEngineering Conference

2013 07/10
Malcook90@naver.com

Challenges : Basic 02

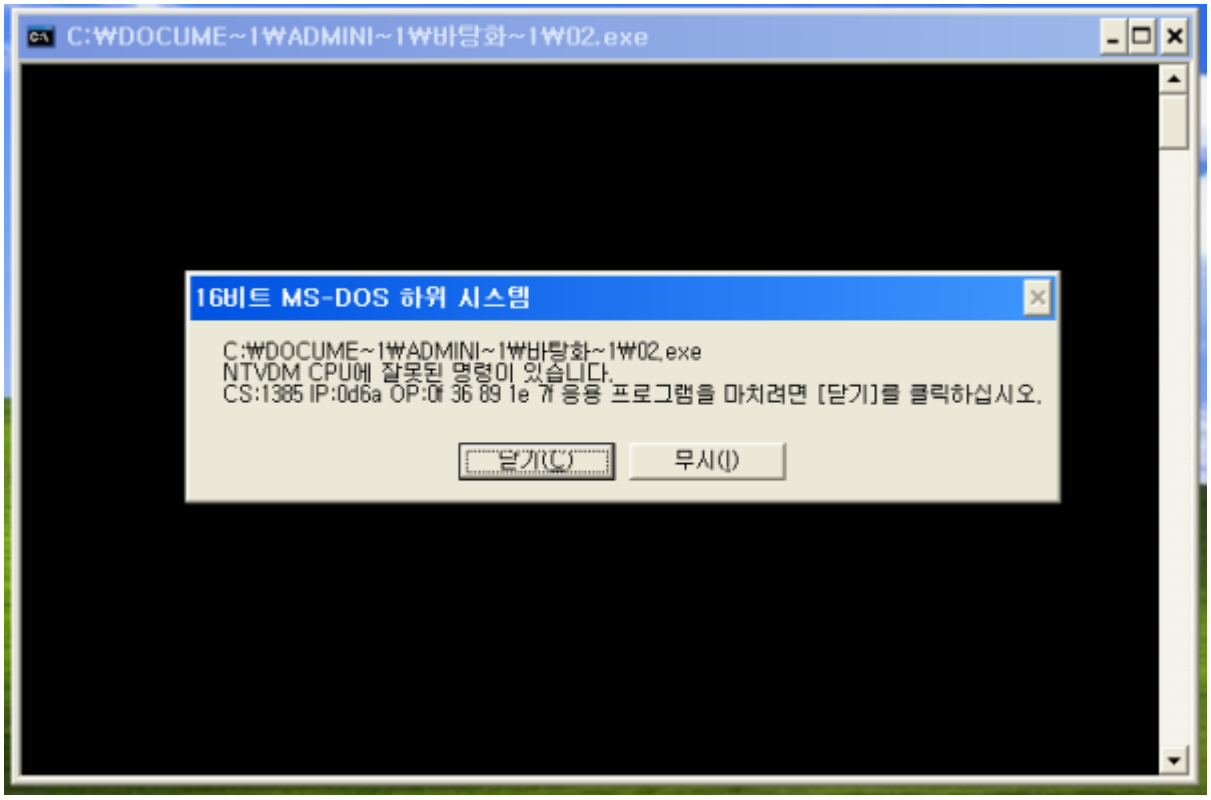
Author : ArturDents

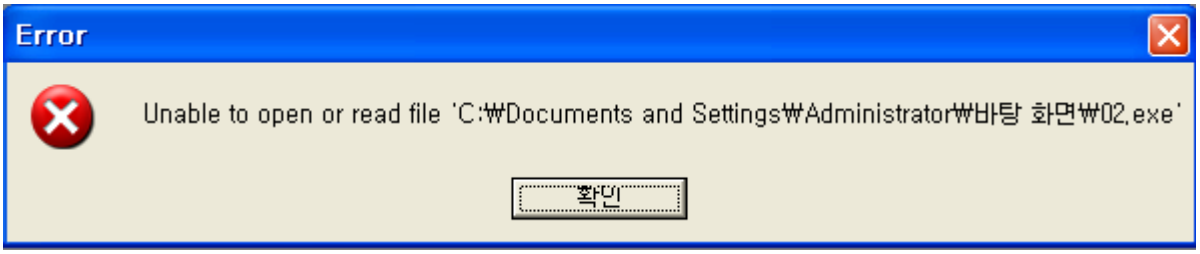
Korea :
패스워드로 인증하는 실행파일이 손상되어 실행이 안되는 문제가 생겼다. 패스워드가 무엇인지 분석하시오

English :
The program that verifies the password got messed up and ceases to execute. Find out what the password is.

