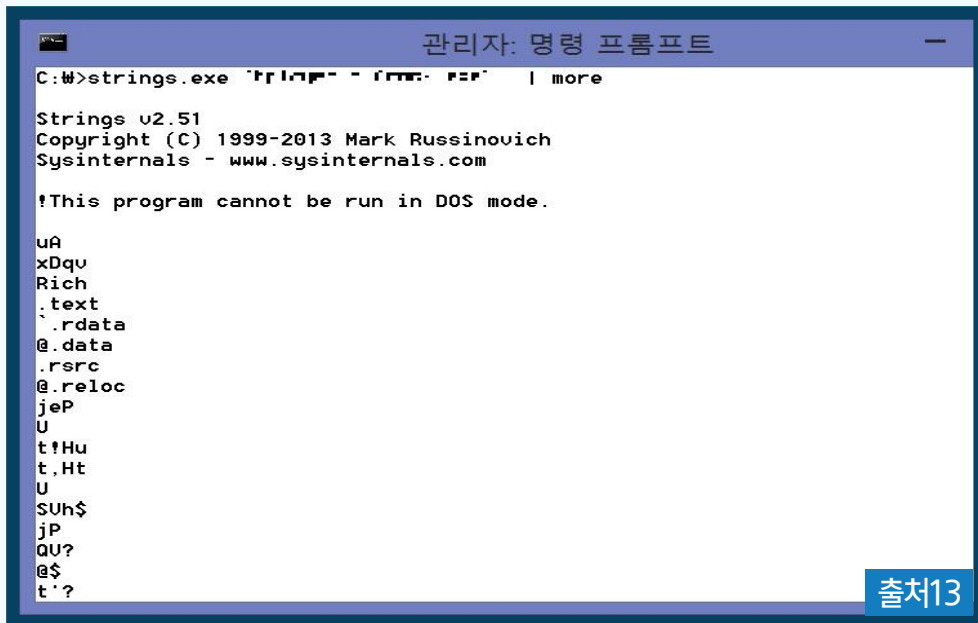


- 문자열 검색
- PE 파일 포맷
- 패킹과 난독화된 악성코드

1 Strings.exe

- ⚙️ sysinternals suite에서 제공하는 툴로 실행파일에서 문자열을 찾아 추출함
- ⚙️ 추출한 문자열의 유효성은 사용자가 하나하나 검사해야 함
- ⚙️ 코딩 단계에서 사용한 문자열이나 사용한 API 등을 알아낼 수 있음



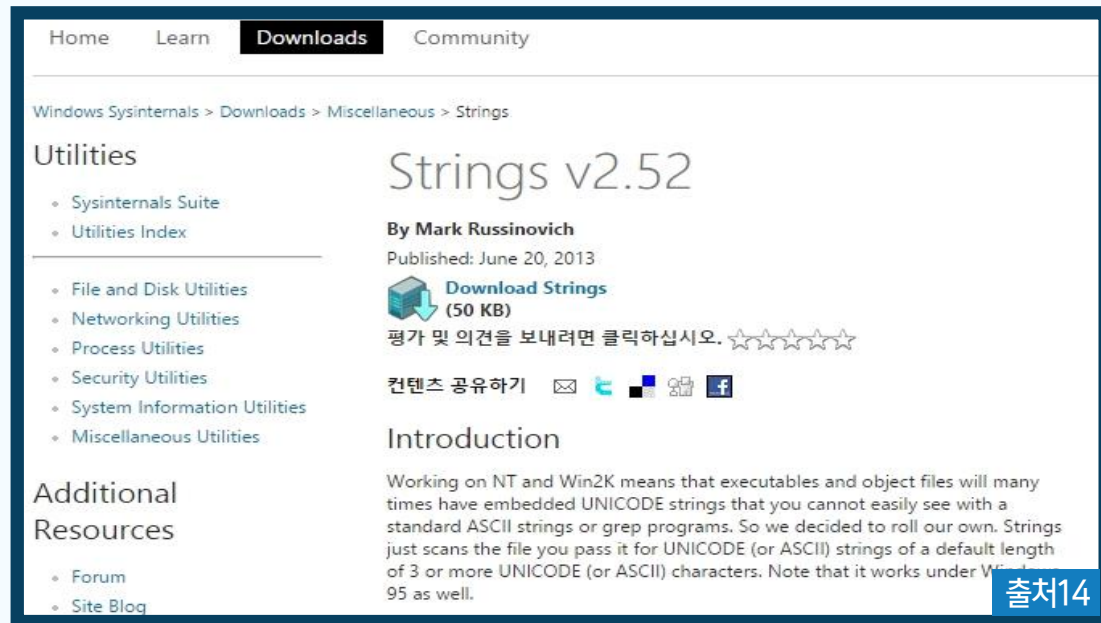
```
관리자: 명령 프롬프트
C:\>strings.exe "C:\Program Files\Internet Explorer\iexplore.exe" | more

Strings v2.51
Copyright (C) 1999-2013 Mark Russinovich
Sysinternals - www.sysinternals.com

!This program cannot be run in DOS mode.

uA
xDqu
Rich
.text
.rdata
@.data
.rsrc
@.reloc
jeP
U
t!Hu
t.Ht
U
SUn$
jP
QU?
@$
t'?
```

출처13



Home Learn Downloads Community


Windows Sysinternals > Downloads > Miscellaneous > Strings

Utilities





- Sysinternals Suite
- Utilities Index
- File and Disk Utilities
- Networking Utilities
- Process Utilities
- Security Utilities
- System Information Utilities
- Miscellaneous Utilities

Strings v2.52

By Mark Russinovich
Published: June 20, 2013

 **Download Strings**
(50 KB)

평가 및 의견을 보내려면 클릭하십시오. ☆☆☆☆☆

컨텐츠 공유하기    

Introduction

Working on NT and Win2K means that executables and object files will many times have embedded UNICODE strings that you cannot easily see with a standard ASCII strings or grep programs. So we decided to roll our own. Strings just scans the file you pass it for UNICODE (or ASCII) strings of a default length of 3 or more UNICODE (or ASCII) characters. Note that it works under Windows 95 as well.

