Manual Unpack By Debugger

2012-12-01

A-FIRST
고흥환 책임연구원

AhnLab

Contents

Packer

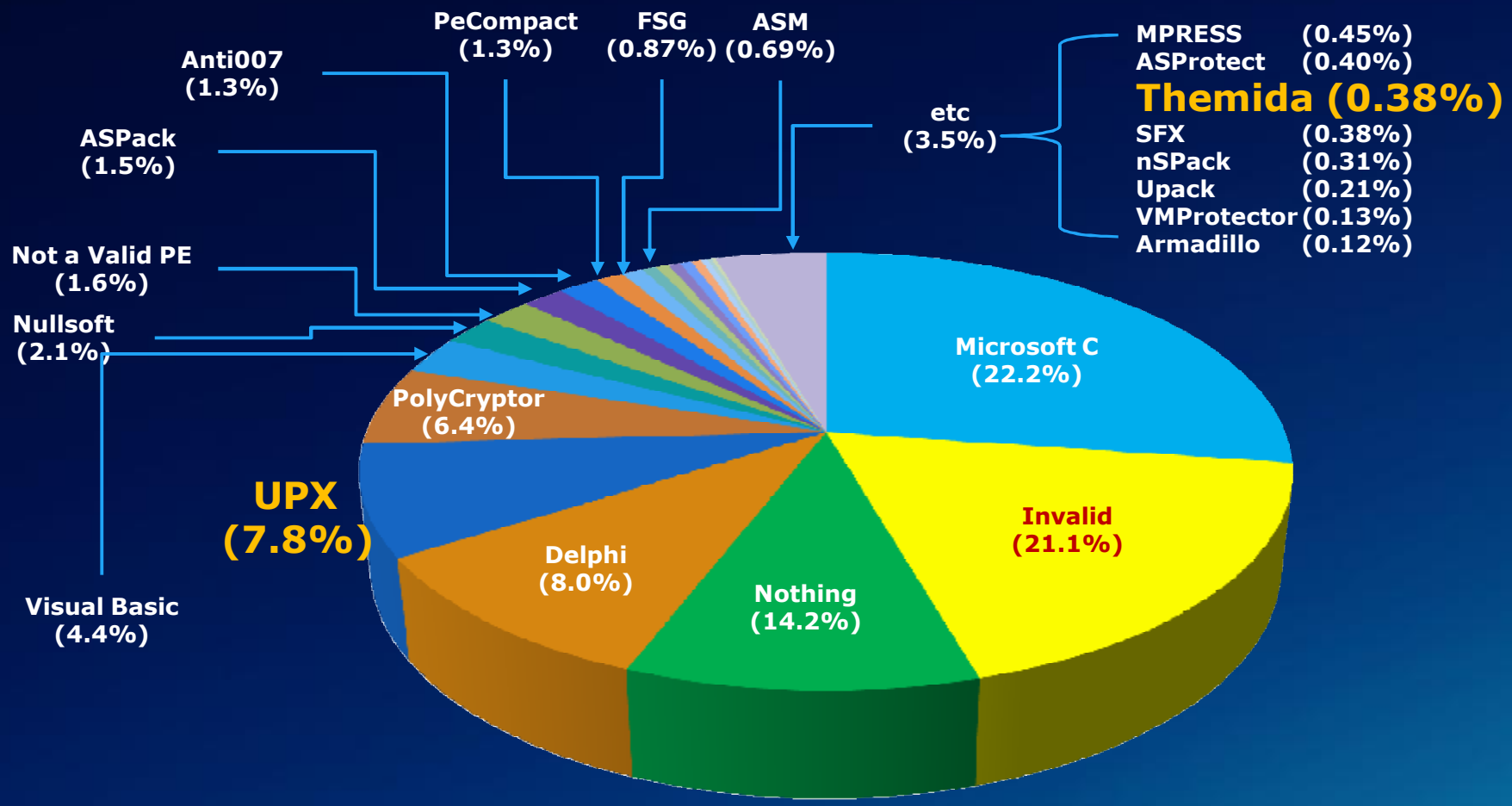
Executable compression
= Runtime Packer
= Packer

is any means of compressing an executable file and combining the compressed data with decompression code into a single executable.

I. Encryption
II. Compression
III. Redirection
IV. Substitution
V. Obfuscation
VI. Polymorphism
VII. Metamorphism
VIII. Protection
IX. Virtualization

Name	Latest stable	Software license	x86-64 support
.netshrink	2.3 (March 29, 2012 (2012-03-29))[1]	Proprietary	Yes
Armadillo Packer	8.60 (July 6, 2011 (2011-07-06))	Proprietary	Yes
ASPack	2.29 (August 3, 2011 (2011-08-03))	Proprietary	?
ASPR (ASProtect)	1.64 (September 1, 2011 (2011-09-01))	Proprietary	?
BoxedApp Packer	2.2 (June 16, 2009 (2009-06-16))[2]	Proprietary	Yes
CExe	1.0b (July 20, 2001 (2001-07-20))	GPL	No
Enigma Protector	3.80 (August 2, 2012 (2012-08-02))[3]	Proprietary	Yes
EXE Bundle	3.11 (January 7, 2011 (2011-01-07))[4]	Proprietary	?
EXE Stealth	4.14 (June 29, 2011 (2011-06-29))[5]	Proprietary	?
eXPressor	1.8.0.1 (January 14, 2010 (2010-01-14))	Proprietary	?
MPRESS	2.19 (January 2, 2012 (2012-01-02))	Freeware	Yes
Obsidium	1.4.6 (July 18, 2012 (2012-07-18))[6]	Proprietary	Yes
PELock	1.0.694 (January 23, 2012 (2012-01-23))[7]	Proprietary	No
PESpin	1.33 (May 3, 2011 (2011-05-03))	Freeware	Yes
RLPack Basic	1.21 (October 31, 2008 (2008-10-31))	GPL	No
Smart Packer Pro	1.7 (November 5, 2011 (2011-11-05))	Proprietary	Yes
Themida	2.2.1.0 (July 25, 2012 (2012-07-25))	Proprietary	?
UPX	3.08 (December 12, 2011 (2011-12-12))	GPL	No
VMProtect	2.1 (September 26, 2011 (2011-09-26))	Proprietary	Yes
XComp/XPack	0.98 (February 18, 2007 (2007-02-18))	Freeware	No

Themida & UPX



2011 AhnLab 10,000,000 파일 대상

Debugger Detection

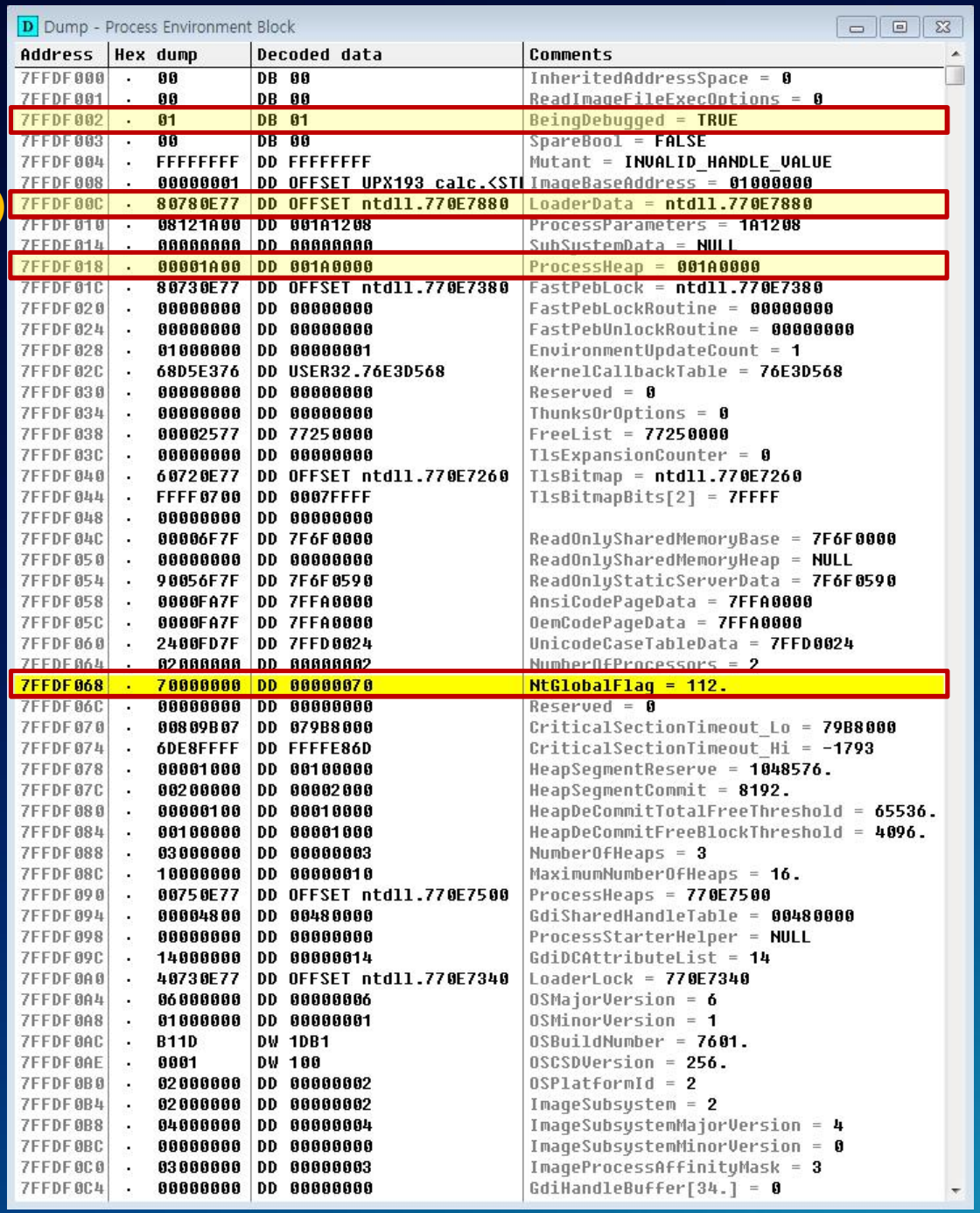
BeingDebugged (PEB+0x2)