손상 되었다는 말로 PE파일과 관련 있는 문제라는 것을 유추해 볼 수 있다.

우선 실행시켜 보도록 하자





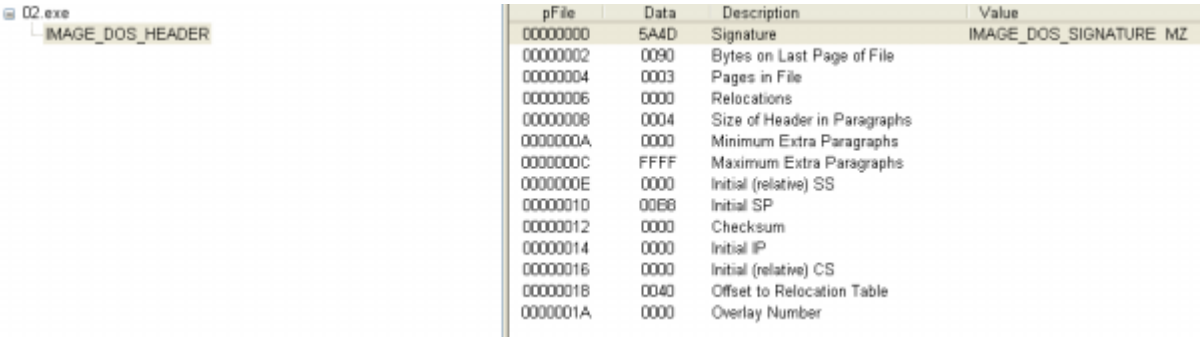
1) .exe 파일로 실행한 경우

2) OllyDBG 실행 했을 경우

둘 다 오류가 뜨는 것을 확인할 수 있다.

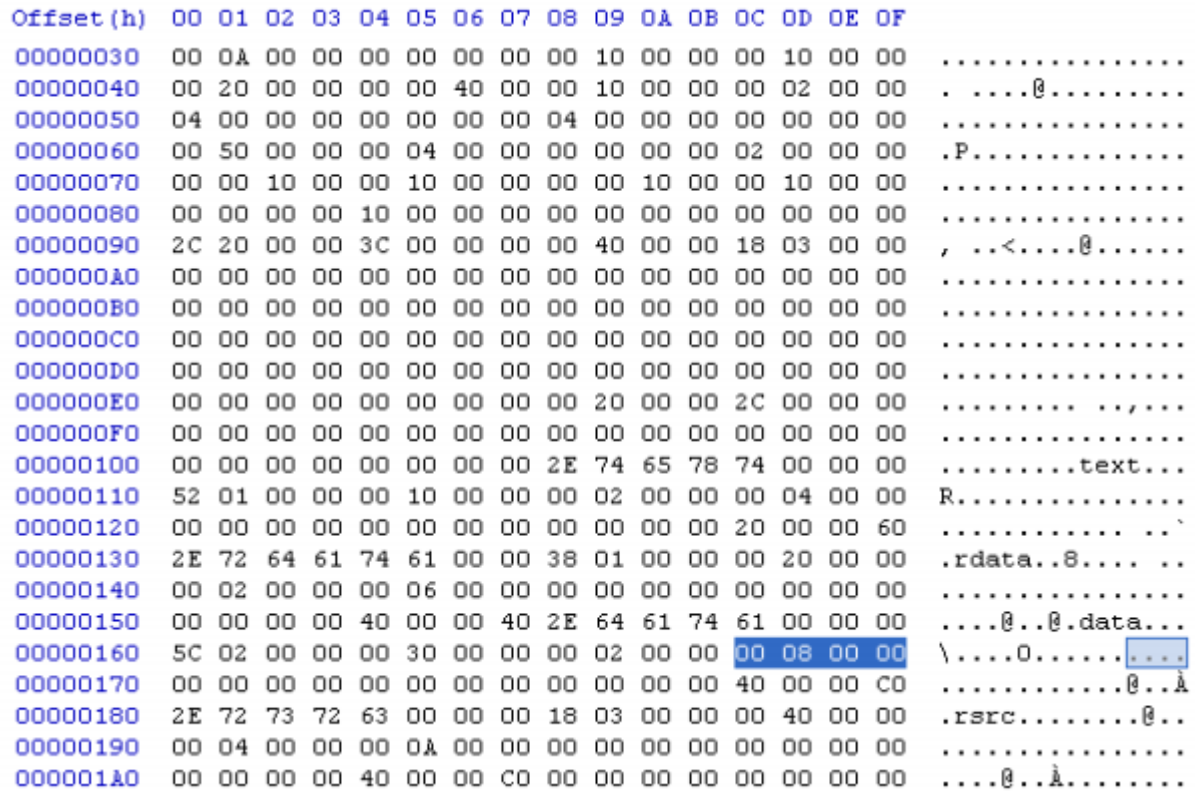
예상대로 PE파일과 관련된 문제임을 알 수 있다.

