

PE 재배치

SCP 20학번 박준형

목차

table of contents

- 1 PE 재배치란?
- 2 PE 재배치 발생시 수행되는 작업
- 3 PE 재배치 동작 원리
- 4 PE 재배치 동작 실습

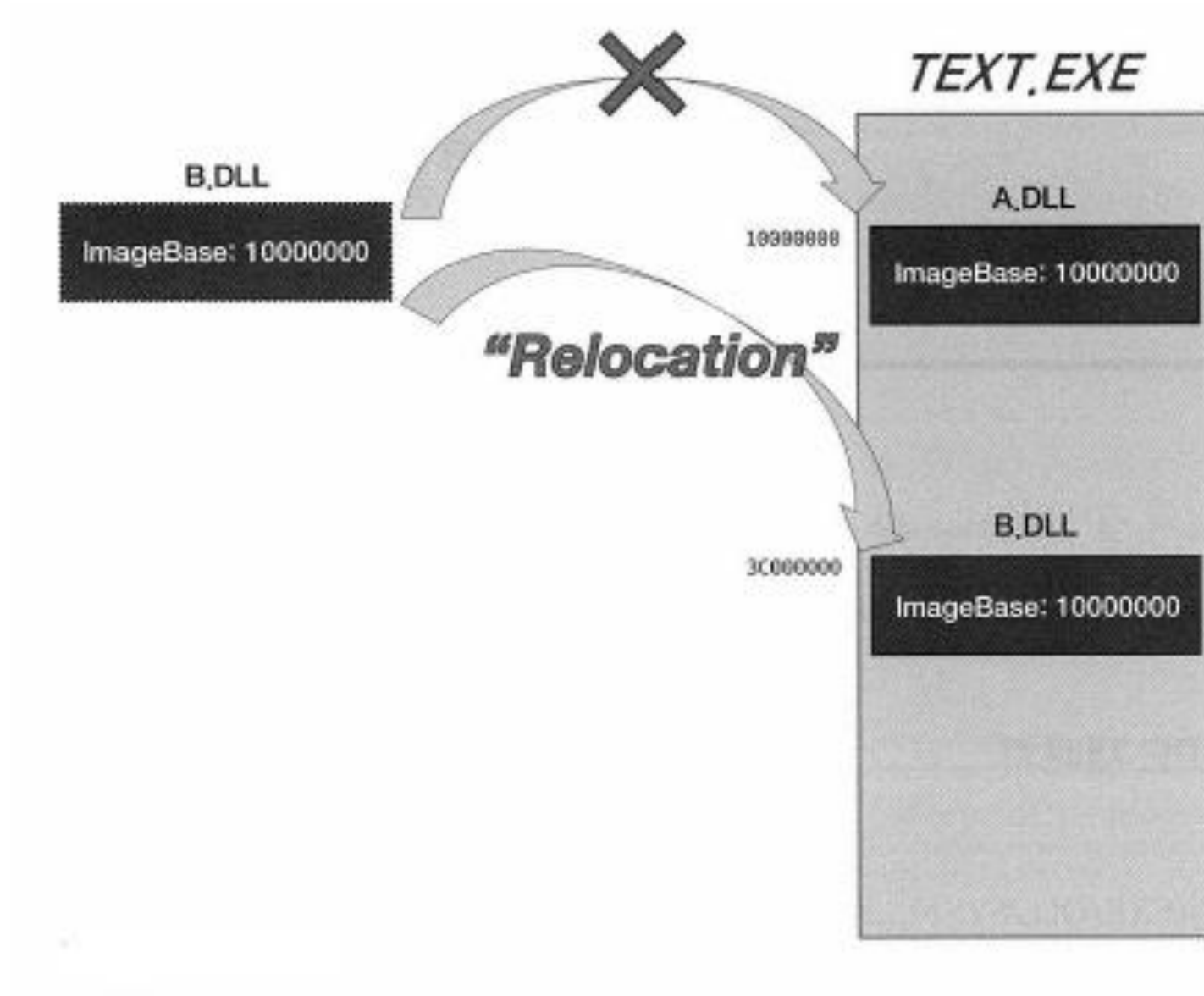
PE 재배치란?

Part 1 PE 재배치란?

PE 파일(EXE, DLL, SYS)이 프로세스 가상 메모리에 로딩(loading) 될 때 PE 헤더의 ImageBase 주소에 로딩됩니다. DLL(SYS) 파일의 경우 ImageBase 위치에 이미 다른 DLL(SYS) 파일이 로딩되어 있다면 다른 비어 있는 주소 공간에 로딩됩니다.

