<https://www.52pojie.cn/forum.php?mod=viewthread&tid=64760>

【文章标题】: 手脱街头篮球某外挂 ZProtect 1.6 壳  
【文章作者】: 2666fff  
【作者邮箱】: [wan0001@brauer.vic.edu.au](mailto:wan0001@brauer.vic.edu.au)  
【软件名称】: 大牛外挂  
【下载地址】: 自己搜下  
【加壳方式】: Zprotect 1.6  
【编写语言】: 易语言  
【使用工具】: OD IR LoadPe ZPfixer ZPdump  
【作者声明】: 只是感兴趣，没有其他目的。失误之处敬请诸位大侠赐教!  
  视频版  : <http://www.52pojie.cn/thread-64651-1-1.html>  
--------------------------------------------------------------------------------  
【详细过程】  
  cektop大牛放出 Zp1.6 bypass All 了（至于在哪放的，自己找吧 ），网上无key解码也有很多，Zp定要再次更新，我就做一个完整的 Zp 1.6  
   非试用控制的[脱壳](https://www.52pojie.cn/forum-5-1.html" \t "_blank)教程。  
    
  感谢Cektop分享工具！  
    
  壳是Zp 1.6，虽然说没有注册框，但是有dll模拟，以及代码抽取，比默认的高级一点吧。  
  Zp 1.6 默认加壳会自动抽取代码，意味着壳的难度的增加，不像以前修了IAT就能运行的。  
  正文开始。  
    
  一、OEP  
  正如HC所说，壳的世界，进入了后OEP时代，也就是说，找到OEP，仅仅是往里长征的第一步。  
  ZProtect 就完全放弃了对OEP的处理，如同压缩壳一样，ESP就能到达OEP。  
  OD载入

1. 005452FF >  68 0C535400     push 大牛辅助.0054530C                       ; F8 然后ESP 找OEP
2. 00545304    68 19535400     push 大牛辅助.00545319                       ; ASCII "`h!ST"
3. 00545309    C3              retn
4. 0054530A    51              push ecx
5. 0054530B    97              xchg eax,edi
6. 0054530C    68 19535400     push 大牛辅助.00545319                       ; ASCII "`h!ST"
7. 00545311    C3              retn
8. 00545312    CD 0E           int 0xE

复制代码

    
  hr esp后，F9一次来到这里。

1. 00CF4DCE   /E9 CD830000     jmp 00CFD1A0
2. 00CF4DD3   |F2:             prefix repne:
3. 00CF4DD4   |8B45 08         mov eax,dword ptr ss:[ebp+0x8]
4. 00CF4DD7   |E9 AE890000     jmp 00CFD78A
5. 00CF4DDC   |9C              pushfd
6. 00CF4DDD   |DAB3 0F852C2A   fidiv dword ptr ds:[ebx+0x2A2C850F]
7. 00CF4DE3   |0000            add byte ptr ds:[eax],al

复制代码

  继续F9

1. 00477DB7    55              push ebp                                 ; OEP
2. 00477DB8    8BEC            mov ebp,esp
3. 00477DBA    6A FF           push -0x1
4. 00477DBC    68 A0E54C00     push 大牛辅助.004CE5A0
5. 00477DC1    68 A4A54700     push 大牛辅助.0047A5A4
6. 00477DC6    64:A1 00000000  mov eax,dword ptr fs:[0]
7. 00477DCC    50              push eax
8. 00477DCD    64:8925 0000000>mov dword ptr fs:[0],esp
9. 00477DD4    83EC 58         sub esp,0x58
10. 00477DD7    53              push ebx
11. 00477DD8    56              push esi
12. 00477DD9    57              push edi

复制代码

  删除分析，到达OEP，长征开始。  
    
  此时，数据窗口中跟随地址 00477DB7 地址，下断硬件写入断点，为之后的 劫持代码，即修复抽取代码做准备。  
    
  二、DLL模拟，与IAT修复。  
    
  重新载入程序。  
  下断 BP CreateFileA，F9运行一次，看堆栈。

1. 0012F908   00CF5BE9  /CALL 到 CreateFileA 来自 00CF5BE3
2. 0012F90C   0012F934  |FileName = "C:\windows\system32\ntdll.dll"
3. 0012F910   80000000  |Access = GENERIC\_READ
4. 0012F914   00000001  |ShareMode = FILE\_SHARE\_READ
5. 0012F918   00000000  |pSecurity = NULL
6. 0012F91C   00000003  |Mode = OPEN\_EXISTING
7. 0012F920   00000080  |Attributes = NORMAL
8. 0012F924   00000000  \hTemplateFile = NULL
9. 0012F928   0054A515  大牛辅助.0054A515
10. 0012F92C   00CDA7E8  ASCII "ntdll.dll"