출처14

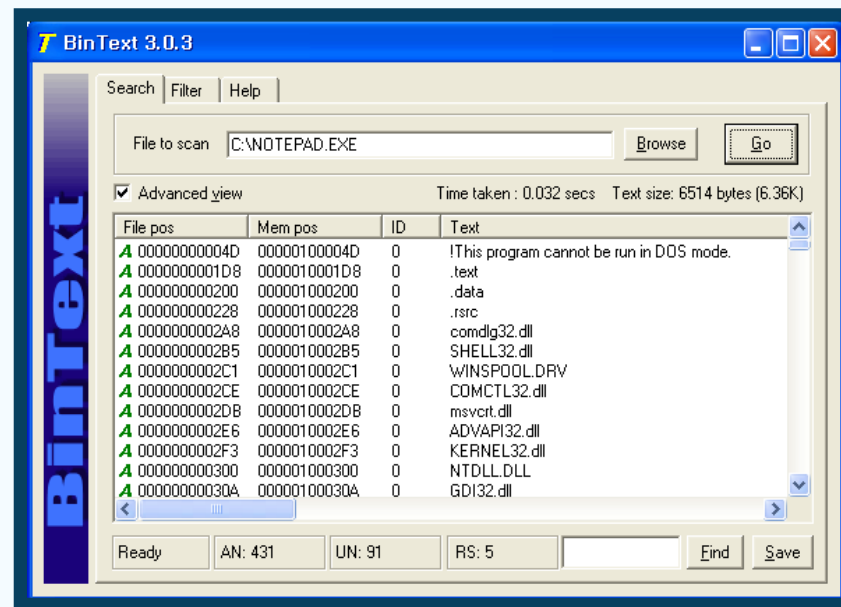
2 BinText

⚙ GUI 환경에서 분석할 수 있는 바이너리 도구



수집된 DLL 정보와 사용된 특정 문자, 사용된 함수를 통해서 소스분석 시 참고할 수 있음

| | | | |
|---|------------------|------------|-------|
| normaliz.dll | 2014-10-29 오전... | 응용 프로그램 확장 | 4KB |
| normidna.nls | 2013-06-19 오전... | NLS 파일 | 71KB |
| normnfc.nls | 2013-06-19 오전... | NLS 파일 | 49KB |
| normnfd.nls | 2013-06-19 오전... | NLS 파일 | 43KB |
| normnfdc.nls | 2013-06-19 오전... | NLS 파일 | 71KB |
| normnfd.nls | 2013-06-19 오전... | NLS 파일 | 65KB |
| <input checked="" type="checkbox"/> notepad.exe | 2015-07-10 오전... | 응용 프로그램 | 216KB |
| nrpsrv.dll | 2014-10-29 오전... | 응용 프로그램 확장 | 17KB |
| nrpsrv.dll | 2014-10-29 오전... | 응용 프로그램 확장 | 38KB |
| nshipsec.dll | 2014-10-29 오전... | 응용 프로그램 확장 | 411KB |
| nshipwp.dll | 2014-11-10 오전... | 응용 프로그램 확장 | 697KB |
| nsi.dll | 2014-10-29 오후... | 응용 프로그램 확장 | 25KB |
| nsisvc.dll | 2014-10-29 오전... | 응용 프로그램 확장 | 28KB |



1 PE 파일 포맷이란?



PE 포맷

PE 포맷(Portable Executable)은 윈도우 운영 체제에서 사용되는 실행 파일, DLL, object 코드, FON 폰트 파일[1] 등을 위한 파일 형식이다. PE 포맷은 윈도우 로더가 실행 가능한 코드를 관리하는데 필요한 정보를 캡슐화한 데이터 구조체이다. 이것은 링킹을 위한 동적 라이브러리 참조, API 익스포트와 임포트 테이블, 자원 관리 데이터 그리고 TLS 데이터를 포함한다. 윈도우 NT 운영체제에서, PE 포맷은 EXE, DLL, SYS (디바이스 드라이버), 그리고 다른 파일 타입들에서 쓰인다. 통일 확장 펌웨어 인터페이스 (EFI) 설명서는 PE가 EFI 환경에서 표준 실행 파일 형식이라고 언급한다.

윈도우 NT 운영 체제에서, PE는 현재 IA-32, IA-64, x86-64 (AMD64/Intel64), 그리고 ARM instruction set architectures (ISAs)를 지원한다. 윈도우 2000 이전에서, 윈도우 NT는 (그리고 PE) MIPS, Alpha, 그리고 파워PC ISAs를 지원했다. PE가 윈도우 CE에서 사용되므로, 이것은 MIPS, ARM (Thumb 포함), 그리고 SuperH ISA의 변형들을 지원하고 있다.