PEB_LDR_DATA(PEB+0x0C)

ProcessHeap (PEB+0x18)

- **Flags(ProcessHeap+0x0C)**
- **ForceFlags (ProcessHeap+0x10)**

NtGlobalFlag (PEB+0x68)



IsDebuggerPresent()

```
$  64:A1 18000000    MOV EAX,DWORD PTR FS:[18]
.  8B40 30           MOV EAX,DWORD PTR DS:[EAX+30]
.  0FB640 02         MOVZX EAX,BYTE PTR DS:[EAX+2]
.  C3                RETN
```

TEB (Thread Environment Block)

Address	Hex dump	Decoded data	Comments
7EFDD000	C4FF1800	DD 0018FFC4	SEH chain = 18FFC4 -> {Next=FFFFFFFF,Handler=ntdll.77AF71D5}
7EFDD004	00001900	DD 00190000	Thread's stack base = ASCII "Actx "
7EFDD008	00D01800	DD 0018D000	Thread's stack limit = 18D000
7EFDD00C	00000000	DD 00000000	TIB of OS/2 Subsystem = NULL
7EFDD010	001E0000	DD 00001E00	Fiber data = 00001E00
7EFDD014	00000000	DD 00000000	Arbitrary user data = 0
7EFDD018	00D0FD7E	DD 7EFDD000	TIB linear address = 7EFDD000
7EFDD01C	00000000	DD 00000000	Environment pointer = NULL
7EFDD020	7C150000	DD 0000157C	Process ID = 0000157C
7EFDD024	78150000	DD 00001578	Thread ID = 00001578
7EFDD028	00000000	DD 00000000	RPC handle = 00000000
7EFDD02C	2CD0FD7E	DD 7EFDD02C	TLS array = 7EFDD02C
7EFDD030	00E0FD7E	DD 7EFDE000	Process database = 7EFDE000
7EFDD034	B7360000	DD 000036B7	Thread's last error = ERROR_SXS_KEY_NOT_FOUND

PEB (Process Environment Block)

Address	Hex dump	Decoded data	Comments
7EFDE000	00	DB 00	InheritedAddressSpace = 0
7EFDE001	00	DB 00	ReadImageFileExecOptions = 0
7EFDE002	01	DB 01	BeingDebugged = TRUE
7EFDE003	00	DB 00	SpareBool = FALSE
7EFDE004	FFFFFFFF	DD FFFFFFFF	Mutant = INVALID_HANDLE_VALUE
7EFDE008	00004000	DD OFFSET IsDebuggerPresent	ImageBaseAddress = 00400000
7EFDE00C	0002B877	DD OFFSET ntdll.77B80200	LoaderData = ntdll.77B80200
7EFDE010	00175500	DD 00551790	ProcessParameters = 551790
7EFDE014	00000000	DD 00000000	SubSystemData = NULL
7EFDE018	00005500	DD 00550000	ProcessHeap = 00550000

CheckRemoteDebuggerPresent(ProcessId, &bPresent)

```
75D0504E      8BFF            MOV EDI,EDI
75D05050  ┌·  55              PUSH EBP
75D05051  ·   8BEC            MOV EBP,ESP
75D05053  ·   837D 08 00      CMP DWORD PTR SS:[EBP+8],0
75D05057  ·   56              PUSH ESI
75D05058  ·-  74 34           JE SHORT 75D0508E
75D0505A  ·   8B75 0C         MOV ESI,DWORD PTR SS:[EBP+0C]
75D0505D  ·   85F6            TEST ESI,ESI
75D0505F  ·-  74 2D           JZ SHORT 75D0508E
75D05061  ·   6A 00           PUSH 0                          ┌pLength = NULL
75D05063  ·   6A 04           PUSH 4                          │Bufsize = 4
75D05065  ·   8D45 08         LEA EAX,[EBP+8]                 │
75D05068  ·   50              PUSH EAX                        │Buffer
75D05069  ·   6A 07           PUSH 7                          │ProcessInfoClass = ProcessDebugPort
75D0506B  ·   FF75 08         PUSH DWORD PTR SS:[EBP+8]       │ProcessHandle
75D0506E  ·   FF15 4C15CD75   CALL DWORD PTR DS:[<&ntdll.NtQueryInformationProcess>] └NTDLL.ZwQueryInformationProcess
75D05074  ·   85C0            TEST EAX,EAX
75D05076  ·-  0F8C AA580300   JL 75D3A926
75D0507C  ·   33C0            XOR EAX,EAX
75D0507E  ·   3945 08         CMP DWORD PTR SS:[EBP+8],EAX
75D05081  ·   0F95C0          SETNE AL
75D05084  ·   8906            MOV DWORD PTR DS:[ESI],EAX
75D05086  ·   33C0            XOR EAX,EAX
75D05088  ·   40              INC EAX
75D05089  >   5E              POP ESI
75D0508A  ·   5D              POP EBP
75D0508B  ·   C2 0800         RETN 8
75D0508E  >   6A 57           PUSH 57                         ┌ErrCode = ERROR_INVALID_PARAMETER
75D05090  ·   FF15 1016CD75   CALL DWORD PTR DS:[<&ntdll.RtlSetLastWin32Error>] └NTDLL.RtlSetLastWin32Error
75D05096  └-  E9 91580300     JMP 75D3A92C
```

timeGetTime(), GetTickCount(), NtQueryPerformanceCounter(), RDTSC

```
00455353    83C4 04             ADD ESP,4
00455356    8D85 84C63C0C       LEA EAX,[EBP+0C3CC684]
0045535C    68 BF370000         PUSH 37BF
00455361    890424              MOV DWORD PTR SS:[ESP],EAX
00455364    68 601D0000         PUSH 1D60
00455369    891C24              MOV DWORD PTR SS:[ESP],EBX
0045536C    FF95 FB34380C       CALL DWORD PTR SS:[EBP+0C3834FB]    kernel32.GetProcAddress
00455372    8985 D017380C       MOV DWORD PTR SS:[EBP+0C3817D0],EAX
00455378    8D85 DD033D0C       LEA EAX,[EBP+0C3D03DD]
0045537E    8985 5E2A380C       MOV DWORD PTR SS:[EBP+0C382A5E],EAX
00455384    FF95 D017380C       CALL DWORD PTR SS:[EBP+0C3817D0]    winmm.timeGetTime
```

Garbage Codes

timeGetTime()

```
00452B0F    FF95 D017380C       CALL DWORD PTR SS:[EBP+0C3817D0]    winmm.timeGetTime
```

Garbage Codes

```
0045310D    3BD8                CMP EBX,EAX
0045310F    0F84 07290000       JE 00455A1C
```

SEH (Structured Exception Handler)

```
0065DDDD   33C0              XOR EAX,EAX
0065DDDF   E9 0D000000       JMP ARP_Atta.0065DDF1
0065DDE4   0F6451 15         PCMPGTB MM2,QWORD PTR DS:[ECX+15]
0065DDE8   91                XCHG EAX,ECX
0065DDE9   1D 140E2908       SBB EAX,8290E14
0065DDEE   26:3C D9          CMP AL,0D9                  Superfluous prefix
0065DDF1   40                INC EAX
0065DDF2   0F3F              ???                         Unknown command
```

Stack

```
0012FF9C   0012FFE0   Pointer to next SEH record
0012FFA0   0065DE82   SE handler
```

Exception Handler

```
0065DE82   8B4C24 0C             MOV ECX,DWORD PTR SS:[ESP+C]
0065DE86   C781 A4000000 FFFFFFFF MOV DWORD PTR DS:[ECX+A4],-1
0065DE90   8381 B8000000 04       ADD DWORD PTR DS:[ECX+B8],4
0065DE97   33C0                   XOR EAX,EAX
0065DE99   C3                     RETN
```

Exception Handler

```
0065DDF6   64:8F05 00000000   POP DWORD PTR FS:[0]          0012FFE0
0065DDFD   83C4 04            ADD ESP,4
0065DE00   66:8BF9            MOV DI,CX
0065DE03   01B5 35247409      ADD DWORD PTR SS:[EBP+9742435],ESI
0065DE09   83FB FF            CMP EBX,-1
0065DE0C   0F84 43000000      JE ARP_Atta.0065DE55
```

CreateFileA "\\.\SICE"

"\\.\SIWVID"

"\\.\NTICE"

```
00654FDD   FFD0              CALL EAX                                  kernel32.CreateFileA
00654FDF   899D 61337409     MOV DWORD PTR SS:[EBP+9743361],EBX
00654FE5   40                INC EAX
00654FE6   0F85 FF030000     JNZ ARP_Atta.006553EB
```

```
HANDLE WINAPI CreateFile(
        __in        LPCTSTR                 lpFileName,
        __in        DWORD                   dwDesiredAccess,
        __in        DWORD                   dwShareMode,
        __in_opt    LPSECURITY_ATTRIBUTES   lpSecurityAttributes,
        __in        DWORD                   dwCreationDisposition,
        __in        DWORD                   dwFlagsAndAttributes,
        __in_opt    HANDLE                  hTemplateFile
);
```

FindWindow "FilemonClass"

"File Monitor – Sysinternals: www.sysinternals.com"

"Filem"

"DeepFrz"

"PROCMON_WINDOW_CLASS"