-> 즉, PE 재배치란 PE 파일이 ImageBase에 로딩되지 못하고 다른 주소에 로딩될 때 수행되는 작업들을 의미합니다.

Part 1 PE 재배치란? - DLL/SYS



Part 1 PE 재배치란? – EXE

ASLR(Address Space Layout Randomization): EXE 파일이 실행될 때
마다 랜덤한 주소에 로딩하는 것

00000100	00022400	Size of Initialized Data
00000104	00000000	Size of Uninitialized Data
00000108	00003689	Address of Entry Point
0000010C	00001000	Base of Code
00000110	0000C000	Base of Data
00000114	01000000	Image Base
00000118	00001000	Section Alignment

첫 번째 실행

Address	Hex dump	Disassembly
00BC3689	\$ E8 C5F9FFFF	CALL notepad.00BC3053
00BC368E	. 6A 58	PUSH 58

두 번째 실행

Address	Hex dump	Disassembly
008A3689	\$ E8 C5F9FFFF	CALL notepad.008A3053
008A368E	. 6A 58	PUSH 58

세 번째 실행

Address	Hex dump	Disassembly
008A3689	\$ E8 C5F9FFFF	CALL notepad.008A3053
008A368E	. 6A 58	PUSH 58

PE 재배치 발생시 수행되는 작업

Part 2 PE 재배치 발생시 수행되는 작업

PEView - C:\Users\User\Desktop\SCP\개인공부\리버싱 개인공부\snapshot_2023-04-30_13-19\실습예제_2중압축\실습예제\W02_PE_File_Forma...

File View Go Help

notepad.exe

- IMAGE_DOS_HEADER
- MS-DOS Stub Program
- IMAGE_NT_HEADERS
 - Signature
 - IMAGE_FILE_HEADER
 - IMAGE_OPTIONAL_HEADER
- IMAGE_SECTION_HEADER .text
- IMAGE_SECTION_HEADER .data
- IMAGE_SECTION_HEADER .rsrc
- IMAGE_SECTION_HEADER .reloc
- BOUND_IMPORT Directory Table
- BOUND_IMPORT DLL Names
- SECTION .text
- SECTION .data
- SECTION .rsrc
- SECTION .reloc

pFile	Data	Description	Value
000000F8	010B	Magic	IMAGE_NT_OPTIONAL_HDR32_MAGIC
000000FA	09	Major Linker Version	
000000FB	00	Minor Linker Version	
000000FC	0000A800	Size of Code	
00000100	00022400	Size of Initialized Data	
00000104	00000000	Size of Uninitialized Data	
00000108	00003689	Address of Entry Point	
0000010C	00001000	Base of Code	
00000110	0000C000	Base of Data	
00000114	01000000	Image Base	
00000118	00001000	Section Alignment	
0000011C	00000200	File Alignment	
00000120	0006	Major O/S Version	
00000122	0001	Minor O/S Version	
00000124	0006	Major Image Version	
00000126	0001	Minor Image Version	
00000128	0006	Major Subsystem Version	
0000012A	0001	Minor Subsystem Version	
0000012C	00000000	Win32 Version Value	
00000130	00030000	Size of Image	
00000134	00000400	Size of Headers	
00000138	00039741	Checksum	
0000013C	0002	Subsystem	IMAGE_SUBSYSTEM_WINDOWS_GUI
0000013E	8140	DLL Characteristics	IMAGE_DLLCHARACTERISTICS_DYNAMIC_BASE IMAGE_DLLCHARACTERISTICS_NX_COMPAT IMAGE_DLLCHARACTERISTICS_TERMINAL_SERVER_
00000140	00040000	Size of Stack Reserve	