출처16

② PE 파일의 종류

⚙ PE 파일 포맷을 가지고 있는 주요 파일들

| 종류 | 확장자 |
|----------|--------------------|
| 실행 파일 | exe, scr |
| 라이브러리 파일 | dll, ocx, cpl, drv |
| 오브젝트 파일 | obj |
| 드라이버 파일 | sys, vxd |

출처17

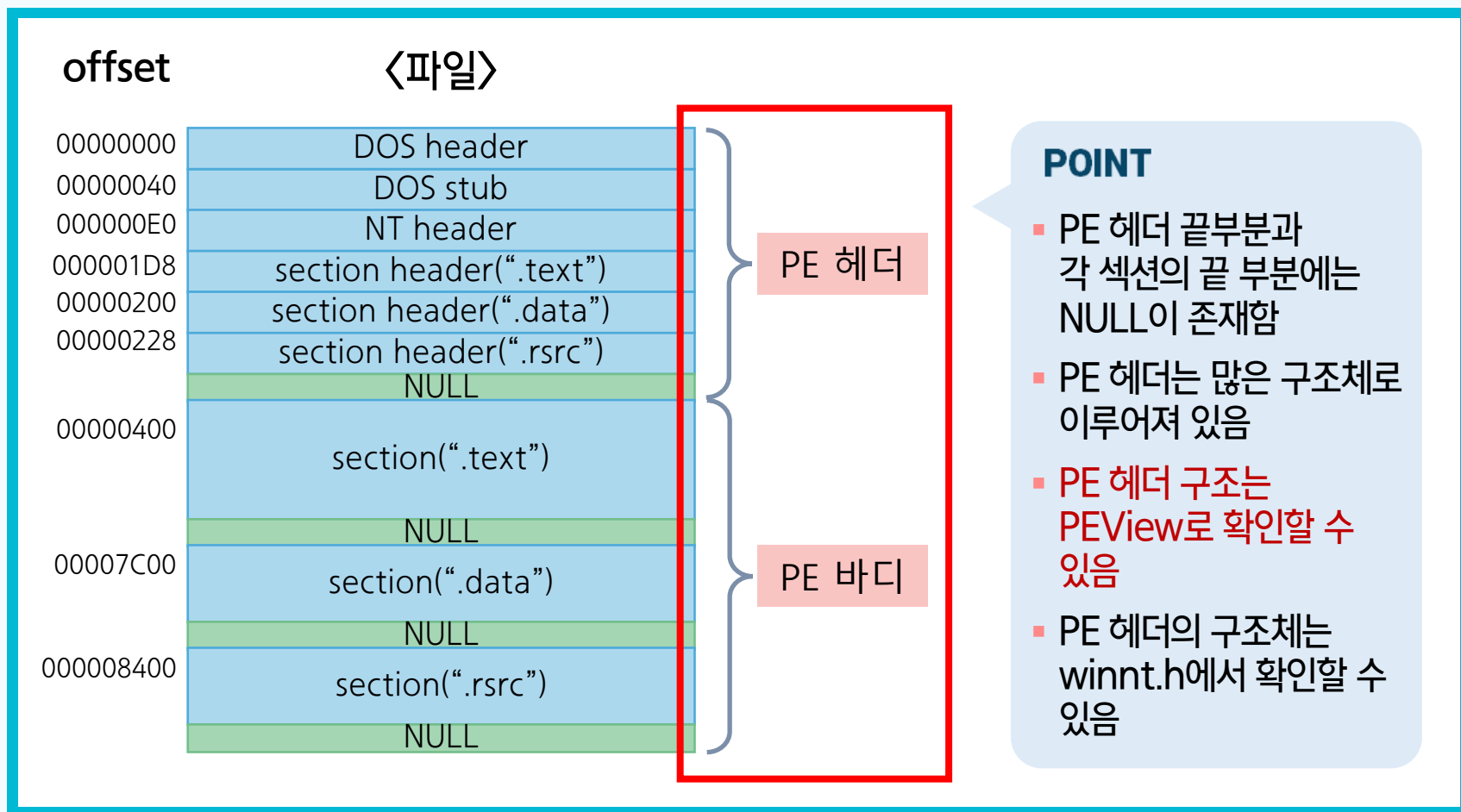
3 PE 파일의 구조

0 PE 파일이 메모리에 로딩되는 모습

| offset | 〈파일〉 |
|-----------|-------------------------|
| 00000000 | DOS header |
| 00000040 | DOS stub |
| 000000E0 | NT header |
| 000001D8 | section header(".text") |
| 00000200 | section header(".data") |
| 00000228 | section header(".rsrc") |
| | NULL |
| 00000400 | section(".text") |
| | NULL |
| 00007C00 | section(".data") |
| | NULL |
| 000008400 | section(".rsrc") |
| | NULL |

| 〈메모리〉 | address |
|-------------------------|----------|
| DOS header | 01000000 |
| DOS stub | 01000040 |
| NT header | 010000E0 |
| section header(".text") | 010001D8 |
| section header(".data") | 01000200 |
| section header(".rsrc") | 01000228 |
| NULL | |
| section(".text") | 01001000 |
| NULL | |
| section(".data") | 01009000 |
| NULL | |
| section(".rsrc") | 0100B000 |
| NULL | |