"Process Monitor – Sysinternals: www.sysinternals.com"

"PROCEXP"

"RegmonClass"

"Registry Monitor – Sysinternals: www.sysinternals.com"

"18467-41"

"REGMON"

"regsys"

"sysregm"

"PROCMON"

NtQuerySystemInformation “iceext.sys”

“ntice.sys”

“Syser.sys”

“HanOlly.sys”

“extrem.sys”

“FRDTSC.sys”

```
NTSTATUS WINAPI NtQuerySystemInformation(
        _In_    SYSTEM_INFORMATION_CLASS    SystemInformationClass,
        _Inout_ PVOID                       SystemInformation,
        _In_    ULONG                       SystemInformationLength,
        _Out_opt_ PULONG                    ReturnLength
);
```

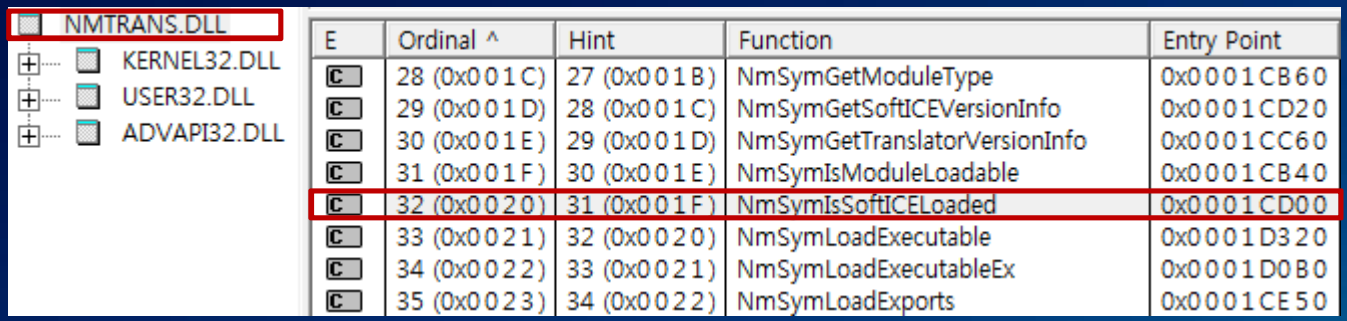
RegOpenKeyA "SOFTWARE\NuMega\DriverStudio"

→ RegQueryValueEx "InstallDir"

→ LoadLibraryA "~\SoftIce\NMTRANS.DLL"

→ GetProcAddress "NmSymIsSoftICELoaded"

→ Call NmSymIsSoftICELoaded



E	Ordinal ^	Hint	Function	Entry Point
C	28 (0x001C)	27 (0x001B)	NmSymGetModuleType	0x0001CB60
C	29 (0x001D)	28 (0x001C)	NmSymGetSoftICEVersionInfo	0x0001CD20
C	30 (0x001E)	29 (0x001D)	NmSymGetTranslatorVersionInfo	0x0001CC60
C	31 (0x001F)	30 (0x001E)	NmSymIsModuleLoadable	0x0001CB40
C	32 (0x0020)	31 (0x001F)	NmSymIsSoftICELoaded	0x0001CD00
C	33 (0x0021)	32 (0x0020)	NmSymLoadExecutable	0x0001D320
C	34 (0x0022)	33 (0x0021)	NmSymLoadExecutableEx	0x0001D0B0
C	35 (0x0023)	34 (0x0022)	NmSymLoadExports	0x0001CE50

Anti Tracing

STI, INT 1

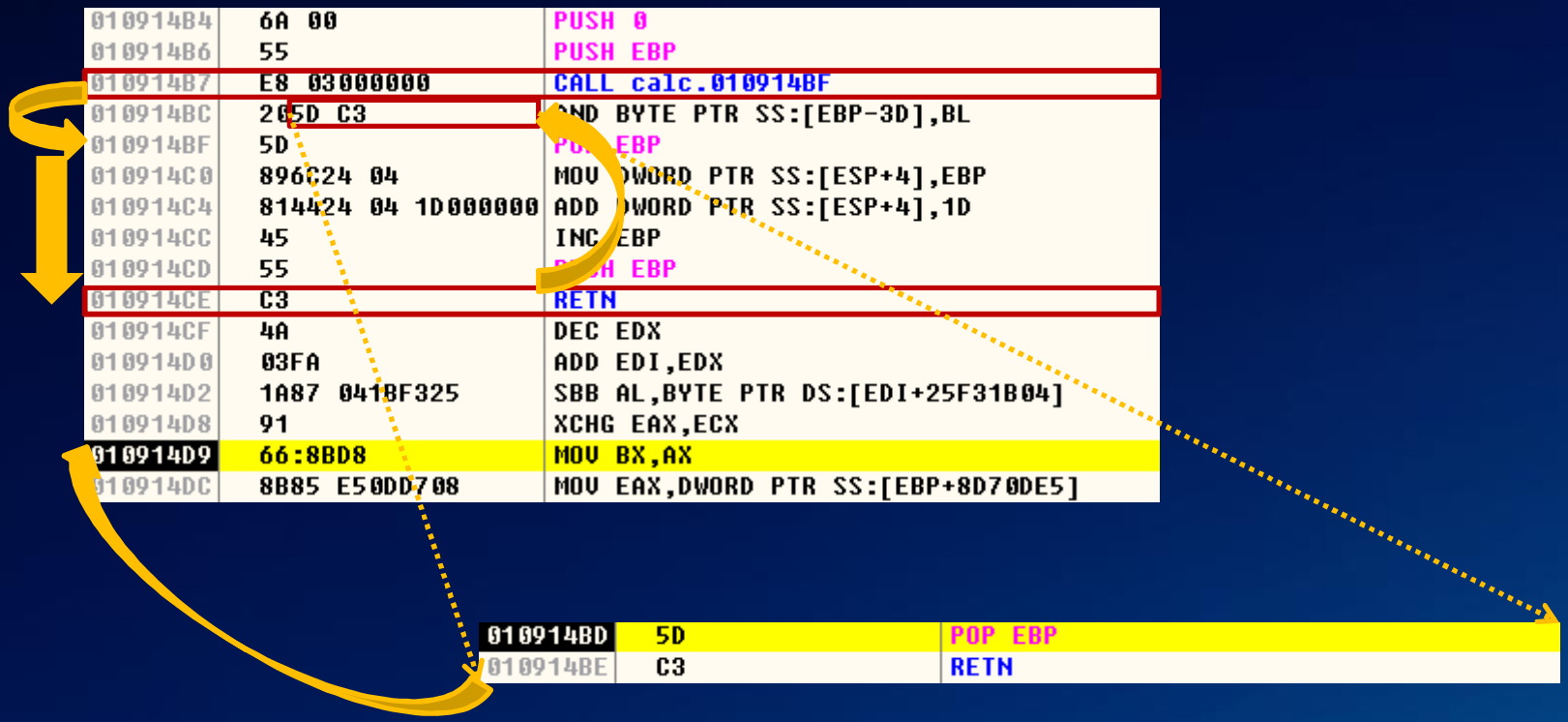
```
0045993A   8B2424          MOV ESP,DWORD PTR SS:[ESP]
0045993D   FB              STI
0045993E   50              PUSH EAX
0045993F   E9 46FFFFFF     JMP 0045988A
```

SetEvent, DelayExecution

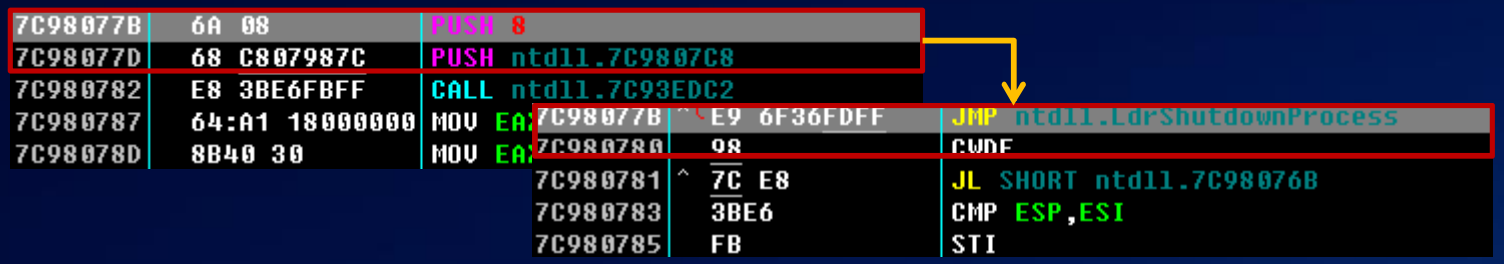
```
010913F6   BD 6665DA08     MOV EBP,8DA6566
010913FB   FF95 9421D708   CALL DWORD PTR SS:[EBP+8D72194]      SetEvent
01091401   89B5 791BD708   MOV DWORD PTR SS:[EBP+8D71B79],ESI
01091407   6A 00           PUSH 0
01091409   FF95 9B21D708   CALL DWORD PTR SS:[EBP+8D7219B]      DelayExecution
0109140F   89C0            MOV EAX,EAX
01091411   EB F4           JMP SHORT calc.01091407
01091413   F1              INT1
```

```
010913F6   BD 6665DA08     MOV EBP,8DA6566
010913FB   FF95 9421D708   CALL DWORD PTR SS:[EBP+8D72194]      SetEvent
01091401   89B5 791BD708   MOV DWORD PTR SS:[EBP+8D71B79],ESI
01091407   6A 00           PUSH 0
01091409   FF95 9B21D708   CALL DWORD PTR SS:[EBP+8D7219B]      DelayExecution
0109140F   89C0            MOV EAX,EAX
01091411   EB 5F           JMP SHORT calc.01091472
01091413   8046 6B 19       ADD BYTE PTR DS:[ESI+6B],19
```