HxD 로 열어보기 전.. PEview 로 먼저 체크해 보도록 하자



상단의 IMAGE_DOS_HEADER 밖에 나오질 않는다.

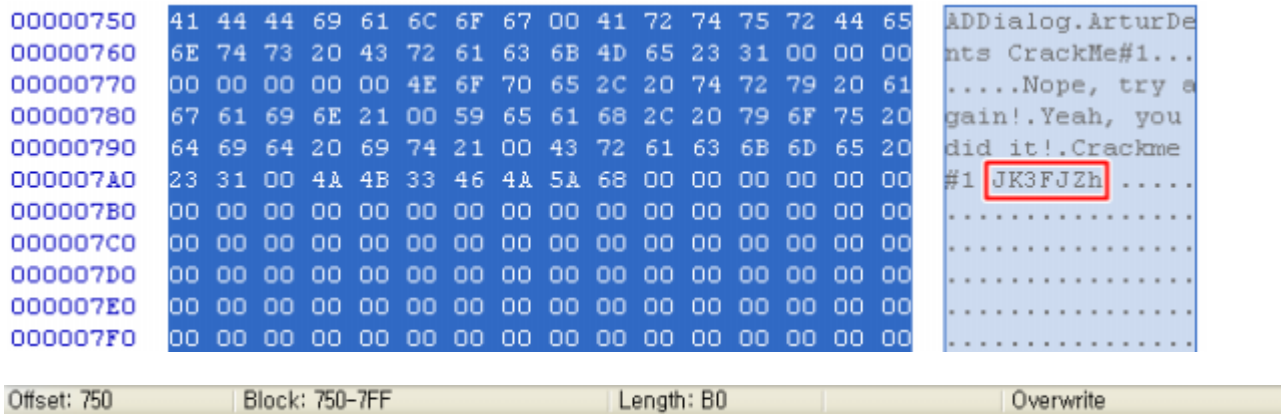
이로써 밑부분에 있는 IMAGE_NT_HEADER 에 문제가 있는 것을 알 수 있다.



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000030	00	0A	00	00	00	00	00	00	00	10	00	00	00	10	00	00	.....
00000040	00	20	00	00	00	00	40	00	00	10	00	00	00	02	00	00	. ....@.....
00000050	04	00	00	00	00	00	00	00	04	00	00	00	00	00	00	00	.....
00000060	00	50	00	00	00	04	00	00	00	00	00	00	02	00	00	00	.P.....
00000070	00	00	10	00	00	10	00	00	00	00	10	00	00	10	00	00	.....
00000080	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00	.....
00000090	2C	20	00	00	3C	00	00	00	00	40	00	00	18	03	00	00	, ..<....@.....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000E0	00	00	00	00	00	00	00	00	00	20	00	00	2C	00	00	00	....,...
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000100	00	00	00	00	00	00	00	00	2E	74	65	78	74	00	00	00	.....text...
00000110	52	01	00	00	00	10	00	00	00	02	00	00	00	04	00	00	R.....
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	20	00	00	60	....`
00000130	2E	72	64	61	74	61	00	00	38	01	00	00	00	20	00	00	.rdata..8.... ..
00000140	00	02	00	00	00	06	00	00	00	00	00	00	00	00	00	00	.....
00000150	00	00	00	00	40	00	00	40	2E	64	61	74	61	00	00	00	....@..@.data...
00000160	5C	02	00	00	00	30	00	00	00	02	00	00	00	08	00	00	\....0.....
00000170	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0	.....@..À
00000180	2E	72	73	72	63	00	00	00	18	03	00	00	00	40	00	00	.rsrc.....@..
00000190	00	04	00	00	00	0A	00	00	00	00	00	00	00	00	00	00	.....
000001A0	00	00	00	00	40	00	00	C0	00	00	00	00	00	00	00	00	....@..À.....