3 PE 파일의 구조



파일에서는 offset, 메모리에서는 VA(절대 주소)로 위치를 표현함

2 PE 파일 포맷

3 PE 파일의 구조

PEView

PEView - C:\Windows\notepad.exe

File View Go Help

notepad.exe

- IMAGE_DOS_HEADER
- MS-DOS Stub Program
- IMAGE_NT_HEADERS
 - IMAGE_SECTION_HEADER .text
 - IMAGE_SECTION_HEADER .data
 - IMAGE_SECTION_HEADER .pdata
 - IMAGE_SECTION_HEADER .idata
 - IMAGE_SECTION_HEADER .rsrc
 - IMAGE_SECTION_HEADER .reloc
- SECTION .text
- SECTION .data
- SECTION .pdata
- SECTION .idata
- SECTION .rsrc
- SECTION .reloc

| pFile | Raw Data | Value |
|----------|---|-----------------------|
| 00000000 | 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 | MZ..... |
| 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 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00 |!..L.!Th |
| 00000040 | 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 |is program canno |
| 00000050 | 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F | t be run in DOS |
| 00000060 | 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 | mode...\$..... |
| 00000070 | 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 | s...!7...r7...r |
| 00000080 | 73 B2 8D 21 37 D3 E3 72 37 D3 E3 72 37 D3 E3 72 | ...r6...r...r5...r |
| 00000090 | EA 2C 2C 72 36 D3 E3 72 EA 2C 2E 72 35 D3 E3 72 | ...-r"...r...r(r...r |
| 000000A0 | EA 2C 2D 72 22 D3 E3 72 EA 2C 28 72 2C D3 E3 72 | 7...r...r...r1r\$...r |
| 000000B0 | 37 D3 E2 72 EA D3 E3 72 EA 2C 31 72 24 D3 E3 72 | ...*r6...r...r/r6...r |
| 000000C0 | EA 2C 2A 72 36 D3 E3 72 EA 2C 2F 72 36 D3 E3 72 | Rich7...r..... |
| 000000D0 | 52 69 63 68 37 D3 E3 72 00 00 00 00 00 00 00 00 |PE..d... |
| 000000E0 | 00 00 00 00 00 00 00 00 50 45 00 00 64 86 06 00 | ...U....." |
| 000000F0 | CC AB 9E 55 00 00 00 00 00 00 00 00 F0 00 22 00 |@..... |
| 00000100 | 0B 02 0B 00 00 82 01 00 00 EE 01 00 00 00 00 00 | |
| 00000110 | 80 2C 00 00 00 10 00 00 00 00 00 40 01 00 00 00 | |
| 00000120 | 00 10 00 00 00 02 00 00 06 00 03 00 06 00 03 00 | |
| 00000130 | 06 00 03 00 00 00 00 00 00 B0 03 00 00 04 00 00 | |
| 00000140 | 33 CE 03 00 02 00 60 C1 00 00 08 00 00 00 00 00 | 3..... |
| 00000150 | 00 10 01 00 00 00 00 00 00 00 10 00 00 00 00 00 | |
| 00000160 | 00 10 00 00 00 00 00 00 00 00 00 00 10 00 00 00 | |

Viewing notepad.exe

출처18


3 PE 파일의 구조

1 DOS header

2 DOS stub

3 NT header

4 section header

- PE 파일 포맷을 만들 때 그 당시에 쓰던 도스 파일에 대한 하위 호환성을 고려함
 - 중요한 멤버는 첫 번째 **e_magic**(DOS signature)와 마지막 멤버인 **e_lfanew**(NT header의 시작 주소) 두 가지임
-  **POINT**
 - e_magic은 DOS 시절 실행파일을 설계한 Mark Zbikowski의 이니셜로 만든 것임
 - e_lfanew는 NT Header의 시작 주소를 나타냄
- DOS header의 e_magic, e_lfanew 값 중 하나라도 변경된다면 프로그램이 정상적으로 실행되지 않음

3 PE 파일의 구조

1 DOS header

2 DOS stub

3 NT header

4 section header

[PE파일 구조 중 DOS header]

| 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 | MZ.....ÿÿ.. |
| 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 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | EO | 00 | 00 | 00 |à... |
| 00000040 | 0E | 1F | BA | 0E | 00 | B4 | 09 | CD | 21 | B8 | 01 | 4C | CD | 21 | 54 | 68 | ..°..'.í!..Lí!Th |
| 00000050 | 69 | 73 | 20 | 70 | 72 | 6F | 67 | 72 | 61 | 6D | 20 | 63 | 61 | 6E | 6E | 6F | is program canno |
| 00000060 | 74 | 20 | 62 | 65 | 20 | 72 | 75 | 6E | 20 | 69 | 6E | 20 | 44 | 4F | 53 | 20 | t be run in DOS |
| 00000070 | 6D | 6F | 64 | 65 | 2E | 0D | 0D | 0A | 24 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | mode....\$..... |
| 00000080 | EC | 85 | 5B | A1 | A8 | E4 | 35 | F2 | A8 | E4 | 35 | F2 | A8 | E4 | 35 | F2 | i...[;`ä50`ä50`ä50 |
| 00000090 | 6B | EB | 3A | F2 | A9 | E4 | 35 | F2 | 6B | EB | 55 | F2 | A9 | E4 | 35 | F2 | kë:ò@ä50këUò@ä50 |
| 000000A0 | 6B | EB | 68 | F2 | BB | E4 | 35 | F2 | A8 | E4 | 34 | F2 | 63 | E4 | 35 | F2 | këhò»ä50`ä4òcä50 |
| 000000B0 | 6B | EB | 6B | F2 | A9 | E4 | 35 | F2 | 6B | EB | 6A | F2 | BF | E4 | 35 | F2 | këkò@ä50këjòçä50 |
| 000000C0 | 6B | EB | 6F | F2 | A9 | E4 | 35 | F2 | 52 | 69 | 63 | 68 | A8 | E4 | 35 | F2 | këoò@ä50Rich`ä50 |
| 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 | 87 | 52 | 02 | 48 | 00 | 00 | 00 | 00 | PE..L...+R.H.... |
| 000000F0 | 00 | 00 | 00 | 00 | EO | 00 | 0F | 01 | 0B | 01 | 07 | 0A | 00 | 78 | 00 | 00 |à.....x.. |

