软件安全—恶意代码机理与防护

C3 PE文件格式

彭国军 教授

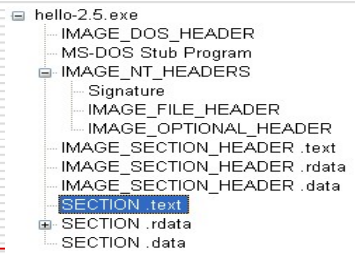
武汉大学国家网络安全学院

guojpeng@whu.edu.cn

本讲的内容提纲

3.1 PE文件及其表现形式
3.2 PE文件格式与恶意软件的关系
3.3 PE文件格式总体结构
3.4 代码节与数据节
3.5 引入函数节：PE文件的引入函数机制
3.6 引出函数节：DLL文件的函数引出机制
3.7 资源节：文件资源索引、定位与修改
3.8 重定位节：镜像地址改变后的地址自动修正

3.4.1 代码节

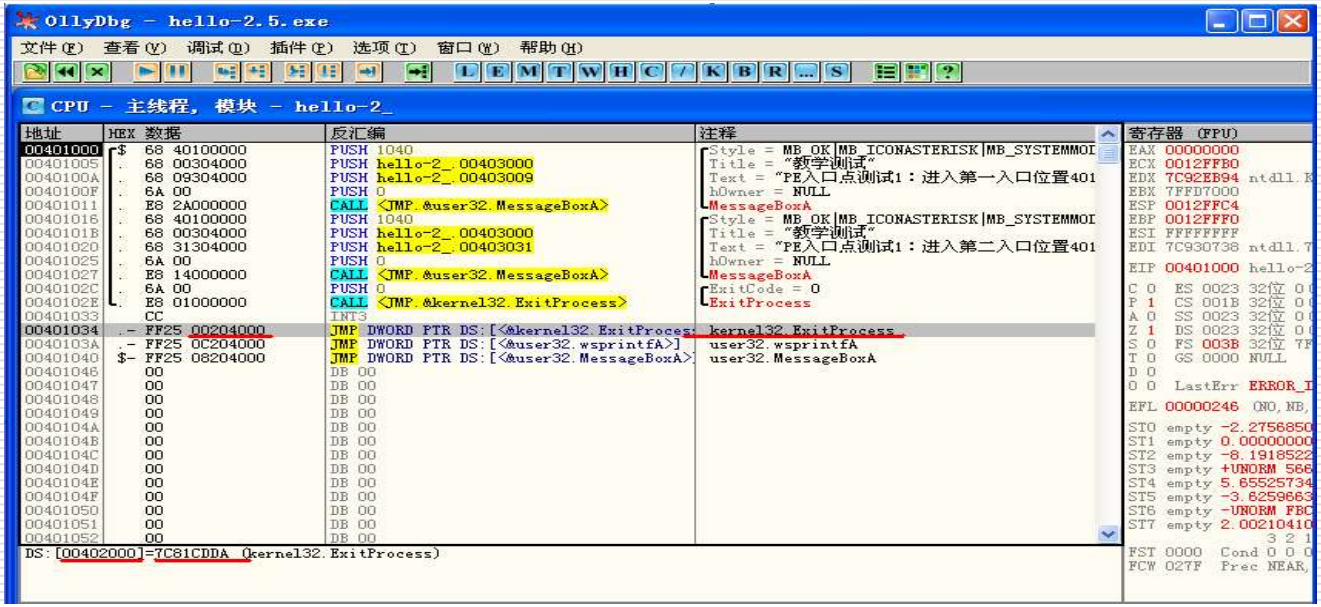



```
00000400h: 68 40 10 00 00 68 00 30 40 00 68 09 30 40 00 6A ; h@...h.0@.h.0@.j
00000410h: 00 E8 2A 00 00 00 68 40 10 00 00 68 00 30 40 00 ; .?...h@...h.0@.
00000420h: 68 31 30 40 00 6A 00 E8 14 00 00 00 6A 00 E8 01 ; h10@.j.?...j.?
00000430h: 00 00 00 CC FF 25 00 20 40 00 FF 25 0C 20 40 00 ; ...?%. @. %. @.
00000440h: FF 25 08 20 40 00 00 00 00 00 00 00 00 00 00 00 ;  %. @.........
00000450h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000460h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000470h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000480h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000490h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000500h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000510h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000520h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000530h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000540h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000550h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000560h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000570h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000580h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000590h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
```