Section 에는 .text, .rdata, .data, .rsrc 4개로 구성되어 있는 것을 알수있다.

패스워드가 무엇인지 분석하라 라는 말로 보아 .data 에 패스워드가 저장되어 있을 가능성이 있다. 한번 이동해 보도록 하자.



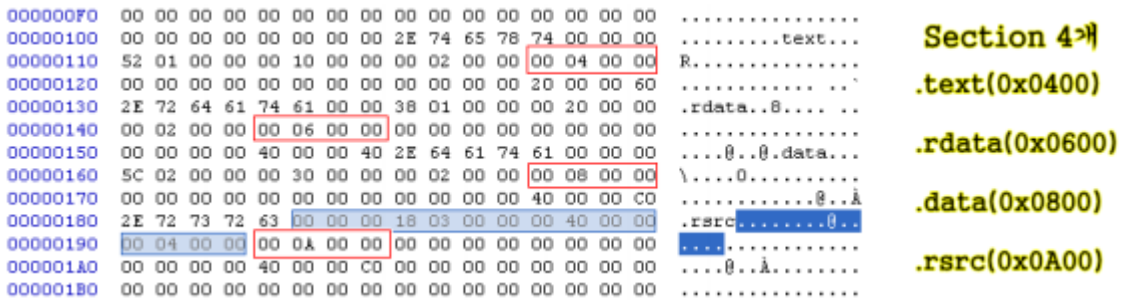
00000750	41	44	44	69	61	6C	6F	67	00	41	72	74	75	72	44	65	ADDialog.ArturDe
00000760	6E	74	73	20	43	72	61	63	6B	4D	65	23	31	00	00	00	nts CrackMe#1...
00000770	00	00	00	00	00	4E	6F	70	65	2C	20	74	72	79	20	61	.....Nope, try a
00000780	67	61	69	6E	21	00	59	65	61	68	2C	20	79	6F	75	20	gain!.Yeah, you
00000790	64	69	64	20	69	74	21	00	43	72	61	63	6B	6D	65	20	did it!.Crackme
000007A0	23	31	00	4A	4B	33	46	4A	5A	68	00	00	00	00	00	00	#1 JK3FJZh .....
000007B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000007C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000007D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000007E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000007F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Offset: 750 | Block: 750-7FF | Length: B0 | Overwrite

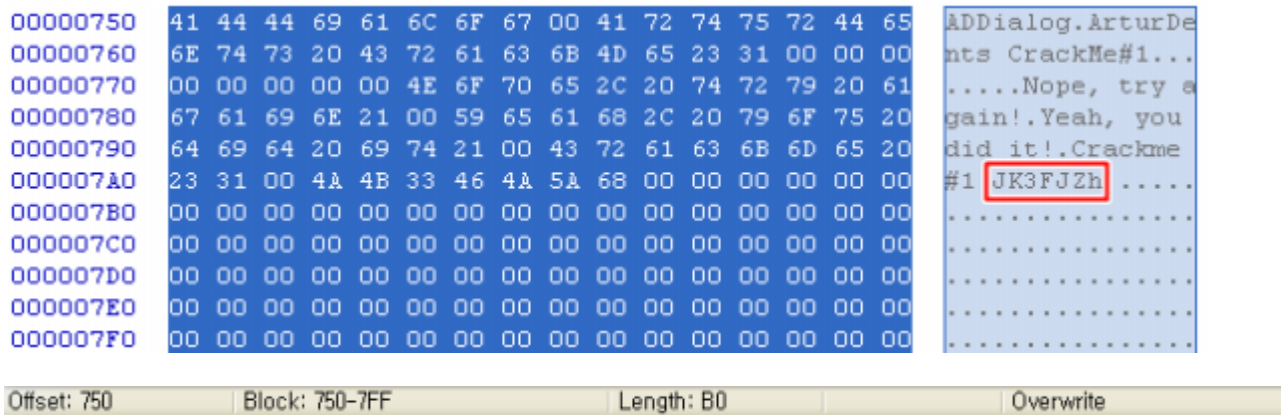
NT_HEADER 가 손상된 점을 감안할 때 0x0800 에서 위로 조금 올라가다 보면 0x0750부터 시작하는 것을 볼 수 있다.

예상한 대로 'JK3FJZh' 라는 패스워드가 보인다.