3 PE 파일의 구조

1 DOS header

2 DOS stub

3 NT header

4 section header

- DOS stub의 존재 여부는 옵션이며 크기도 일정하지 않음
 - 있어도 되고 없어도 되는 부분임
 - 크기가 정해져 있는 것이 아니기 때문에 DOS stub 부분에 악성코드를 심기도 함
- DOS stub에는 어셈블리어 코드와 This program cannot be run in DOS mode라는 문자열이 있음
- DOS 모드에서 해당 파일을 실행하면 위의 문자열을 출력하고 종료함

[PE파일 구조 중 DOS stub]

| | | |
|----------|---|-------------------|
| 00000030 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |à... |
| 00000040 | 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 | ..°...'.í!_.Lí!Th |
| 00000050 | 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F | is program canno |
| 00000060 | 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 | t be run in DOS |
| 00000070 | 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 | mode....\$...... |
| 00000080 | EC 85 5B A1 A8 E4 35 F2 A8 E4 35 F2 A8 E4 35 F2 | i...[;`a5`a5`a5` |
| 00000090 | 6B EB 3A F2 A9 E4 35 F2 6B EB 55 F2 A9 E4 35 F2 | kë:ò@a5òkëUò@a5ò |
| 000000A0 | 6B EB 68 F2 BB E4 35 F2 A8 E4 34 F2 63 E4 35 F2 | këhò»a5`a4òcä5ò |
| 000000B0 | 6B EB 6B F2 A9 E4 35 F2 6B EB 6A F2 BF E4 35 F2 | këkò@a5òkëjòçä5ò |
| 000000C0 | 6B EB 6F F2 A9 E4 35 F2 52 69 63 68 A8 E4 35 F2 | këoò@a5òRich`a5ò |
| 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 87 52 02 48 00 00 00 00 | PE...I...#R.H.... |

3 PE 파일의 구조

1 DOS header

2 DOS stub

3 NT header

4 section header

■ NT header 구조체

- DOS header의 마지막 멤버가 가리키는 주소에서 시작함
- 3개의 멤버로 구성되어 있음

signature

file header

optional header

→ signature는 50450000h (PE..)의 값을 가지고 나머지 두 멤버는 또 다른 구조체임

[PE파일 구조 중 DOS header]

| | | |
|----------|--|------------------|
| 00000000 | 50 45 00 00 4C 01 03 00 87 52 02 48 00 00 00 00 | PE..L...+R.H... |
| 000000E0 | 00 00 00 00 E0 00 0F 01 0B 01 07 0A 00 78 00 00 | ...à.....x.. |
| 00000100 | 00 8C 00 00 00 00 00 00 9D 73 00 00 00 10 00 00 | .@.....s..... |
| 00000110 | 00 90 00 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 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 00 00 10 00 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 00 04 83 00 00 | .v..È....°...f.. |
| 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 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 |"....@... |
| 000001B0 | 50 02 00 00 D0 00 00 00 00 10 00 00 48 03 00 00 | P...D.....H... |
| 000001C0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 000001D0 | 00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00 |text... |

3 PE 파일의 구조

- #### 4 section header

- file header
 - 4가지 주요 멤버

| | |
|------------------------|---|
| machine ① | <ul style="list-style-type: none"> CPU별 고유한 값, Intel x86=0x014C, Intel x64=0x0200의 값을 가짐 |
| numberofsections ② | <ul style="list-style-type: none"> 섹션의 개수를 나타냄 |
| sizeofoptionalheader ③ | <ul style="list-style-type: none"> optional header의 크기를 나타냄 |
| characteristics ④ | <ul style="list-style-type: none"> 파일의 속성을 나타냄 ✓ 실행이 가능한지 또는 DLL파일인지 등의 정보가 있음 |

[NT header 중 file header]

| | | | | | | |
|----------|----------|---|-------------|---|----------------------|-----------------|
| 00000000 | 00 00 00 | 1 | 00 00 00 00 | 2 | 00 00 00 00 00 00 00 | |
| 000000E0 | 50 45 00 | | 4C 01 03 00 | | 52 02 48 00 00 00 00 | PE..L...#R.H... |
| 000000F0 | 00 00 00 | 3 | E0 00 0F 01 | 4 | 01 07 0A 00 78 00 00 |à.....x.. |
| 00000100 | 00 00 00 | | 00 00 00 00 | | 70 00 00 00 10 00 00 | m - |

3 PE 파일의 구조

1 DOS header

2 DOS stub

3 NT header

4 section header

- optional header
 - 10가지 주요 멤버

[NT header 중 optional header]

| | | |
|----------|---|------------------|
| 00000000 | 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 7 52 02 48 00 00 00 00 | PE..L...#R.H.... |
| 000000F0 | 00 00 00 00 00 00 0F 01 0B 01 07 0A 78 00 00 |à.....x.. |
| 00000100 | 00 8C 00 00 00 00 00 00 9D 73 00 00 10 00 00 | Æ.....s..... |
| 00000110 | 00 90 00 00 00 00 01 00 00 10 00 00 00 02 00 00 | |
| 00000120 | 05 00 01 00 05 00 01 00 00 00 00 00 00 00 00 | |
| 00000130 | 00 40 01 00 00 04 00 00 26 01 00 02 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 | |
| 00000160 | 04 76 00 00 C8 00 00 00 00 B0 00 00 04 83 00 00 | .v..È....°...f.. |
| 00000170 | 00 00 00 00 00 00 00 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 | |
| 000001A0 | 00 00 00 00 00 00 00 00 A8 18 00 00 40 00 00 00 |"....@... |
| 000001B0 | 50 02 00 00 D0 00 00 00 00 10 00 00 48 03 00 00 | P...Ð.....H... |
| 000001C0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 000001D0 | 00 00 00 00 00 00 00 00 25 65 78 74 00 00 00 |text... |
| 000001E0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .. |

