리버스 엔지니어링 바이블

ch04. DLL 분석

저자 : 강병탁

김영철

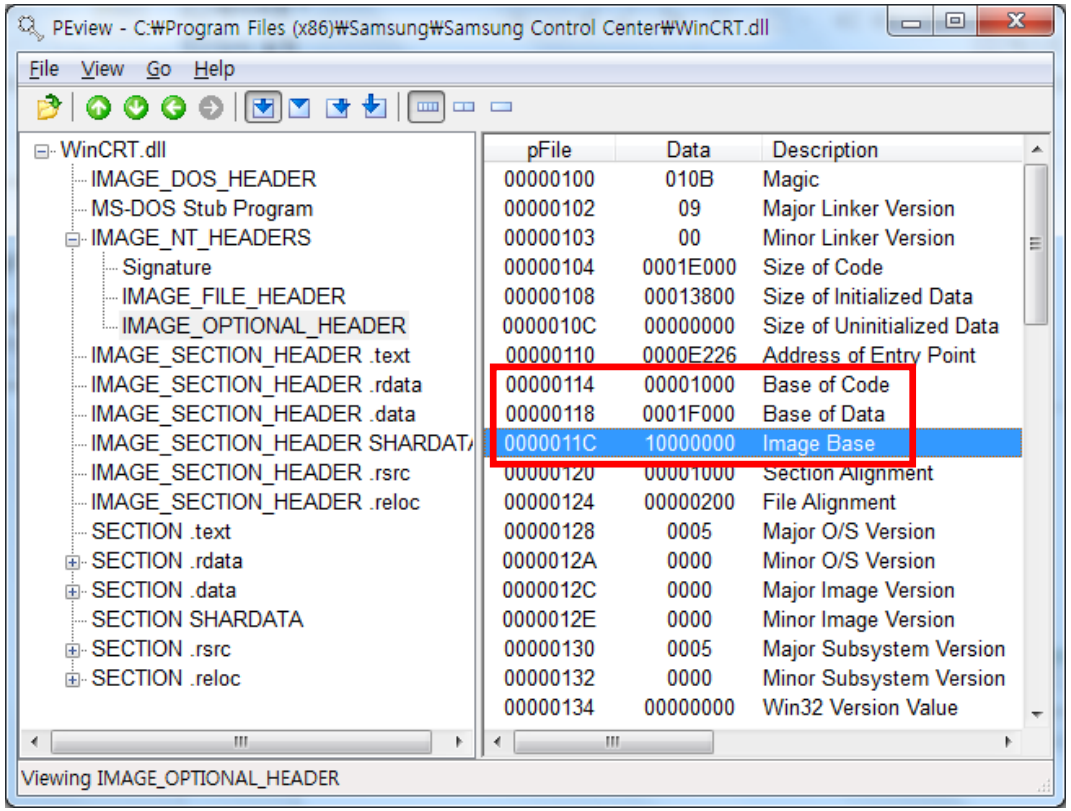
2015. 11. 11.

DLL

- Dynamic Link Library
  : 동적으로 로드 됨.
  : 메모리 절약

DLL 번지 계산법

- 분석 도구로 확인한 Image Base




```
.text:10001000 ; File Name   : C:₩Users₩Areee₩Desktop₩WinCRT.dll
.text:10001000 ; Format      : Portable executable for 80386 (PE)
.text:10001000 ; Imagebase   : 10000000
.text:10001000 ; Section 1. (virtual address 00001000)
.text:10001000 ; Virtual size                  : 0001DE1B ( 122395.)
.text:10001000 ; Section size in file          : 0001E000 ( 122880.)
.text:10001000 ; Offset to raw data for section: 00000400
.text:10001000 ; Flags 60000020: Text Executable Readable
.text:10001000 ; Alignment     : default
.text:10001000 ; ===========================================================
.text:10001000
.text:10001000 ; Segment type: Pure code
.text:10001000 ; Segment permissions: Read/Execute
.text:10001000 _text           segment para public 'CODE' use32
.text:10001000                 assume cs:_text
.text:10001000                 ;org 10001000h
.text:10001000                 assume es:nothing, ss:nothing, ds:_data,
.text:10001000
```