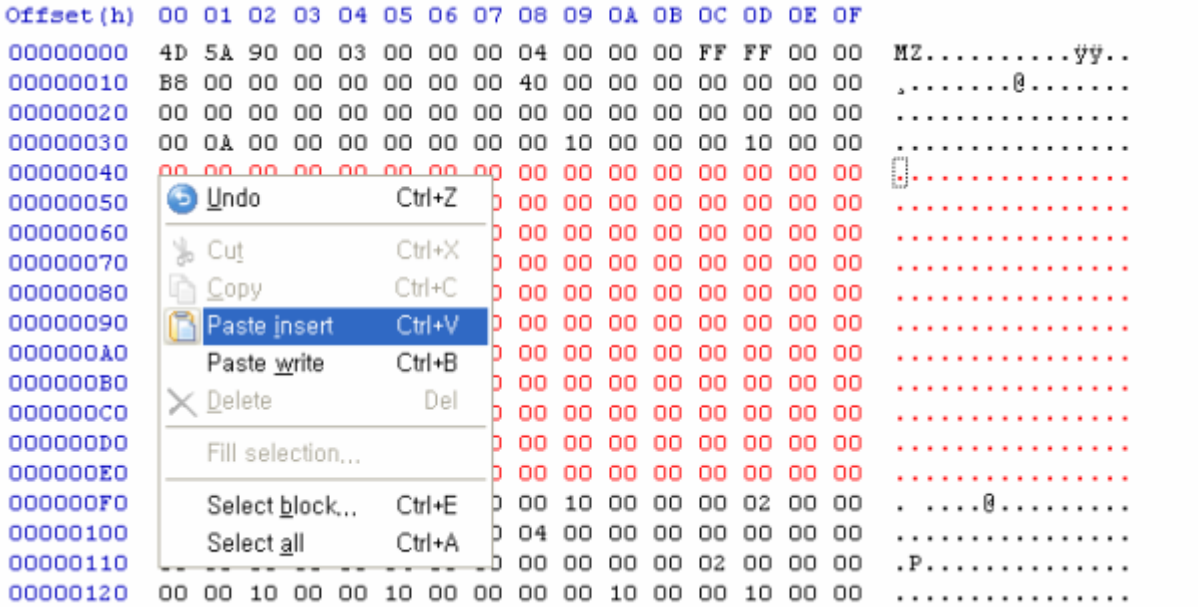
※ [+] File Recovery



각 Section은 현재 IMAGE_NT_HEADER 가 손상되어 현 위치보다 조금 위에 존재합니다.



그 위치는 0xB0 만큼 떨어져 있는 것을 확인할 수 있습니다.



0xB0 만큼의 크기을 DOS_HEADER 와 SECTION_HEADER 사이에 삽입한다.

이곳에 DOS Stub Code, FILE_HEADER, OPTIONAL_HEADER 가 들어간다.

- OPTIONAL_HEADER 의 DATA_DIRECTORY 존재 하므로 그 외 것들 수정

```
typedef struct _IMAGE_OPTIONAL_HEADER {
    //
    // Standard fields.
    //

    WORD    Magic;
    BYTE    MajorLinkerVersion;
    BYTE    MinorLinkerVersion;
    DWORD   SizeOfCode;
    DWORD   SizeOfInitializedData;
    DWORD   SizeOfUninitializedData;
    DWORD   AddressOfEntryPoint;
    DWORD   BaseOfCode;
    DWORD   BaseOfData;
```