③ PE 파일의 구조

1 DOS header

2 DOS stub

3 NT header

4 section header

- optional header
 - 10가지 주요 멤버

Magic ①

- magic 넘버는 32비트면 10B, 64비트면 20B를 가짐

AddressOfEntryPoint ②

- EP의 RVA(Relative Virtual Address) 값을 가지고 있음

ImageBase ③

- PE 파일이 로딩되는 시작 주소임
- 파일을 메모리에 로딩 후 EIP 레지스터의 값
$$= \text{ImageBase} + \text{AddressOfEntryPoint}$$

SectionAlignment ④

- 메모리에서 섹션의 최소단위

FileAlignment ⑤

- 파일에서 섹션의 최소단위

3 PE 파일의 구조

1 DOS header

2 DOS stub

3 NT header

4 section header

- optional header
 - 10가지 주요 멤버

SizeOfImage ⑥

- PE 파일이 메모리에 로딩되었을 때의 크기

SizeOfHeader ⑦

- PE헤더의 전체 크기

Subsystem ⑧

- GUI, CUI 버전인지 드라이브파일인지 나타냄
✓ 1 = 드라이버, 2 = GUI, 3 = CUI

NumberOfRvaAndSizes ⑨

- DataDirectory 배열의 개수를 나타냄

DataDirectory ⑩

- IMAGE_DATA_DIRECTORY 구조체의 배열, 각 항목은 정해진 값을 가지고 각 항목 table이 어디에 위치해 있는지 RVA 값과 크기를 나타냄

3 PE 파일의 구조

1 DOS header

2 DOS stub

3 NT header

4 section header

5가지 주요 멤버

| | |
|--------------------|----------------------|
| VirtualSize ① | 메모리에서 섹션이 차지하는 크기 |
| VirtualAddress ② | 메모리에서 섹션의 시작 주소(RVA) |
| SizeOfRawData ③ | 파일에서 섹션이 차지하는 크기 |
| PointerToRawData ④ | 파일에서 섹션의 시작 위치 |
| Characteristics ⑤ | 섹션의 속성(bit OR) |

[PE파일 구조 중 section header]

| | | | |
|----------|-------------------------|-------------------------|---------------|
| 000001D0 | 00 00 00 00 00 00 00 00 | 2E 75 78 74 00 00 00 00 |text... |
| 000001E0 | 48 77 00 00 00 10 00 00 | 00 78 00 00 00 04 00 00 | Hw.....x..... |
| 000001F0 | 00 00 00 00 00 00 00 00 | 00 00 00 00 20 00 00 60 |` |
| 00000200 | 2E 5A 51 74 51 00 00 00 | 38 1B 00 00 00 00 00 00 | data " |

3 PE 파일의 구조

1 DOS header

2 DOS stub

3 NT header

4 section header

■ 종류

| | |
|-----------|--|
| .text 섹션 | 컴파일된 코드, 코드, 실행, 읽기 속성, 컴파일 후의 결과가 저장되는 섹션 |
| .data 섹션 | 읽기/쓰기가 가능하고 초기화된 전역변수, 상수가 존재하는 섹션 |
| .rdata 섹션 | 읽기만 가능하고 문자열 상수나 const로 선언된 변수처럼 읽기만 가능한 읽기 전용데이터 섹션 |
| .idata 섹션 | 읽기/쓰기 모두 가능하고 IAT와 관련된 정보가 들어있는 섹션 |
| .edata 섹션 | 읽기만 가능하고 EAT와 관련된 정보가 들어 있는 섹션 |
| .bss 섹션 | 읽기/쓰기가 가능하고 초기화 되지 않은 전역변수 |
| .rsrc 섹션 | 읽기만 가능하고 리소스가(아이콘 등)이 저장되는 섹션 |

3 패킹과 난독화된 악성코드

1 패킹이란?