Viewing IMAGE_OPTIONAL_HEADER

Part 2 PE 재배치 발생시 수행되는 작업

OllyDbg - notepad.exe

```
File View Debug Options Window Help
CPU - main thread, module notepad

002F3689 $ E8 C5F9FFFF CALL notepad.002F3053
002F368E . 6A 58 PUSH 58
002F3690 . 68 A0372F00 PUSH notepad.002F37A0
002F3695 . E8 72040000 CALL notepad.002F3B0C
002F369A . 33DB XOR EBX,EBX
002F369C . 895D E4 MOV DWORD PTR SS:[EBP-1C],EBX
002F369F . 895D FC MOV DWORD PTR SS:[EBP-4],EBX
002F36A2 . 8D45 98 LEA EAX,DWORD PTR SS:[EBP-68]
002F36A5 . 50 PUSH EAX
002F36A6 . FF15 FC102F00 CALL DWORD PTR DS:[<&KERNEL32.GetStartupInfoA>]
002F36AC . C745 FC FEFFFF MOV DWORD PTR SS:[EBP-4],-2
002F36B3 . C745 FC 010000 MOV DWORD PTR SS:[EBP-4],1
002F36BA . 64:A1 18000000 MOV EAX,DWORD PTR FS:[18]
002F36C0 . 8B70 04 MOV ESI,DWORD PTR DS:[EAX+4]
002F36C3 . BF 5CC22F00 MOV EDI,notepad.002FC25C
002F36C8 > 6A 00 PUSH 0
002F36CA . 56 PUSH ESI
002F36CB . 57 PUSH EDI
002F36CC . FF15 00112F00 CALL DWORD PTR DS:[<&KERNEL32.InterlockedCompareExchange>]
002F36D2 . 85C0 TEST EAX,EAX
002F36D4 . 0F85 36350000 JNZ notepad.002F6C10
002F36DA . 33F6 XOR ESI,ESI
002F36DC . 46 INC ESI
002F36DD > A1 A4C02F00 MOV EAX,DWORD PTR DS:[2FC0A4]
002F36E2 . 3BC6 CMP EAX,ESI
002F36E4 . 0F84 44350000 JE notepad.002F6C2E
002F36EA . A1 A4C02F00 MOV EAX,DWORD PTR DS:[2FC0A4]
002F36EF . 85C0 TEST EAX,EAX
002F36F1 . 0F85 7A0C0000 JNZ notepad.002F4371
002F36F7 . 8935 A4C02F00 MOV DWORD PTR DS:[2FC0A4],ESI

DS:[002F10FC]=76391320 (KERNEL32.GetStartupInfoA)
```

PE 재배포치 동작 원리

Part 3 PE 재배치 동작 원리

- 1) 프로그램에서 하드코딩된 주소 위치를 찾는다.
- 2) 값을 읽은 후 ImageBase를 뺀다. (VA -> RVA)
- 3) 실제로 로딩된 주소를 더한다. (RVA -> VA)

Part 3 PE 재배치 동작 원리 – Base Relocation Table

PEView - C:\Users\User\Desktop\SCP\개인공부\리버싱 개인공부\snapshot_2023-04-30_13-19\실습예제_2중압축\실습예제\W02_PE_File_Forma...

File View Go Help

notepad.exe

- ... IMAGE_DOS_HEADER
- ... MS-DOS Stub Program
- [-] IMAGE_NT_HEADERS
 - ... Signature
 - ... IMAGE_FILE_HEADER
 - ... IMAGE_OPTIONAL_HEADER
 - ... IMAGE_SECTION_HEADER .text
 - ... IMAGE_SECTION_HEADER .data
 - ... IMAGE_SECTION_HEADER .rsrc
 - ... IMAGE_SECTION_HEADER .reloc
- ... BOUND_IMPORT Directory Table
- ... BOUND_IMPORT DLL Names
- [+] SECTION .text
- ... SECTION .data
- [+] SECTION .rsrc
- [+] SECTION .reloc

RVA	Data	Description	Value
0000014C	00001000	Size of Heap Commit	
00000150	00000000	Loader Flags	
00000154	00000010	Number of Data Directories	
00000158	00000000	RVA	EXPORT Table
0000015C	00000000	Size	
00000160	0000A048	RVA	IMPORT Table
00000164	0000012C	Size	
00000168	0000F000	RVA	RESOURCE Table
0000016C	0001F160	Size	
00000170	00000000	RVA	EXCEPTION Table
00000174	00000000	Size	
00000178	00000000	Offset	CERTIFICATE Table
0000017C	00000000	Size	
00000180	0002F000	RVA	BASE RELOCATION Table
00000184	00000E34	Size	
00000188	0000B62C	RVA	DEBUG Directory
0000018C	00000038	Size	
00000190	00000000	RVA	Architecture Specific Data
00000194	00000000	Size	
00000198	00000000	RVA	GLOBAL POINTER Register
0000019C	00000000	Size	
000001A0	00000000	RVA	TLS Table
000001A4	00000000	Size	
000001A8	00006D58	RVA	LOAD CONFIGURATION Table
000001AC	00000040	Size	
000001B0	00000278	RVA	BOUND_IMPORT Table
000001B4	00000128	Size	
000001B8	00001000	RVA	IMPORT Address Table

Viewing IMAGE_OPTIONAL_HEADER

Part 3 PE 재배치 동작 원리 – Base Relocation Table

PEView - C:\Users\User\Desktop\SCP\개인공부\리버싱 개인공부\snapshot_2023-04-30_13-19\실습예제_2중압축\실습예제\W02_PE_File_Forma...