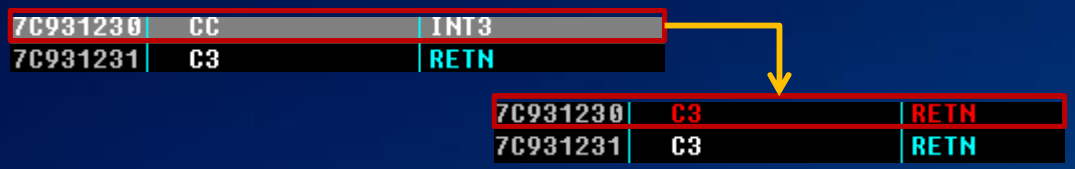
Garbage Code - Linear Sweep Disassembly



DbgUiRemoteBreakin Patch



DbgBreakPoint Patch



Virtual Machine Detection

I. Virtual Machine Artifacts
in Processes, File System, and Registry

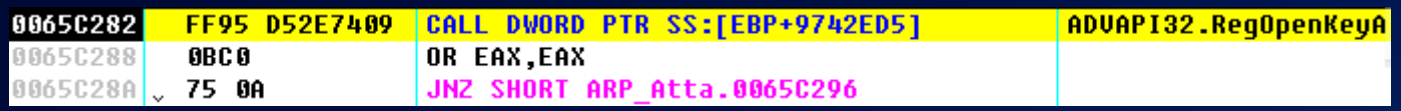
II. Virtual Machine Artifacts
in Memory

III. Virtual Machine Specific Virtual Hardware

IV. Virtual Machine Specific Processor
Instructions and Capabilities

< On the Cutting Edge : Thwarting Virtual Machine Detection 참조 >

RegOpenKeyA “Software\Wine”
"HARDWARE\ACPI\DSDT\VBOX__"




```
LONG WINAPI RegOpenKey(
        __in        HKEY        hKey,
        __in_opt    LPCTSTR     lpSubKey,
        __out       PHKEY       phkResult
);
```

RegOpenKeyA “HARDWARE\DESCRIPTION\System”

↳ RegQueryValueEx “SystemBiosVersion"

Vmware

```
010603FB   B8 68584D56      MOV EAX,564D5868     // Magic Number "VMXh"
01060400   B9 14000000      MOV ECX,14           // BACKDOOR_COMMAND_NUMBER
01060405   66:BA 5856       MOV DX,5658          // Port Number
01060409   ED               IN EAX,DX            // I/O command
```

```
0105F878   B9 0A000000      MOV ECX,0A
0105F87D   B8 04D75548      MOV EAX,4855D704
0105F882   05 6481F70D      ADD EAX,0DF78164
0105F887   BB 65D48586      MOV EBX,8685D465
0105F88C   BA 40B63400      MOV EDX,34B640
0105F891   81EA E85F3400    SUB EDX,345FE8
0105F897   ED               IN EAX,DX            // I/O command
0105F898   81FB 68584D56    CMP EBX,564D5868
0105F89E   75 0A            JNZ SHORT 0105F8AA
```

Manual Unpack UPX 1.9.3

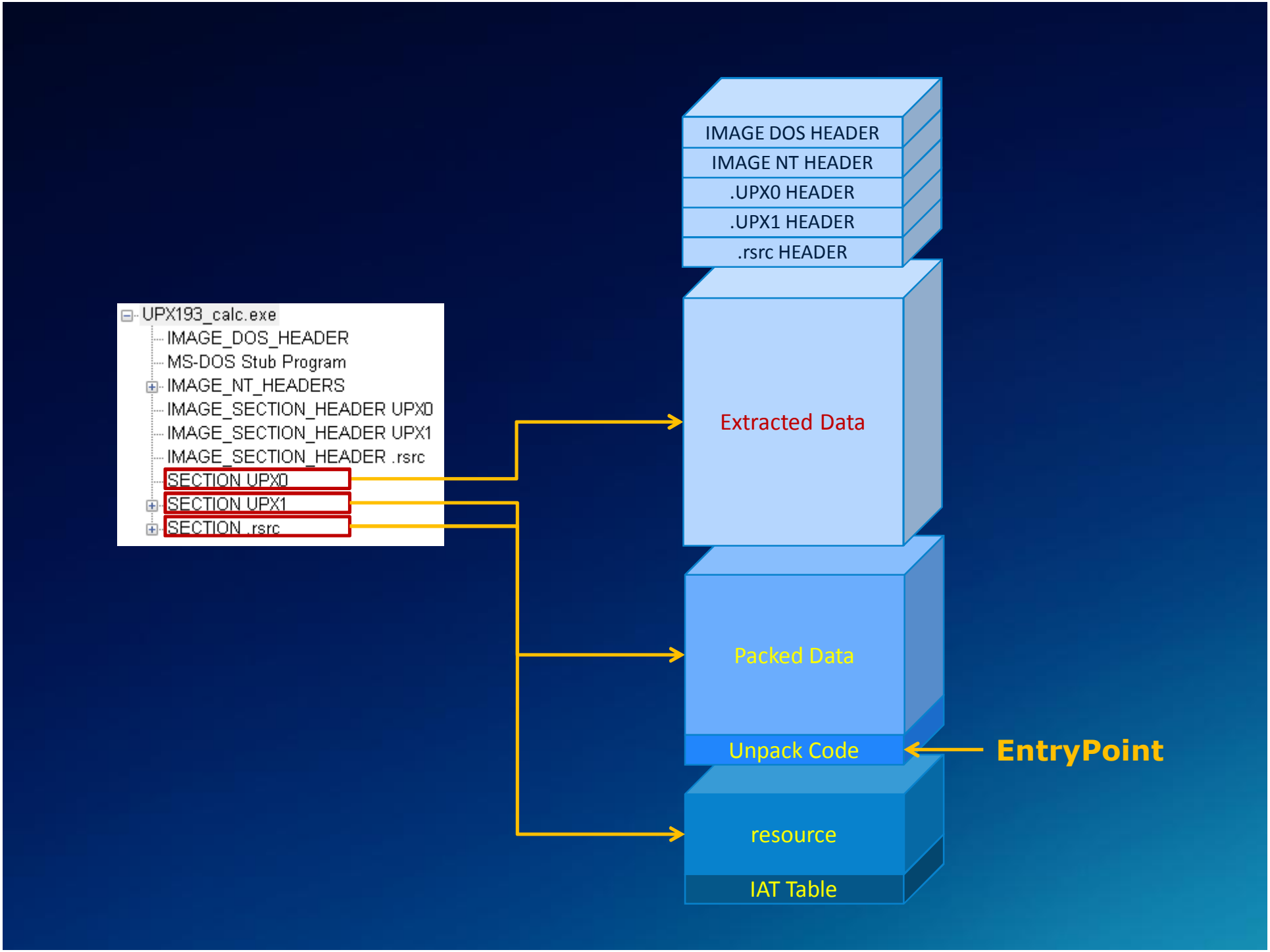
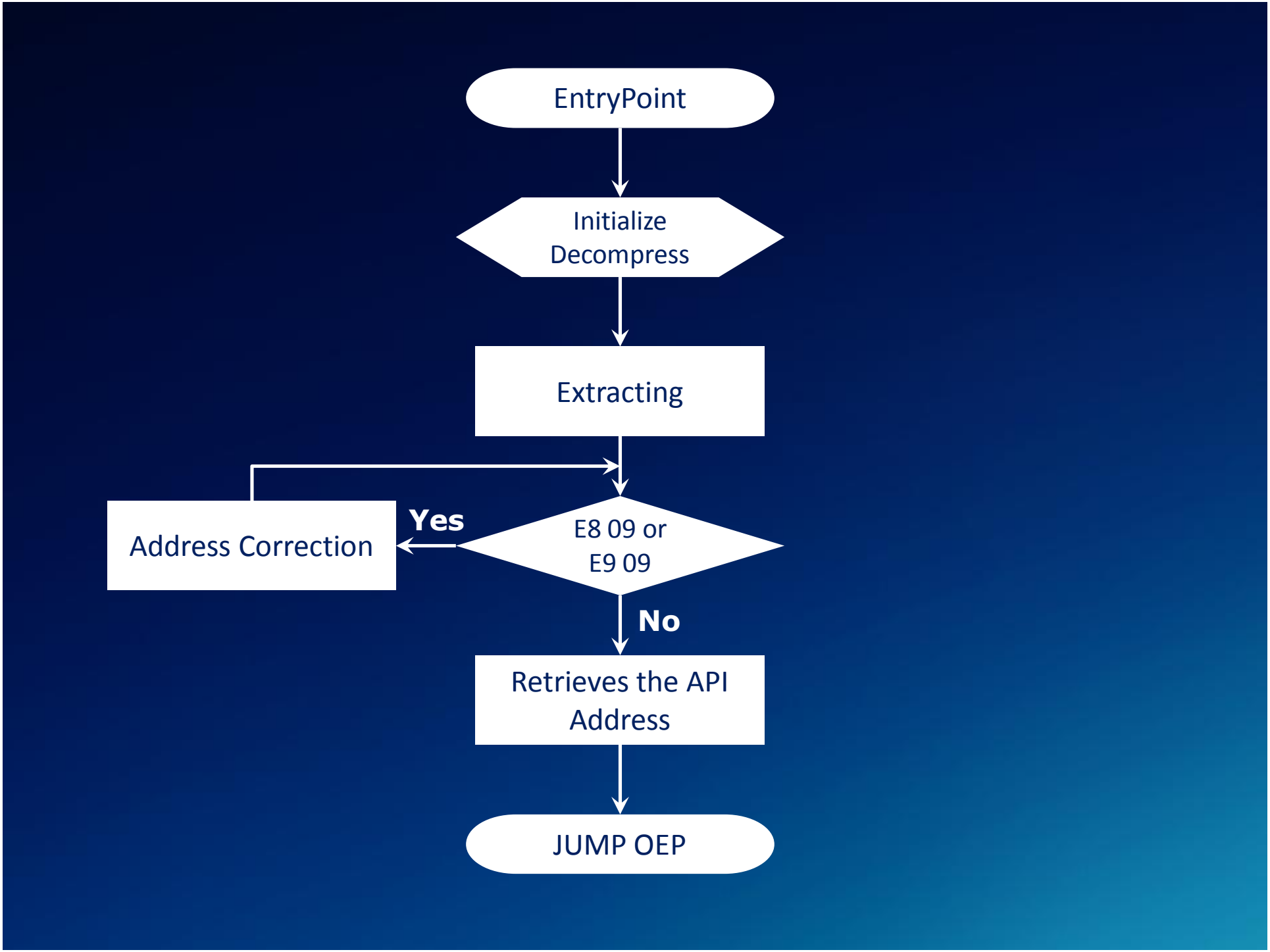