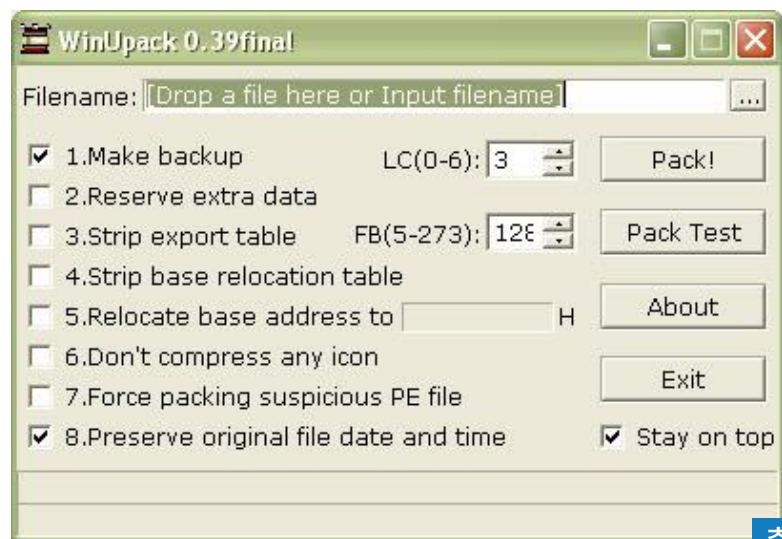
패킹



대표적인 예 : UPX, ASPack, Upack, PESpin 등이 있음

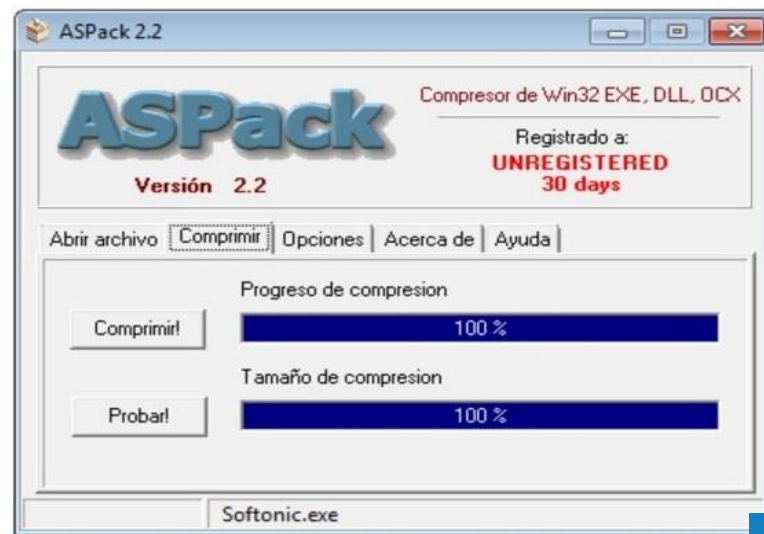
PE 파일의 크기를 줄이거나 내부 코드와 리소스를 감추기 위해 사용하는 기술

WinUpack 0.39final



출처19

ASPack 2.2



출처20

2 UPX

0 UPX를 이용해서 notepad.exe를 패킹함

```
C:\W>c:\W\upx\upx.exe c:\W\notepad.exe
```

```
Ultimate Packer for eXecutables
```

```
Copyright (C) 1996 - 2011
```

```
UPX 3.08w Markus Oberhumer, Laszlo Molnar & John Reiser Dec 12th 2011
```

| File size | Ratio | Format | Name |
|----------------|--------|----------|-------------|
| 67584 -> 48128 | 71.21% | win32/pe | notepad.exe |

```
Packed 1 file.
```

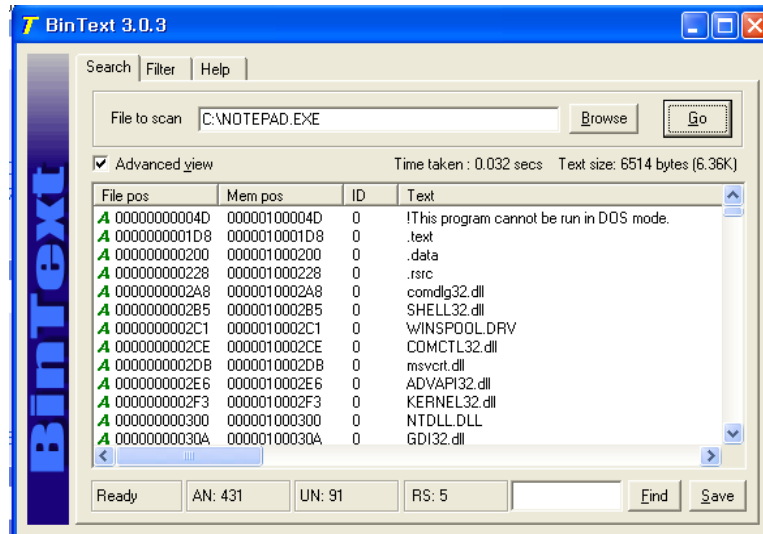
출처21

3 패킹과 난독화된 악성코드

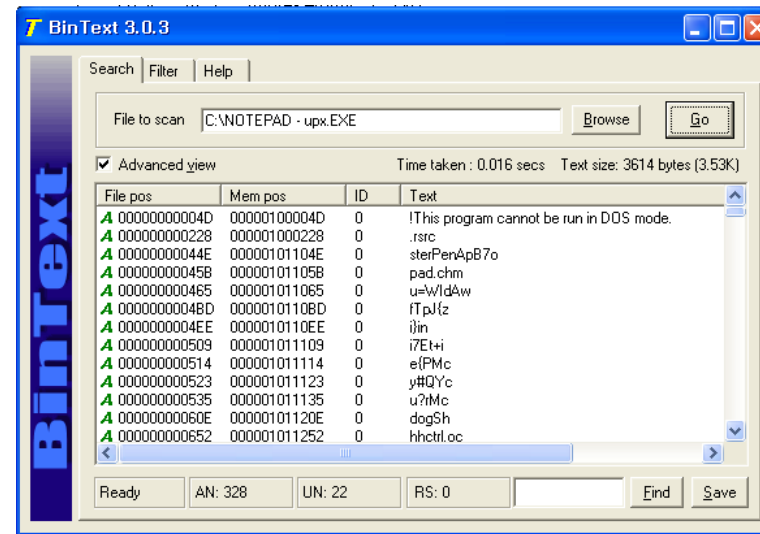
2 UPX

UPX를 이용해서 notepad.exe를 패킹한 결과

[패킹 전]



[패킹 후]