<FILE_HEADER>

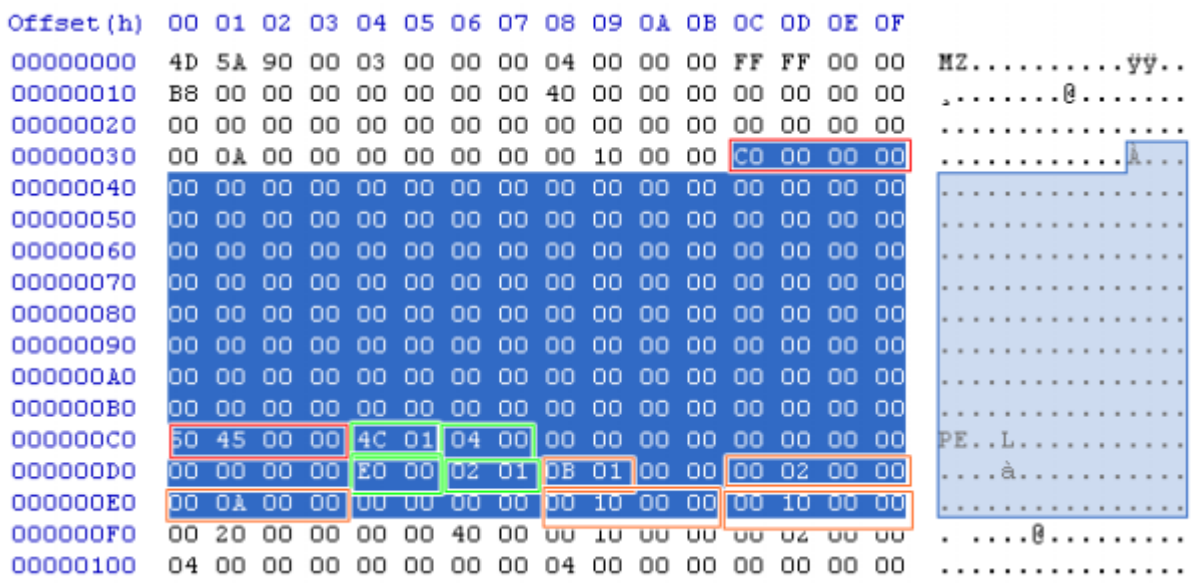
Machine : Intel 32bit 이므로 **0x014C**
NumberOfSections : **0x0004(.text .rdata .data .rsrc)**
TimeDateStamp / PointerToSymbolTable / NumberOfSymbols – **0x00000000**
SizeOfOptionalHeader : 32bit 경우 기본값으로 **0xE0** 사용
Characteristics : 파일의 속성을 나타내는 값으로 여기선 **0x0102** 로 표현
0x0100 : 32bit word Machine

```
typedef struct _IMAGE_FILE_HEADER {
    WORD    Machine;
    WORD    NumberOfSections;
    DWORD   TimeDateStamp;
    DWORD   PointerToSymbolTable;
    DWORD   NumberOfSymbols;
    WORD    SizeOfOptionalHeader;
    WORD    Characteristics;
} IMAGE_FILE_HEADER, *PIMAGE_FILE_HEADER;
```

<OPTIONAL_HEADER>

Magic : 32bit 이므로 **0x010B**
Major / Minor – **0x00**
SizeOfCode : text Section 의 크기**(0x00000200)**
SizeOfInitializedDate : Section 의 시작부터 끝**(0x00000A00)**
SizeOfUnitializedDate : 일반적으로 0**(0x00000000)**
AddressOfEntryPoint : 최초로 실행되는 코드시작 주소**(0x00001000)**
BaseOfCode : Code 의 시작주소**(0x00001000)**

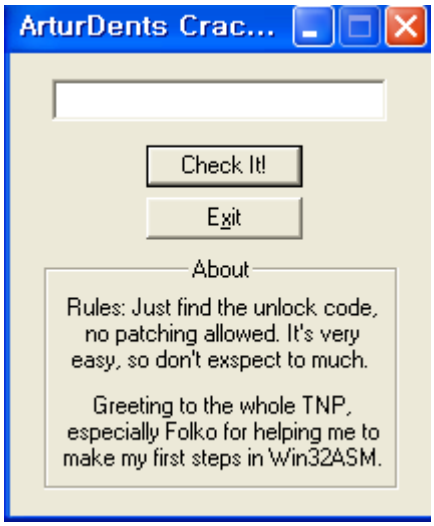
마지막으로 PE Signature 와 DOS_HEADER 의 e_lfanew 값을 수정해 주면 끝!



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	0A	00	00	00	00	00	00	00	10	00	00	C0	00	00	00
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000C0	50	45	00	00	4C	01	04	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	E0	00	02	01	0B	01	00	00	00	02	00	00
000000E0	00	0A	00	00	00	00	00	00	00	10	00	00	00	10	00	00
000000F0	00	20	00	00	00	00	40	00	00	10	00	00	00	02	00	00
00000100	04	00	00	00	00	00	00	00	04	00	00	00	00	00	00	00

ANSI
MZ.....ÿÿ..
¸.....@.....
.....
.....À...
.....
.....
.....
.....
.....
.....
.....
.....
.....
PE..L.....
....à.....
.....
. ....@.....
.....

저장 후 실행하면 복구 된 것을 확인할 수 있다.



참고자료 : codeengn explanation.PDF by Deok9
리버싱:핵심원리