IMAGE DOS HEADER
IMAGE NT HEADER
.UPX0 HEADER
.UPX1 HEADER
.rsrc HEADER
UPX193_calc.exe
IMAGE_DOS_HEADER
MS-DOS Stub Program
IMAGE_NT_HEADERS
IMAGE_SECTION_HEADER UPX0
IMAGE_SECTION_HEADER UPX1
IMAGE_SECTION_HEADER .rsrc
SECTION UPX0
SECTION UPX1
SECTION .rsrc
Extracted Data
Packed Data
Unpack Code
EntryPoint
resource
IAT Table


EntryPoint
Initialize
Decompress
Extracting
E8 09 or
E9 09
Yes
Address Correction
No
Retrieves the API
Address
JUMP OEP

UPX0 – Compressed Data / UPX1 – Decompressed Data

```
01020CD0      60              PUSHAD
01020CD1      BE 00A00101     MOV ESI,0101A000                .UPX1
01020CD6      8DBE 0070FEFF   LEA EDI,[ESI+FFFE7000]         .UPX0
01020CDC      57              PUSH EDI
01020CDD      83CD FF         OR EBP,FFFFFFFF
01020CE0      EB 10           JMP SHORT 01020CF2
```

Extracting Algorithm

```
01020CE8      8A06            MOV AL,BYTE PTR DS:[ESI]
01020CEA      46              INC ESI
01020CEB      8807            MOV BYTE PTR DS:[EDI],AL
01020CED      47              INC EDI
01020CEE      01DB            ADD EBX,EBX
01020CF0      75 07           JNE SHORT 01020CF9
```

...

```
01020D91      8907            MOV DWORD PTR DS:[EDI],EAX
01020D93      83C7 04         ADD EDI,4
01020D96      83E9 04         SUB ECX,4
01020D99      77 F1           JA SHORT 01020D8C
01020D9B      01CF            ADD EDI,ECX
01020D9D      E9 4CFFFFFF     JMP 01020CEE
```

E8 09 (CALL) / E9 09 (JMP) Address Correction

```
01020DA2  >  5E            POP ESI
01020DA3  .  89F7          MOV EDI,ESI
01020DA5  .  B9 4D090000   MOV ECX,94D
01020DAA  >  8A07          MOV AL,BYTE PTR DS:[EDI]
01020DAC  .  47            INC EDI
01020DAD  .  2C E8         SUB AL,0E8
01020DAF  >  3C 01         CMP AL,1
01020DB1  .^ 77 F7         JA SHORT 01020DAA
01020DB3  .  803F 09       CMP BYTE PTR DS:[EDI],9
01020DB6  .^ 75 F2         JNE SHORT 01020DAA
01020DB8  .  8B07          MOV EAX,DWORD PTR DS:[EDI]
01020DBA  .  8A5F 04       MOV BL,BYTE PTR DS:[EDI+4]
01020DBD  .  66:C1E8 08    SHR AX,8
01020DC1  .  C1C0 10       ROL EAX,10
01020DC4  .  86C4          XCHG AH,AL
01020DC6  .  29F8          SUB EAX,EDI
01020DC8  .  80EB E8       SUB BL,0E8
01020DCB  .  01F0          ADD EAX,ESI
01020DCD  .  8907          MOV DWORD PTR DS:[EDI],EAX
01020DCF  .  83C7 05       ADD EDI,5
01020DD2  .  88D8          MOV AL,BL
01020DD4  .^ E2 D9         LOOP SHORT 01020DAF
```

Retrieves the address

```
01020DD6  .   8DBE 00E00100   LEA EDI,[ESI+1E000]
01020DDC  >   8B07            MOV EAX,DWORD PTR DS:[EDI]          ImportDllName 위치값
01020DDE  .   09C0            OR EAX,EAX
01020DE0  .v  74 3C           JE SHORT 01020E1E
01020DE2  .   8B5F 04         MOV EBX,DWORD PTR DS:[EDI+4]
01020DE5  .   8D8430 08680200 LEA EAX,[ESI+EAX+26808]             DLL's Name
01020DEC  .   01F3            ADD EBX,ESI
01020DEE  .   50              PUSH EAX
01020DEF  .   83C7 08         ADD EDI,8
01020DF2  .   FF96 94680200   CALL DWORD PTR DS:[ESI+26894]       LoadLibraryA
01020DF8  .   95              XCHG EAX,EBP
01020DF9  >   8A07            MOV AL,BYTE PTR DS:[EDI]
01020DFB  .   47              INC EDI
01020DFC  .   08C0            OR AL,AL
01020DFE  .^  74 DC           JE SHORT 01020DDC
01020E00  .   89F9            MOV ECX,EDI
01020E02  .   57              PUSH EDI
01020E03  .   48              DEC EAX
01020E04  .   F2:AE           REPNE SCAS BYTE PTR ES:[EDI]
01020E06  .   55              PUSH EBP
01020E07  .   FF96 98680200   CALL DWORD PTR DS:[ESI+26898]       GetProcAddress
01020E0D  .   09C0            OR EAX,EAX
01020E0F  .v  74 07           JE SHORT 01020E18
01020E11  .   8903            MOV DWORD PTR DS:[EBX],EAX
01020E13  .   83C3 04         ADD EBX,4
01020E16  .^  EB E1           JMP SHORT 01020DF9
01020E18  >   FF96 9C680200   CALL DWORD PTR DS:[ESI+2689C]       ExitProcess
```

UPX->IAT

- MANIFEST 0001 0412
- IMPORT Directory Table
- IMPORT Address Table
- IMPORT DLL Names
- IMPORT Hints/Names

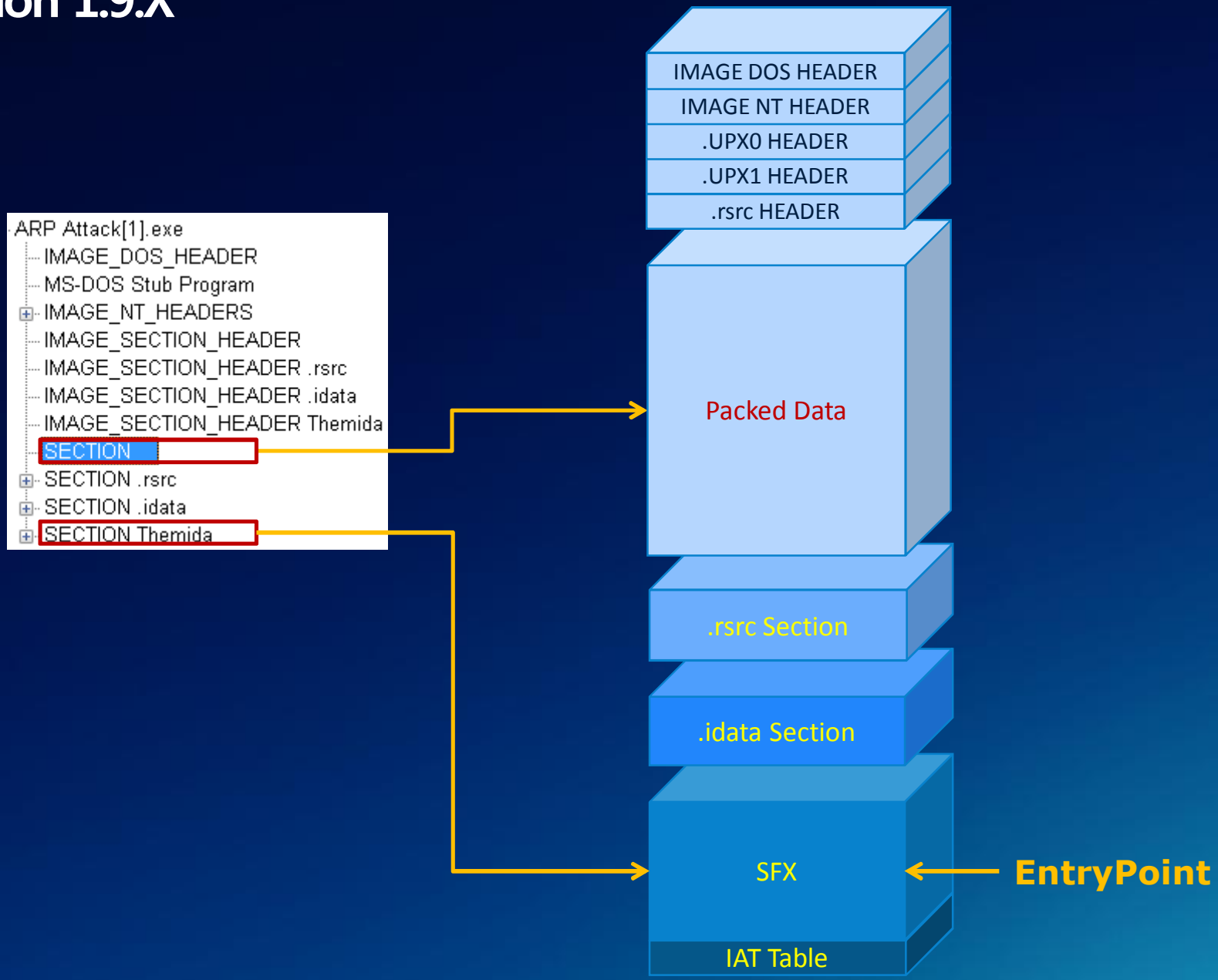
VA	Data	Description	Value
01027894	00027912	Hint/Name RVA	0000 LoadLibraryA
01027898	00027920	Hint/Name RVA	0000 GetProcAddress
0102789C	00027930	Hint/Name RVA	0000 ExitProcess
010278A0	00000000	End of Imports	KERNEL32.DLL

Manual Unpack Themida 1.9.X

Themida ?