DLL 번지 계산법

- 메모리에 실제로 올라갔을 경우

Executable modules

Base	Size	Entry	Name	File version	Path
[redacted]					
10000000	00039000	1000E226	WinCRT	0, 0, 0, 1	C:₩Program Files (x86)₩Samsung₩Samsung Control Center₩WinCRT.dll
[redacted]					
6EB00000	0008C000	6EB0F118	imkrtip	8,1,7600,16385	C:₩windows₩system32₩ime₩imekr8₩imkrtip.dll
6F1F0000	0005E000	6F1F6807	imetip	10,1,7601,17514	C:₩windows₩system32₩ime₩shared₩imetip.dll
6FF60000	00020000	6FF71E95	imkrapi	8,1,7600,16385	C:₩windows₩system32₩ime₩imekr8₩imkrapi.dll
70460000	00013000	70461858	dwmapi	6,1,7600,16385	C:₩windows₩system32₩dwmapi.dll
746C0000	00009000	746C1220	VERSION	6,1,7600,16385	C:₩windows₩system32₩VERSION.dll
74770000	00080000	747837C9	uxtheme	6,1,7600,16385	C:₩windows₩system32₩uxtheme.dll
74860000	0019E000	7488E6C9	COMCTL32	6,10 (win7_rtm,	C:₩windows₩WinSxS₩x86_microsoft.windows.common-controls_6595b641
74A00000	00032000	74A07F4F	nvinit	8,17,12,6754	C:₩windows₩SysWOW64₩nvinit.dll
74AF0000	00051000	74B1988C	WINSPOOL	6,1,7600,16385	C:₩windows₩system32₩WINSPOOL.DRV
74C10000	0000C000	74C110E1	CRYPTBAS	6,1,7601,19018	C:₩windows₩syswow64₩CRYPTBASE.dll
74C20000	00060000	74C3A3B3	SspiCli	6,1,7601,19007	C:₩windows₩syswow64₩SspiCli.dll
74CD0000	000AC000	74CDA472	msvcrt	7,0,7601,17744	C:₩windows₩syswow64₩msvcrt.dll
74D80000	00047000	74D874C1	KERNELBA	6,1,7601,18015	C:₩windows₩syswow64₩KERNELBASE.dll
74E00000	00060000	74E1158F	IMM32	6,1,7601,17514	C:₩windows₩system32₩IMM32.DLL
74E60000	0009D000	74E940EA	USP10	1,0626,7601,184	C:₩windows₩syswow64₩USP10.dll
74F40000	00057000	74F59BA6	SHLWAPI	6,1,7600,16385	C:₩windows₩syswow64₩SHLWAPI.dll
74FB0000	000A1000	74FC494D	ADVAPI32	6,1,7601,18869	C:₩windows₩syswow64₩ADVAPI32.dll
75060000	00083000	750623D2	CLBCatQ	2001,12,8530,16	C:₩windows₩syswow64₩CLBCatQ.DLL
750F0000	00110000	75103283	kernel32	6,1,7601,18015	C:₩windows₩syswow64₩kernel32.dll
75200000	0015C000	7524BA3D	ole32	6,1,7600,16385	C:₩windows₩syswow64₩ole32.dll
75770000	0008F000	75773FB1	OLEAUT32	6,1,7601,18679	C:₩windows₩syswow64₩OLEAUT32.dll
75800000	00090000	7581633B	GDI32	6,1,7601,18898	C:₩windows₩syswow64₩GDI32.dll