File View Go Help

notepad.exe

- IMAGE_DOS_HEADER
- MS-DOS Stub Program
- IMAGE_NT_HEADERS
 - Signature
 - IMAGE_FILE_HEADER
 - IMAGE_OPTIONAL_HEADER
 - IMAGE_SECTION_HEADER .text
 - IMAGE_SECTION_HEADER .data
 - IMAGE_SECTION_HEADER .rsrc
 - IMAGE_SECTION_HEADER .reloc
- BOUND_IMPORT Directory Table
- BOUND_IMPORT DLL Names
- SECTION .text
- SECTION .data
- SECTION .rsrc
- SECTION .reloc
 - IMAGE_BASE_RELOCATION

RVA	Data	Description	Value
0002F000	00001000	RVA of Block	
0002F004	00000150	Size of Block	
0002F008	3420	Type RVA	00001420 IMAGE_REL_BASED_HIGHLOW
0002F00A	342D	Type RVA	0000142D IMAGE_REL_BASED_HIGHLOW
0002F00C	3436	Type RVA	00001436 IMAGE_REL_BASED_HIGHLOW
0002F00E	3461	Type RVA	00001461 IMAGE_REL_BASED_HIGHLOW
0002F010	3467	Type RVA	00001467 IMAGE_REL_BASED_HIGHLOW
0002F012	3475	Type RVA	00001475 IMAGE_REL_BASED_HIGHLOW
0002F014	347B	Type RVA	0000147B IMAGE_REL_BASED_HIGHLOW
0002F016	349D	Type RVA	0000149D IMAGE_REL_BASED_HIGHLOW
0002F018	34AF	Type RVA	000014AF IMAGE_REL_BASED_HIGHLOW
0002F01A	34B5	Type RVA	000014B5 IMAGE_REL_BASED_HIGHLOW
0002F01C	34BB	Type RVA	000014BB IMAGE_REL_BASED_HIGHLOW
0002F01E	34C9	Type RVA	000014C9 IMAGE_REL_BASED_HIGHLOW
0002F020	34D3	Type RVA	000014D3 IMAGE_REL_BASED_HIGHLOW
0002F022	353B	Type RVA	0000153B IMAGE_REL_BASED_HIGHLOW
0002F024	3551	Type RVA	00001551 IMAGE_REL_BASED_HIGHLOW
0002F026	3566	Type RVA	00001566 IMAGE_REL_BASED_HIGHLOW
0002F028	35B7	Type RVA	000015B7 IMAGE_REL_BASED_HIGHLOW
0002F02A	35F3	Type RVA	000015F3 IMAGE_REL_BASED_HIGHLOW
0002F02C	3624	Type RVA	00001624 IMAGE_REL_BASED_HIGHLOW
0002F02E	363A	Type RVA	0000163A IMAGE_REL_BASED_HIGHLOW
0002F030	3670	Type RVA	00001670 IMAGE_REL_BASED_HIGHLOW
0002F032	367F	Type RVA	0000167F IMAGE_REL_BASED_HIGHLOW
0002F034	368F	Type RVA	0000168F IMAGE_REL_BASED_HIGHLOW
0002F036	36B2	Type RVA	000016B2 IMAGE_REL_BASED_HIGHLOW
0002F038	36B8	Type RVA	000016B8 IMAGE_REL_BASED_HIGHLOW
0002F03A	36E3	Type RVA	000016E3 IMAGE_REL_BASED_HIGHLOW
0002F03C	36FD	Type RVA	000016FD IMAGE_REL_BASED_HIGHLOW