- Themida
  Advanced Windows Software Protection System
- WinLicense
  Professional Software Protection & Licensing Management
- Code Virtualizer
  Total Obfuscation against Reverse Engineering

Version 1.9.X


ARP Attack[1].exe
IMAGE_DOS_HEADER
MS-DOS Stub Program
IMAGE_NT_HEADERS
IMAGE_SECTION_HEADER
IMAGE_SECTION_HEADER .rsrc
IMAGE_SECTION_HEADER .idata
IMAGE_SECTION_HEADER Themida
SECTION
SECTION .rsrc
SECTION .idata
SECTION Themida
IMAGE DOS HEADER
IMAGE NT HEADER
.UPX0 HEADER
.UPX1 HEADER
.rsrc HEADER
Packed Data
.rsrc Section
.idata Section
SFX
IAT Table
EntryPoint

00400000	00001000	ARP_Atta		PE header	Imag	R	RWE
00401000	00124000	ARP_Atta		code	Imag	R	RWE
00525000	0002F000	ARP_Atta	.rsrc	data,resources	Imag	R	RWE
00554000	00001000	ARP_Atta	.idata	imports	Imag	R	RWE
00555000	001B6000	ARP_Atta	Themida	SFX	Imag	R	RWE

Memory map

Address	Size	Owner	Section	Contains	Type	Access	Initial	Mapped as
00010000	00001000				Priv	RW	RW	
00020000	00001000				Priv	RW	RW	
0012C000	00001000				Priv	RW Guar	RW	
0012D000	00003000			stack of main thr3ad	Priv	RW Guar	RW	
00130000	00003000				Map	R	R	
00140000	00001000				Priv	RWE	RWE	
00150000	00005000				Priv	RW	RW	
00250000	00006000				Priv	RW	RW	
00260000	00003000				Map	RW	RW	
00270000	00016000				Map	R	R	\Device\HarddiskVolume1
00290000	00041000				Map	R	R	\Device\HarddiskVolume1
002E0000	00041000				Map	R	R	\Device\HarddiskVolume1
00330000	00006000				Map	R	R	\Device\HarddiskVolume1
00340000	00041000				Map	R	R	
00390000	00008000				Priv	RW	RW	
003A0000	00001000				Priv	RW	RW	
003B0000	00001000				Priv	RW	RW	
003C0000	00004000				Priv	RW	RW	
003D0000	00003000				Map	R	R	\Device\HarddiskVolume1
003E0000	00002000				Map	R	R	
00400000	00001000	ARP_Atta		PE header	Imag	R	RWE	
00401000	00124000	ARP_Atta		code	Imag	R	RWE	
00525000	0002F000	ARP_Atta	.rsrc	data,resources	Imag	R	RWE	
00554000	00001000	ARP_Atta	.idata	imports	Imag	R	RWE	
00555000	001B6000	ARP_Atta	Themida	SFX	Imag	R	RWE	
00710000	00002000				Map	R E	R E	
007D0000	00002000				Map	R E	R E	
007E0000	00103000				Map	R	R	
008F0000	00053000				Map	R E	R E	
5C820000	00001000	COMCTL32		PE header	Imag	R	RWE	
5C821000	00071000	COMCTL32	.text	code,imports,exports	Imag	R	RWE	
5C892000	00003000	COMCTL32	.data	data	Imag	R	RWE	
5C895000	00020000	COMCTL32	.rsrc	resources	Imag	R	RWE	
5C8B5000	00005000	COMCTL32	.reloc	relocations	Imag	R	RWE	
62340000	00001000	LPK		PE header	Imag	R	RWE	
62341000	00005000	LPK	.text	code,imports,exports	Imag	R	RWE	
62346000	00001000	LPK	.data	data	Imag	R	RWE	
62347000	00001000	LPK	.rsrc	resources	Imag	R	RWE	
62348000	00001000	LPK	.reloc	relocations	Imag	R	RWE	

00525000	0002F000	ARP_Atta	00400000	.rsrc	data,resour(	Imag	R	RWE
00554000	00001000	ARP_Atta	00400000	.idata	imports	Imag	R	RWE
00555000	001B6000	ARP_Atta	00400000	Themida	SFX	Imag	R	RWE
00710000	00003000		00710000 (itself)			Map	R E	R E
007D0000	00002000		00710000			Map	R E	R E
007E0000	00103000		007E0000 (itself)			Map	R	R
008F0000	0006B000		008F0000 (itself)			Map	R E	R E
00BF0000	00001000		00BF0000 (itself)			Priv	RW	RW
00C70000	0012C000		00C70000 (itself)			Priv	RWE	RWE
00DA0000	0008D000		00DA0000 (itself)			Priv	RWE	RWE
00E30000	000A4000		00E30000 (itself)			Priv	RWE	RWE

VirtualAlloc, CreateFile, ReadFile “ADVAPI32.DLL”

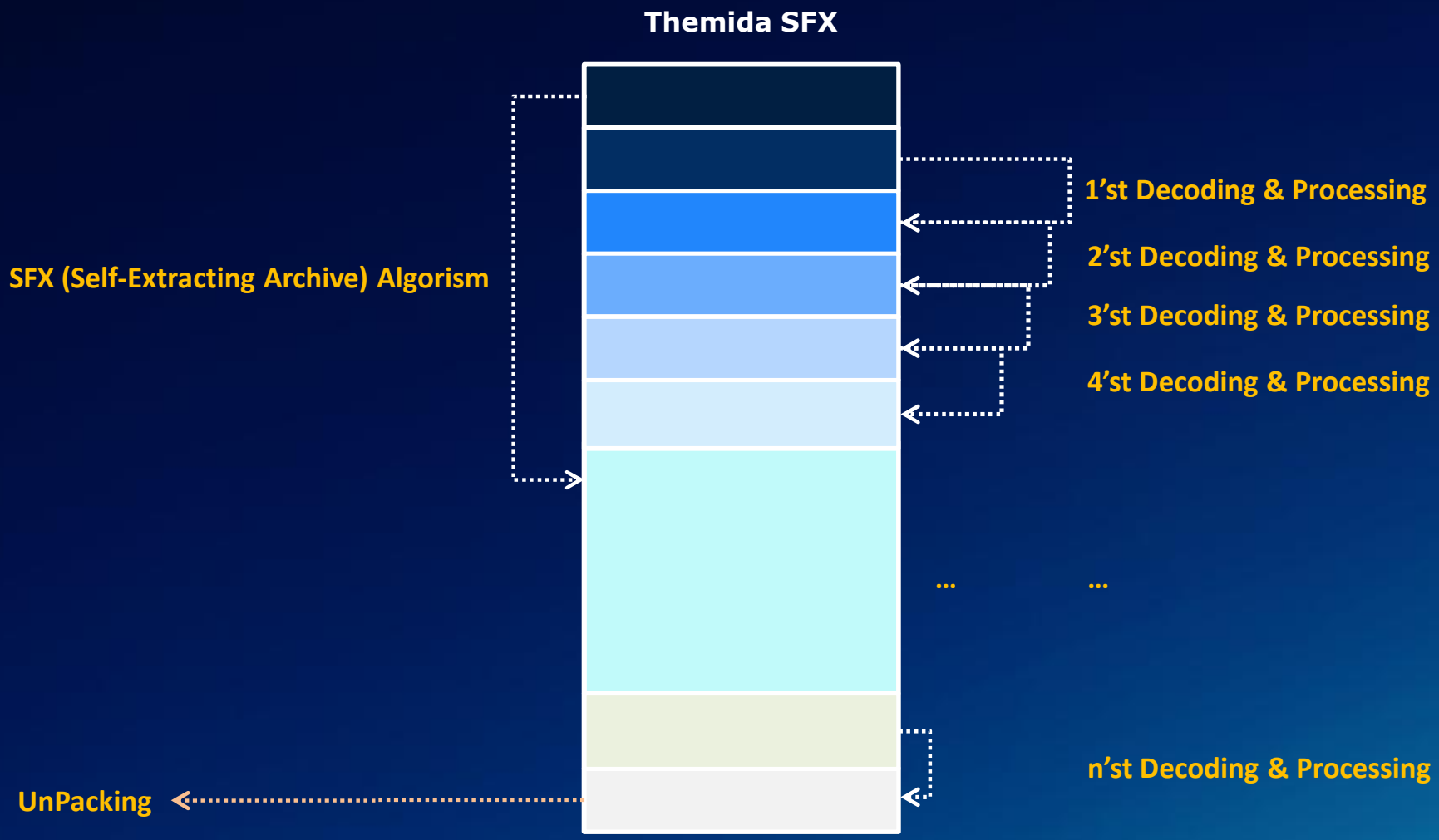
VirtualAlloc, CreateFile, ReadFile “USER32.DLL”