DLL 번지 계산법

- 이미 다른 dll이 올라가 있는 경우

Executable modules

Base	Size	Entry	Name	File version	Path
009C0000	00039000	009CE226	WinCRT2	0, 0, 0, 1	C:₩Users₩Aree₩Desktop₩DLL 실습₩WinCRT2.dll
[illegible]	[illegible]	[illegible]	notepad	[illegible]	[illegible]
10000000	00039000	1000E226	WinCRT	0, 0, 0, 1	C:₩Program Files (x86)₩Samsung₩Samsung Control Center₩WinCRT.dll
6EAD0000	00023000	6EADFCC2	imjkapi	10,1,7600,16385	C:₩windows₩system32₩ime₩shared₩imjkapi.dll
6EB00000	0008C000	6EB0F118	imkrtip	8,1,7600,16385	C:₩windows₩system32₩ime₩imekr8₩imkrtip.dll
6F1F0000	0005E000	6F1F6807	imetip	10,1,7601,17514	C:₩windows₩system32₩ime₩shared₩imetip.dll
6FF60000	00020000	6FF71E95	imkrapi	8,1,7600,16385	C:₩windows₩system32₩ime₩imekr8₩imkrapi.dll
70460000	00013000	70461858	dwmapi	6,1,7600,16385	C:₩windows₩system32₩dwmapi.dll
746C0000	00009000	746C1220	VERSION	6,1,7600,16385	C:₩windows₩system32₩VERSION.dll
74770000	00080000	747837C9	uxtheme	6,1,7600,16385	C:₩windows₩system32₩uxtheme.dll
74860000	0019E000	7488E6C9	COMCTL32	6,10 (win7_rtm,	C:₩windows₩WinSxS₩x86_microsoft.windows.common-controls_6595b641
74A00000	00032000	74A07F4F	nvinit	8,17,12,6754	C:₩windows₩SysWOW64₩nvinit.dll
74AF0000	00051000	74B1988C	WINSPOOL	6,1,7600,16385	C:₩windows₩system32₩WINSPOOL.DRV
74C10000	0000C000	74C110E1	CRYPTBAS	6,1,7601,19018	C:₩windows₩syswow64₩CRYPTBASE.dll
74C20000	00060000	74C3A3B3	SspiCli	6,1,7601,19007	C:₩windows₩syswow64₩SspiCli.dll
74CD0000	000AC000	74CDA472	msvcrt	7,0,7601,17744	C:₩windows₩syswow64₩msvcrt.dll
74D80000	00047000	74D874C1	KERNELBA	6,1,7601,18015	C:₩windows₩syswow64₩KERNELBASE.dll
74E00000	00060000	74E1158F	IMM32	6,1,7601,17514	C:₩windows₩system32₩IMM32.DLL
74E60000	0009D000	74E940EA	USP10	1,0626,7601,184	C:₩windows₩syswow64₩USP10.dll
74F40000	00057000	74F59BA6	SHLWAPI	6,1,7600,16385	C:₩windows₩syswow64₩SHLWAPI.dll
74FB0000	000A1000	74FC494D	ADVAPI32	6,1,7601,18869	C:₩windows₩syswow64₩ADVAPI32.dll
75060000	00083000	750623D2	CLBCatQ	2001,12,8530,16	C:₩windows₩syswow64₩CLBCatQ.DLL
750F0000	00110000	75103283	kernel32	6,1,7601,18015	C:₩windows₩syswow64₩kernel32.dll
75200000	0015C000	7524BA3D	ole32	6,1,7600,16385	C:₩windows₩syswow64₩ole32.dll
75770000	0008F000	75773FB1	OLEAUT32	6,1,7601,18679	C:₩windows₩syswow64₩OLEAUT32.dll

DLL 번지 계산법

- WinCRT.dll Base : 0x10000000 Base of Code : 0x1000
- WinCRT2.dll Base : 0x009c0000

0x10001000 == 0x009c1000

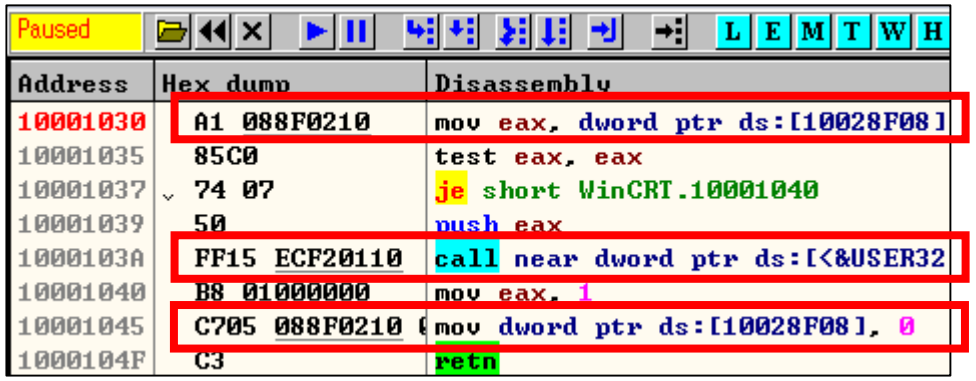
```
.text:10001000 sub_10001000    proc near               ; CODE XR
.text:10001000                                         ; DATA XR
.text:10001000                 mov     eax, offset off_1001F450
.text:10001005                 retn
.text:10001005 sub_10001000    endp
.text:10001005
.text:10001005 ; ---------------------------------------
.text:10001006                 align 10h
```