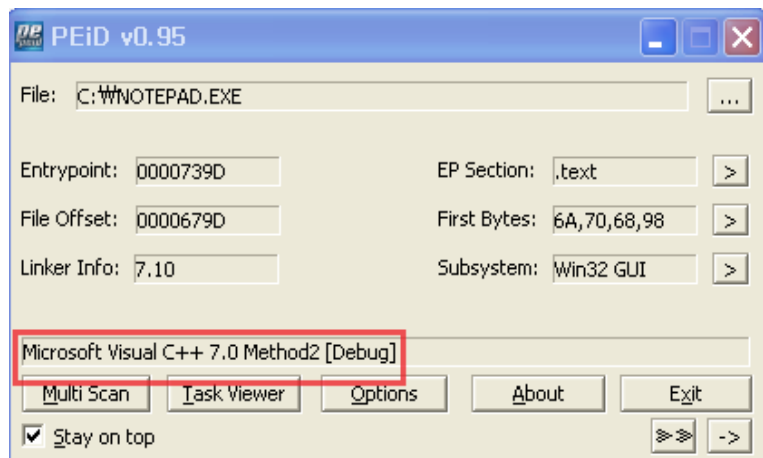
(1/3)

출처22

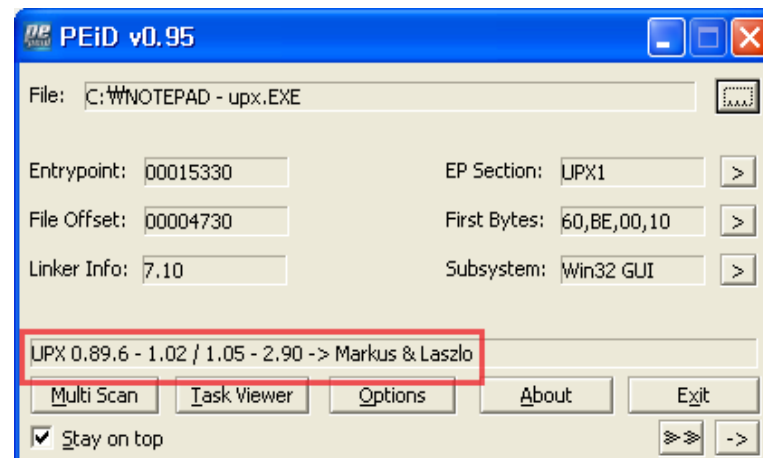
2 UPX

UPX를 이용해서 notepad.exe를 패킹한 결과

[패킹 전]



[패킹 후]



- PEiD로 컴파일러의 종류와 패킹 여부를 알 수 있음



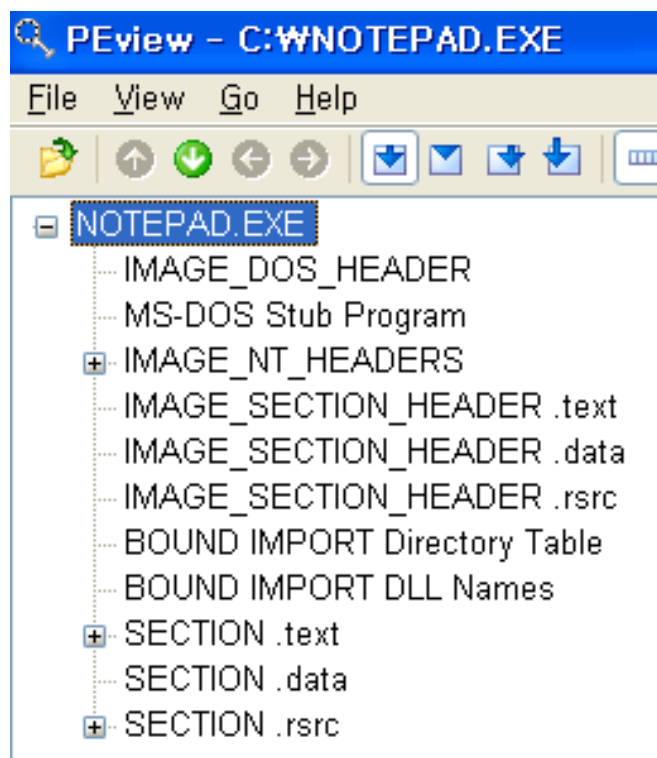
(2/3)

출처23

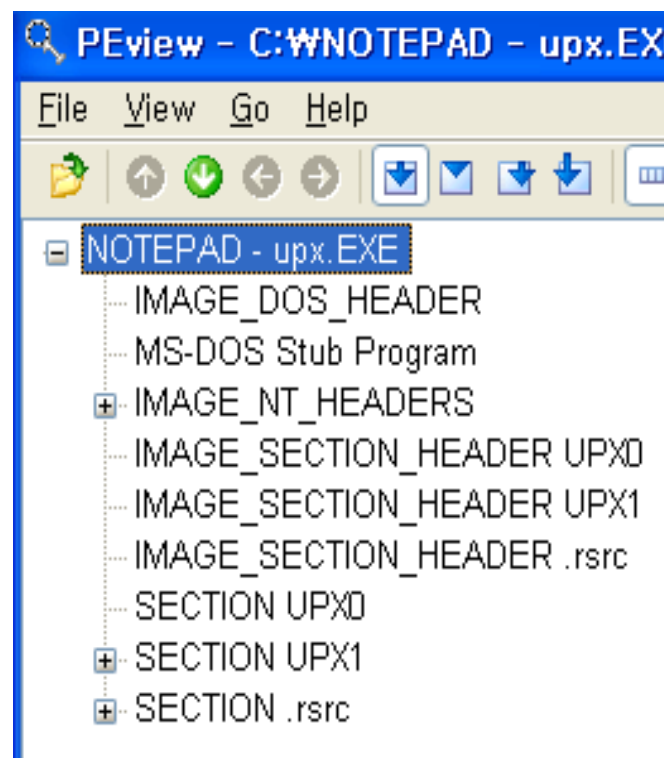
2 UPX

UPX를 이용해서 notepad.exe를 패킹한 결과

[패킹 전]



[패킹 후]



(3/3)

출처24