VirtualAlloc, CreateFile, ReadFile “KERNEL32.DLL”

Subsystem Virtualization

Ident	Entry	Data block	Last error	Status	Priority	User time	System time
000000AC	7C8106E9	7FF9C000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
0000015C	00555014	7FFDF000	ERROR_SUCCESS (00000000)	Active	32 + 0	3.7031 s	53.1718 s
00000304	7C8106E9	7FF9A000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
0000050C	7C8106E9	7FF9F000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
000006D8	7C8106E9	7FFDA000	ERROR_SUCCESS (00000000)	Paused	32 + 0	0.0000 s	0.0000 s
0000073C	7C8106E9	7FF97000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
000008C0	7C8106E9	7FFDE000	ERROR_SUCCESS (00000000)	Paused	32 + 0	0.0000 s	0.0000 s
000009D4	7C8106E9	7FF98000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000B88	7C8106E9	7FFDC000	ERROR_SUCCESS (00000000)	Paused	32 + 0	0.0000 s	0.0000 s
00000B98	7C8106E9	7FFD9000	ERROR_SUCCESS (00000000)	Paused	32 + 0	0.0000 s	0.0000 s
00000BCC	7C8106E9	7FF95000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000BD8	7C8106E9	7FFDB000	ERROR_SUCCESS (00000000)	Paused	32 + 0	0.0000 s	0.0000 s
00000BE8	7C8106E9	7FF9D000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000BF4	7C8106E9	7FFD4000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000BFC	7C8106E9	7FFD8000	ERROR_SUCCESS (00000000)	Paused	32 + 0	0.0000 s	0.0000 s
00000C00	7C8106E9	7FF9B000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000C04	7C8106E9	7FFD7000	ERROR_SUCCESS (00000000)	Paused	32 + 0	0.0000 s	0.0000 s
00000C08	7C8106E9	7FF9E000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000C0C	7C8106E9	7FFD6000	ERROR_SUCCESS (00000000)	Paused	32 + 0	0.0000 s	0.0000 s
00000C10	7C8106E9	7FFD5000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000C18	7C8106E9	7FFD3000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000C1C	7C8106E9	7FF99000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000C40	7C8106E9	7FF96000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000C4C	7C8106E9	7FF93000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s
00000C54	7C8106E9	7FF94000	ERROR_SUCCESS (00000000)	Paused	32 + 2	0.0000 s	0.0000 s