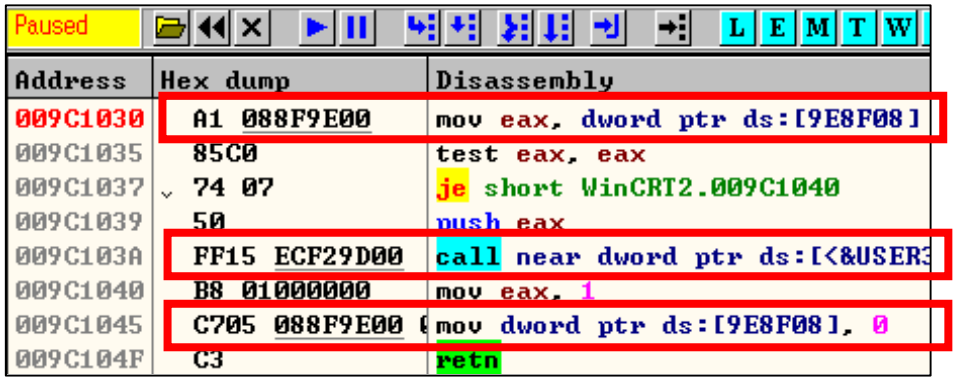
Paused

CPU - thread 00001564, module WinCRT2

Address	Hex dump	Disassembly
009C1000	B8 50F49D00	mov eax, WinCRT2.009DF450
009C1005	C3	retn
009C1006	CC	int3
009C1007	CC	int3
009C1008	CC	int3
009C1009	CC	int3
009C100A	CC	int3
009C100B	CC	int3
009C100C	CC	int3
009C100D	CC	int3
009C100E	CC	int3
009C100F	CC	int3
009C1010	56	push esi
009C1011	6A 00	push 0