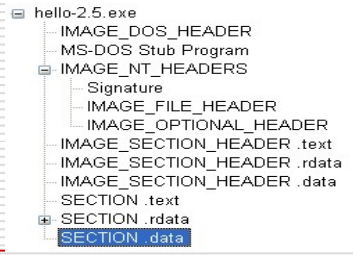
代码节

- 代码节一般名为.text或CODE，该节含有程序的可执行代码。
- 每个PE文件都有代码节

内存中的代码节


OllyDbg - hello-2.5.exe
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H)
CPU - 主线程, 模块 - hello-2_
地址 HEX 数据 反汇编 注释
00401000 $ 68 40100000 PUSH 1040 Style = MB_OK|MB_ICONASTERISK|MB_SYSTEMMOI
00401005 . 68 00304000 PUSH hello-2_.00403000 Title = "数学测试"
0040100A . 68 09304000 PUSH hello-2_.00403009 Text = "PE入口点测试1：进入第一入口位置401
0040100F . 6A 00 PUSH 0 hOwner = NULL
00401011 . E8 2A000000 CALL <JMP.&user32.MessageBoxA> MessageBoxA
00401016 . 68 40100000 PUSH 1040 Style = MB_OK|MB_ICONASTERISK|MB_SYSTEMMOI
0040101B . 68 00304000 PUSH hello-2_.00403000 Title = "数学测试"
00401020 . 68 31304000 PUSH hello-2_.00403031 Text = "PE入口点测试1：进入第二入口位置401
00401025 . 6A 00 PUSH 0 hOwner = NULL
00401027 . E8 14000000 CALL <JMP.&user32.MessageBoxA> MessageBoxA
0040102C . 6A 00 PUSH 0 ExitCode = 0
0040102E . E8 01000000 CALL <JMP.&kernel32.ExitProcess> ExitProcess
00401033 CC INT3
00401034 .- FF25 00204000 JMP DWORD PTR DS:[<&kernel32.ExitProces kernel32.ExitProcess
0040103A .- FF25 0C204000 JMP DWORD PTR DS:[<&user32.wsprintfA>] user32.wsprintfA
00401040 $- FF25 08204000 JMP DWORD PTR DS:[<&user32.MessageBoxA> user32.MessageBoxA
00401046 00 DB 00
00401047 00 DB 00
00401048 00 DB 00
00401049 00 DB 00
0040104A 00 DB 00
0040104B 00 DB 00
0040104C 00 DB 00
0040104D 00 DB 00
0040104E 00 DB 00
0040104F 00 DB 00
00401050 00 DB 00
00401051 00 DB 00
00401052 00 DB 00
DS:[00402000]=7C81CDDA (kernel32.ExitProcess)
寄存器 (FPU)
EAX 00000000
ECX 0012FFB0
EDX 7C92EB94 ntdll.K
EBX 7FFD7000
ESP 0012FFC4
EBP 0012FFF0
ESI FFFFFFFF
EDI 7C930738 ntdll.7
EIP 00401000 hello-2
C 0 ES 0023 32位 0
P 1 CS 001B 32位 0
A 0 SS 0023 32位 0
Z 1 DS 0023 32位 0
S 0 FS 003B 32位 7F
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_I
EFL 00000246 (NO,NB,
ST0 empty -2.2756850
ST1 empty 0.0000000
ST2 empty -8.1918522
ST3 empty +UNORM 566
ST4 empty 5.65525734
ST5 empty -3.6259663
ST6 empty -UNORM FBC
ST7 empty 2.0021041
3 2 1
FST 0000 Cond 0 0 0
FCW 027F Prec NEAR,

3.4.2 已初始化的数据节



- 这个节一般取名为.data或DATA
- 已初始化的数据节中放的是在编译时刻就已确定的数据。
  - 如test.exe中的字符串“PE入口点测试1：进入第一入口位置00401000H！”。

.data节


```
00000800h: BD CC D1 A7 B2 E2 CA D4 00 50 45 C8 EB BF DA B5 ; 数葡P镭?PE入口?
00000810h: E3 B2 E2 CA D4 31 A3 BA BD F8 C8 EB B5 DA D2 BB ; 悴镭?：进入第一
00000820h: C8 EB BF DA CE BB D6 C3 34 30 31 30 30 30 48 21 ; 入口位置401000H!
00000830h: 00 50 45 C8 EB BF DA B5 E3 B2 E2 CA D4 31 A3 BA ; .PE入口点测试1：
00000840h: BD F8 C8 EB B5 DA B6 FE C8 EB BF DA CE BB D6 C3 ; 进入第二入口位置
00000850h: 34 30 31 30 31 36 48 21 00 68 6B 64 6F 6F 72 64 ; 401016H!.hkdoord
00000860h: 6C 6C 2E 64 6C 6C 00 44 6C 6C 52 65 67 69 73 74 ; ll.dll.DllRegist
00000870h: 65 72 53 65 72 76 65 72 00 64 3A 5C 6D 61 73 6D ; erServer.d:\masm
00000880h: 33 32 5C 68 65 6C 6C 6F 74 2E 65 78 65 00 00 00 ; 32\hellot.exe...
00000890h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000900h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000910h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000920h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000930h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000940h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000950h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000960h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000970h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000980h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000990h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
```

3.4.3未初始化的数据节

- 节名称一般叫.bbs。

- 这个节里放有未初始化的全局变量和静态变量。
  - 例如“static int k;”