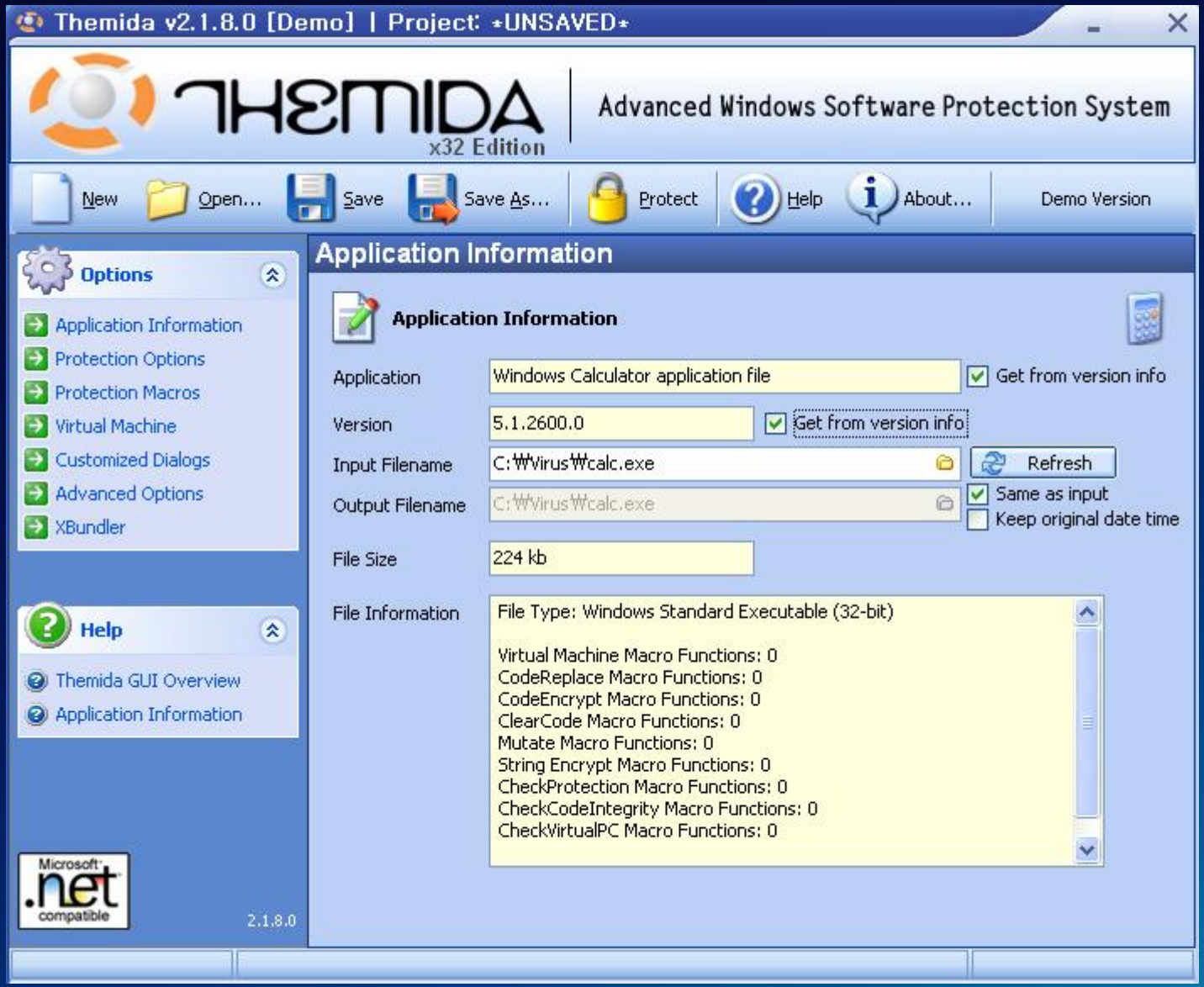
Multi-Thread



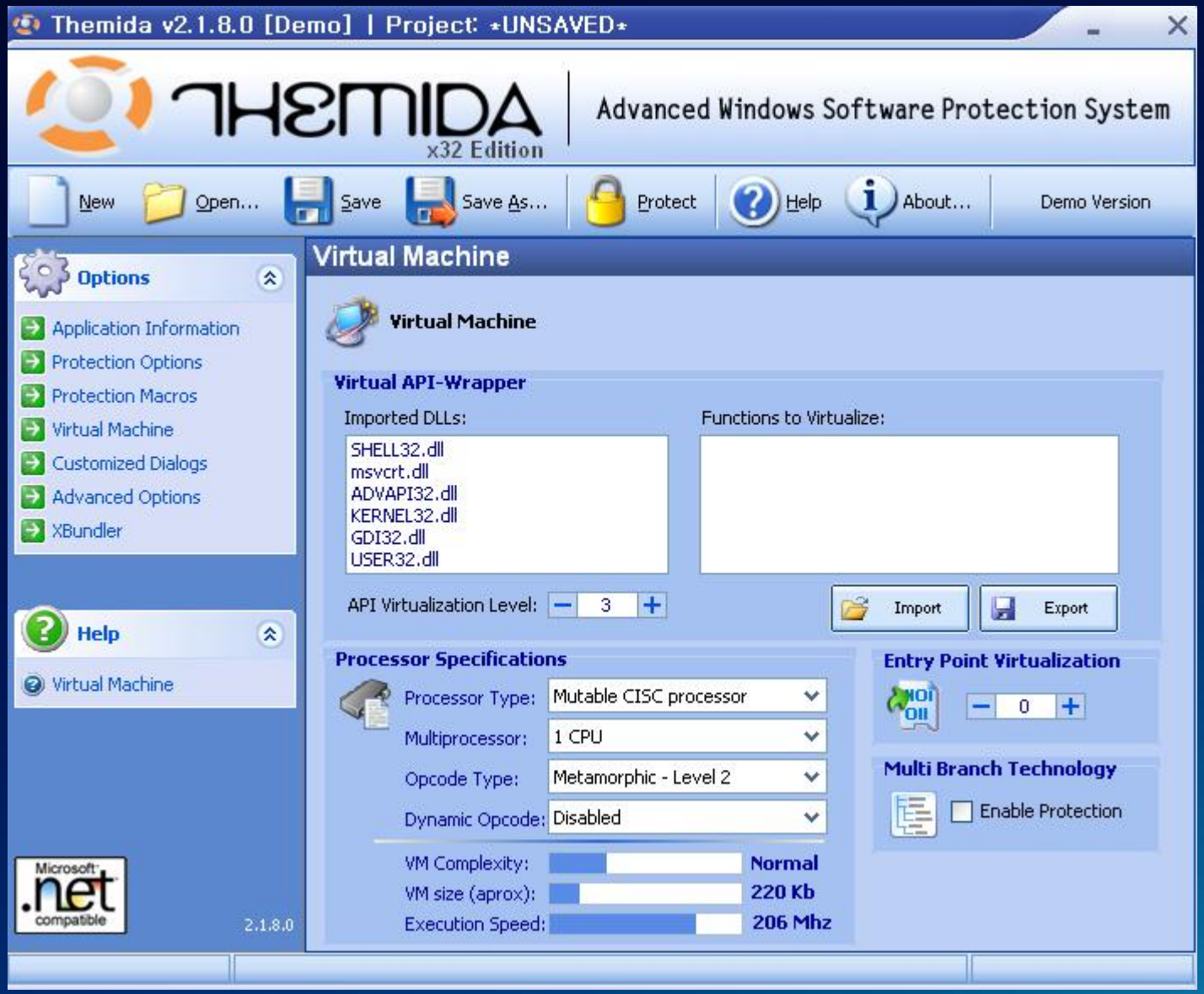
Decode & ReEncode

Manual Unpack Themida 2.1.8.0

New Version 2.1.8.0

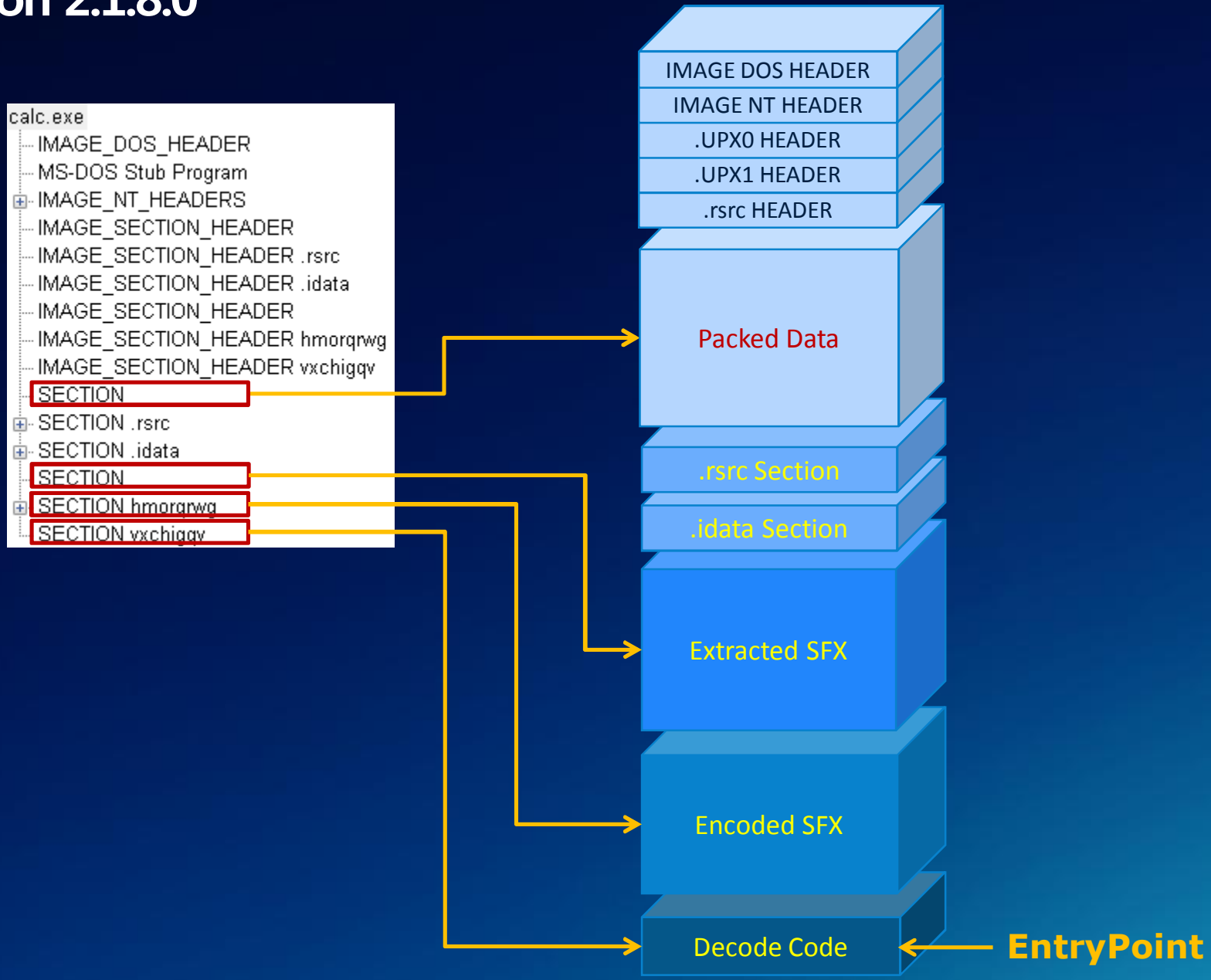


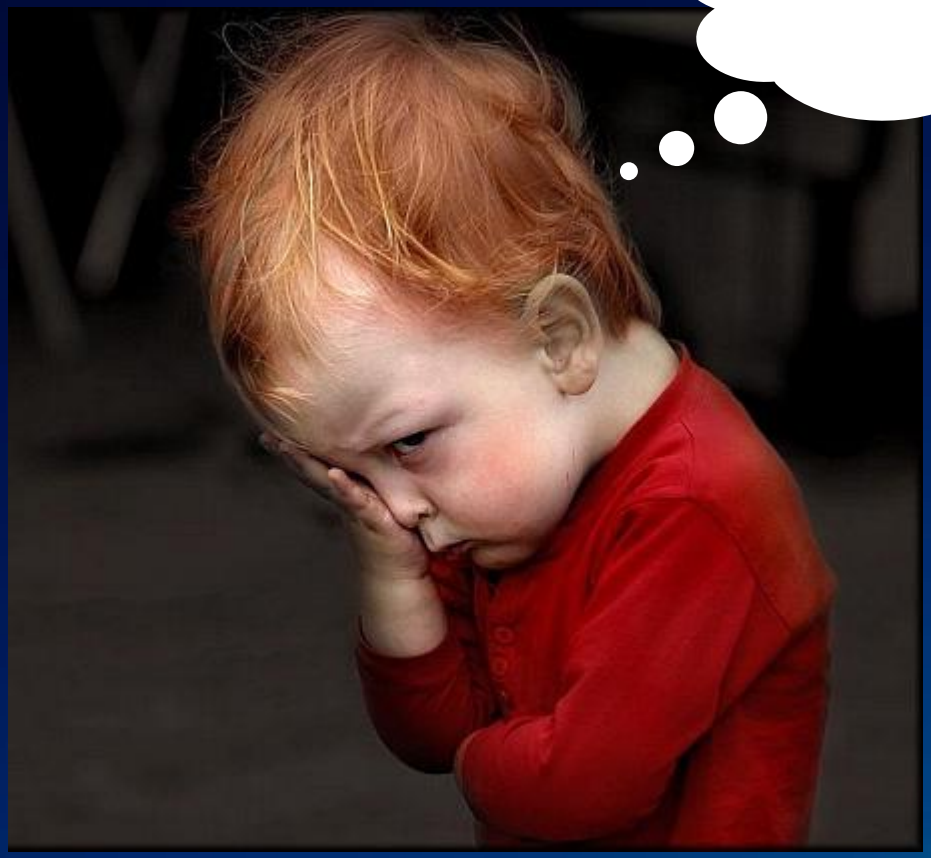

Themida v2.1.8.0 [Demo] | Project: *UNSAVED*
THEMIDA
x32 Edition
Advanced Windows Software Protection System
New
Open...
Save
Save As...
Protect
Help
About...
Demo Version
Options
Application Information
Protection Options
Protection Macros
Virtual Machine
Customized Dialogs
Advanced Options
XBundler
Help
Protection Options
Protect Now
SecureEngine Technology
Microsoft .net compatible
2.1.8.0
Protection Options
Protection Options
Anti-Debugger Detection
Advanced
Advanced API-Wrapping
Level 1
Compression
Application
Resources
SecureEngine
Anti Dumpers
Enable Protection
Anti-Patching
None
Monitor Blockers
Files Monitors
Registry Monitors
Entry Point Obfuscation
Enable Protection
Metamorph Security
Enable Protection
Resources Encryption
Enable Encryption
Memory Guard
Enable Protection
Delphi/BCB Form Protection
Enable Protection
VMWare/Virtual PC
Compatible
When Debugger Found
Display Message


Themida v2.1.8.0 [Demo] | Project: *UNSAVED*
THEMIDA
x32 Edition
Advanced Windows Software Protection System
New
Open...
Save
Save As...
Protect
Help
About...
Demo Version
Options
Application Information
Protection Options
Protection Macros
Virtual Machine
Customized Dialogs
Advanced Options
XBundler
Help
Virtual Machine
Microsoft .net compatible
2.1.8.0
Virtual Machine
Virtual Machine
Virtual API-Wrapper
Imported DLLs:
SHELL32.dll
msvcrt.dll
ADVAPI32.dll
KERNEL32.dll
GDI32.dll
USER32.dll
Functions to Virtualize:
API Virtualization Level: 3
Import
Export
Processor Specifications
Processor Type: Mutable CISC processor
Multiprocessor: 1 CPU
Opcode Type: Metamorphic - Level 2
Dynamic Opcode: Disabled
VM Complexity: Normal
VM size (aprox): 220 Kb
Execution Speed: 206 Mhz
Entry Point Virtualization
0
Multi Branch Technology
Enable Protection


Themida v2.1.8.0 [Demo] | Project: *UNSAVED*
Protect
Powered by SecureEngineAn
SecureEngine
ADVANCED ANTI-CRACKING TECHNOLOGY
Reading SDK Macros
Redirecting and Encrypting SDK Macros
Merging sections / Preparing PE Header
Examining and Wrapping API Calls
Generating Unique SecureEngine Code
Linking SecureEngine Code with Application
Generating Unique Virtual Machine
Compressing Virtual Machine
Virtualizing VM / CodeReplace Macros
Compressing Application and Resources
Encrypting Application and Resources
Cancel
Play sound when protection finishes

Version 2.1.8.0




... 어렵다