재배치를 고려한 방법

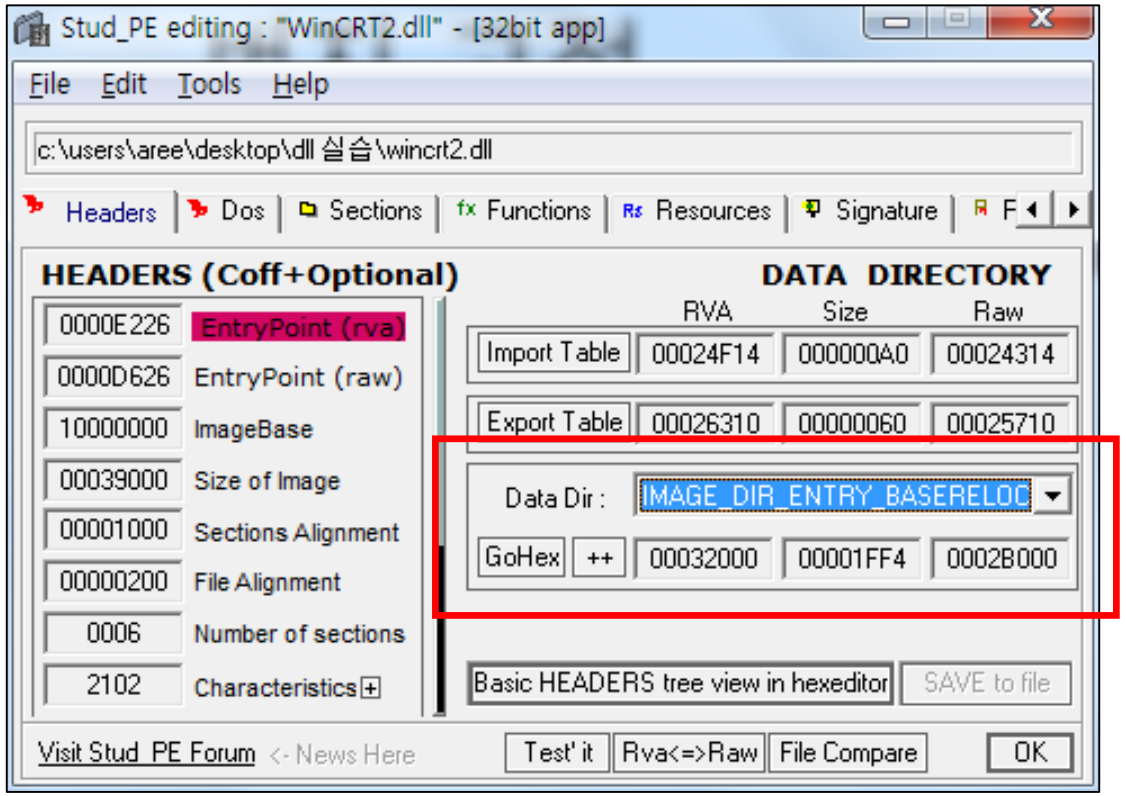




10001030	A1	088F0210	mov eax, dword ptr ds:[10028f08]
009C1030	A1	088F9E00	mov eax, dword ptr ds:[9e8f08]
1000103A	FF15	FCF20110	call near dword ptr ds:[...]
009C103A	FF15	FCF29D00	call near dword ptr ds:[...]

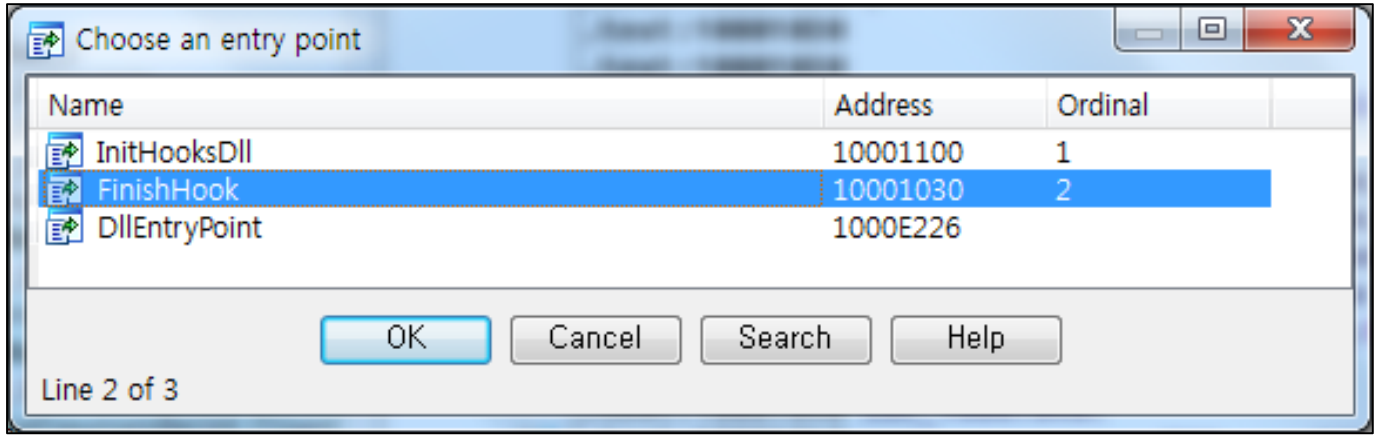
번지 고정

#pragma comment(linker, "/base:0x10000000 /fixed")



Export 함수

- Export 함수 : 외부에서 호출 가능한 함수.



➔ 여러 분석 도구에서 함수의 이름과 주소를 알 수 있음.