Viewing IMAGE_BASE_RELOCATION

Part 3 PE 재배치 동작 원리 - IMAGE BASE RELOCATION

```
typedef struct _IMAGE_BASE_RELOCATION {  
    DWORD VirtualAddress; 각 블록들의 기준 주소. RVA값.  
    DWORD SizeOfBlock; 각 단위 블록의 크기.  
    // WORD TypeOffset[1]; 이 구조체 밑으로 WORD 타입의 배열이 따라온다는 뜻. 프로그램에 하드코딩된 주소들의 offset.  
} IMAGE_BASE_RELOCATION;  
  
typedef IMAGE_BASE_RELOCATION UNALIGNED * PIMAGE_BASE_RELOCATION;  
  
#define IMAGE_REL_BASED_ABSOLUTE          0  
#define IMAGE_REL_BASED_HIGH             1  
#define IMAGE_REL_BASED_LOW              2  
#define IMAGE_REL_BASED_HIGHLOW          3  
#define IMAGE_REL_BASED_HIGHADJ          4  
#define IMAGE_REL_BASED_MIPS_JMPADDR      5  
#define IMAGE_REL_BASED_MIPS_JMPADDR16   9  
#define IMAGE_REL_BASED_IA64_IMM64        9  
#define IMAGE_REL_BASED_DIR64            10
```

Part 3 PE 재배치 동작 원리 -Base Relocation Table의 해석 방법

RVA	Data	Description	Value
0002F000	00001000	RVA of Block	기준 주소
0002F004	00000150	Size of Block	블록의 사이즈
0002F008	3420	Type RVA	00001420 IMAGE_REL_BASED_HIGHLOW
0002F00A	342D	Type RVA	0000142D IMAGE_REL_BASED_HIGHLOW
0002F00C	3436	Type RVA	00001436 IMAGE_REL_BASED_HIGHLOW
0002F00E	3461	Type RVA	00001461 IMAGE_REL_BASED_HIGHLOW

Typeoffset 값

Type(4비트)	Offset(12비트)
3	420

```
#define IMAGE_REL_BASED_HIGHLOW 3
```


PE 재배포치 동작 실습

Part 4 PE 재배치 동작 실습

002F369F	. 895D FC	MOV DWORD PTR SS:[EBP-4],EBX	
002F36A2	. 8D45 98	LEA EAX,DWORD PTR SS:[EBP-68]	
002F36A5	. 50	PUSH EAX	
002F36A6	. FF15 FC102F00	CALL DWORD PTR DS:[<&KERNEL32.GetStartupInfoA>]	[pStartupinfo GetStartupInfoA
002F36AC	. C745 FC FEFFFF	MOV DWORD PTR SS:[EBP-4],-2	
002F36B3	. C745 FC 010000	MOV DWORD PTR SS:[EBP-4],1	
002F36BA	. 64:A1 18000000	MOV EAX,DWORD PTR FS:[18]	
002F36C0	. 8B70 04	MOV ESI,DWORD PTR DS:[EAX+4]	

Part 4 PE 재배치 동작 실습

RVA	Data	Description	Value
0002F362	3691	Type RVA	00003691 IMAGE_REL_BASED_HIGHLOW
0002F364	36A8	Type RVA	000036A8 IMAGE_REL_BASED_HIGHLOW
0002F366	36C4	Type RVA	000036C4 IMAGE_REL_BASED_HIGHLOW
0002F368	36CE	Type RVA	000036CE IMAGE_REL_BASED_HIGHLOW
0002F36A	36DE	Type RVA	000036DE IMAGE_REL_BASED_HIGHLOW
0002F36C	36EB	Type RVA	000036EB IMAGE_REL_BASED_HIGHLOW

Part 4 PE 재배치 동작 실습

RVA	Raw Data	Value
00003680	EA 5E 5D C3 90 90 90 90 90 E8 C5 F9 FF FF 6A 58	. ^] j X
00003690	68 A0 37 00 01 E8 72 04 00 00 33 DB 89 5D E4 89	h . 7 . . . r . . . 3 . .] . .
000036A0	5D FC 8D 45 98 50 FF 15 FC 10 00 01 C7 45 FC FE] . . E . P E . .
000036B0	FF FF FF C7 45 FC 01 00 00 00 64 A1 18 00 00 00 E d
000036C0	8B 70 04 BF 5C C2 00 01 6A 00 56 57 FF 15 00 11	. p . . \ . . . j . W
000036D0	00 01 85 C0 0F 85 36 35 00 00 33 F6 46 A1 A4 C0 65 . . 3 . F . . .
000036E0	00 01 3B C6 0F 84 44 35 00 00 A1 A4 C0 00 01 85	. . ; . . . D5

$$010010FC - 01000000 + 002F0000 =$$

(하드코딩된 주소 - ImageBase + 실제 로딩 주소)

002F36A6 . FF15 FC102F00 CALL DWORD PTR DS:[<&KERNEL32.GetStartupInfoA>] LGetStartupInfoA

Q&A