DllAttach/DllDetach 찾기

- DllMain() 함수를 리버싱 해야 할 경우.

```
BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
```

➔ fdwReason 값을 이용한 switch-case문이 존재

DLL_PROCESS_ATTACH

DLL_PROCESS_DETACH

DLL_THREAD_ATTACH

DLL_THREAD_DETACH

➔ 4개의 값이 존재

DllAttach/DllDetach 찾기

- 아래와 같은 DllMain()함수를 가진 코드 컴파일.

```
BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    HANDLE hThread = NULL;

    g_hMod = (HMODULE)hinstDLL;

    switch( fdwReason )
    {
    case DLL_PROCESS_ATTACH :
        OutputDebugString(L"<myhack.dll> Injection!!!");
        hThread = CreateThread(NULL, 0, ThreadProc, NULL, 0, NULL);
        CloseHandle(hThread);
        break;
    }

    return TRUE;
}
```

DllAttach/DllDetach 찾기

- 디스어셈블러로 분석

```
                push    ebp
                mov     ebp, esp
                mov     eax, [ebp+hinstDLL]
                mov     hModule, eax
                mov     eax, [ebp+fdwReason]
                dec     eax
                jnz     short loc_100010E8
                push    offset OutputString ; "<myhack.dll> Injection!!!"
                call    ds:OutputDebugStringW
                push    0               ; lpThreadId
                push    0               ; dwCreationFlags
                push    0               ; lpParameter
                push    offset StartAddress ; lpStartAddress
                push    0               ; dwStackSize
                push    0               ; lpThreadAttributes
                call    ds:CreateThread
                push    eax             ; hObject
                call    ds:CloseHandle

loc_100010E8:                           ; CODE XREF: DllMain(x,x,x)+F↑j
                mov     eax, 1
                pop     ebp
                retn    0Ch
```

hinstDLL == dword ptr 0x8
fdwReason == dword ptr 0xc
lpvReserved == dword ptr 0x10

fdwReason == 1
OutputDebugStringW();
CreateThread();
CloseHandle();

DllAttach/DllDetach 찾기

- 디스어셈블러로 분석

```
                push    ebp
                mov     ebp, esp
                mov     eax, [ebp+hinstDLL]
                mov     hModule, eax
                mov     eax, [ebp+fdwReason]
                dec     eax
                jnz     short loc_100010E8
                push    offset OutputString ; "<myhack.dll> Injection!!!"
                call    ds:OutputDebugStringW
                push    0               ; lpThreadId
                push    0               ; dwCreationFlags
                push    0               ; lpParameter
                push    offset StartAddress ; lpStartAddress
                push    0               ; dwStackSize
                push    0               ; lpThreadAttributes
                call    ds:CreateThread
                push    eax             ; hObject
                call    ds:CloseHandle

loc_100010E8:                           ; CODE XREF: DllMain(x,x,x)+F↑j
                mov     eax, 1
                pop     ebp
                retn    0Ch
```

DllAtach되면 호출되는 부분
DLL_PROCESS_ATTACH == 1

패킹된 DLL의 DllMain() 찾기

- DllMain()함수를 작성 할 때, 반드시 생성되는 패턴 이용.

```
                push    ebp
                mov     ebp, esp
                mov     eax, [ebp+hinstDLL]
                mov     hModule, eax
                mov     eax, [ebp+fdwReason]
                dec     eax
                jnz     short loc_100010E8
                push    offset OutputString ; "<myhack.dll> Injection!!!"
                call    ds:OutputDebugStringW
                push    0               ; lpThreadId
                push    0               ; dwCreationFlags
                push    0               ; lpParameter
                push    offset StartAddress ; lpStartAddress
                push    0               ; dwStackSize
                push    0               ; lpThreadAttributes
                call    ds:CreateThread
                push    eax             ; hObject
                call    ds:CloseHandle

loc_100010E8:                           ; CODE XREF: DllMain(x,x,x)+F↑j
                mov     eax, 1
                pop     ebp
                retn    0Ch
```

Hex 값으로 8b 45 0c 48

```
Paused
Address   Hex dump     Disassembly
005210BB  8B45 0C      mov eax, dword ptr ss:[ebp+C]
005210BE  48           dec eax
005210BF  75 27        jnz short myhack.005210E8
005210C1  68 78785200  push myhack.00527878
005210C6  FF15 00605200 call near dword ptr ds:[<&KERN
```

DisableThreadLibraryCalls() 함수

- DLL_THREAD_ATTACH
  ➔ 새로운 쓰레드가 생성되면 DllMain()을 호출할 때 fdwReason에 전달하는 값

- DLL_THREAD_DETACH
  ➔ 쓰레드가 종료되기 전 DllMain()을 호출할 때 fdwReason에 전달하는 값

- 쓰레드가 생성되거나 종료될 때 DllMain()이 호출되지 않게 함.