复制代码

  发现存在dll模拟，在下断 BP GetModuleHandleA，F9一次，Alt +F9 返回。

1. 00CCA916    03F8            add edi,eax
2. 00CCA918    FF15 E810CC00   call dword ptr ds:[0xCC10E8]             ; kernel32.GetModuleHandleA
3. 00CCA91E    50              push eax                                 ; 停在这里
4. 00CCA91F    E8 90FDFFFF     call 00CCA6B4
5. 00CCA924    A1 F4A8CD00     mov eax,dword ptr ds:[0xCDA8F4]
6. 00CCA929    43              inc ebx

复制代码

  向上翻动代码

1. 00CCA8E6   /74 53           je short 00CCA93B
2. 00CCA8E8   |3998 AB000000   cmp dword ptr ds:[eax+0xAB],ebx          ; 进行DLL模拟的次数
3. 00CCA8EE   |8B15 0CA9CD00   mov edx,dword ptr ds:[0xCDA90C]          ; 大牛辅助.00547000
4. 00CCA8F4   |57              push edi
5. 00CCA8F5   |8D3C11          lea edi,dword ptr ds:[ecx+edx]
6. 00CCA8F8   |7E 3A           jle short 00CCA934
7. 00CCA8FA   |56              push esi
8. 00CCA8FB   |BE E8A7CD00     mov esi,0xCDA7E8                         ; ASCII "kernel32.dll"
9. 00CCA900   |56              push esi
10. 00CCA901   |57              push edi
11. 00CCA902   |FF35 D8A7CD00   push dword ptr ds:[0xCDA7D8]
12. 00CCA908   |E8 FFE5FFFF     call 00CC8F0C
13. 00CCA90D   |83C4 0C         add esp,0xC
14. 00CCA910   |56              push esi
15. 00CCA911   |A3 E4A7CD00     mov dword ptr ds:[0xCDA7E4],eax
16. 00CCA916   |03F8            add edi,eax
17. 00CCA918   |FF15 E810CC00   call dword ptr ds:[0xCC10E8]             ; kernel32.GetModuleHandleA
18. 00CCA91E   |50              push eax                                 ; 停在这里
19. 00CCA91F   |E8 90FDFFFF     call 00CCA6B4
20. 00CCA924   |A1 F4A8CD00     mov eax,dword ptr ds:[0xCDA8F4]
21. 00CCA929   |43              inc ebx
22. 00CCA92A   |3B98 AB000000   cmp ebx,dword ptr ds:[eax+0xAB]

复制代码

  00CCA8E8 处为比较要模拟的次数，下硬件执行断点，取消前面的两次API断点（不要删除任何硬件的），然后再次重新载入。  
    
  直接F9  
    
  00CCA8E8    3998 AB000000   cmp dword ptr ds:[eax+0xAB],ebx          ; 停在这里  
    
  看下方窗口  
    
  ebx=00000000  
  ds:[00D021E3]=00000006  
    
  右键修改 00000006 为 0，确定。 F9，F9 再次到达OEP。  
  此时先用LoadPe dump一次，然后开始修复IAT。  
  修复IAT可以用脚本，也可以用辅助程序，我推荐用ZPfixer，再次感谢Cektop！  
  先来查找 IATStart 及 IATend  
  运行 Importrec， OEP填 00077DB7 自动获取，此时肯定全部无效。  
  第一个为 0009B000 IATstart 为 0049B000  
  最后一个为 0009B708 IATend 为 0049B708 --- ptr 00543708  
  切换到OD，跟随地址 00543708

1. 00543708    68 86AC3665     push 0x6536AC86
2. 0054370D  ^ E9 1EFFFFFF     jmp 大牛辅助.00543630                    ；enter跟随
3. .........
4. 00543630  - E9 837A7800     jmp 00CCB0B8                             ; enter
5. 00543635    0000            add byte ptr ds:[eax],al
7. ........
9. 00CCB0B8    9C              pushfd
10. 00CCB0B9    60              pushad
11. 00CCB0BA    FF7424 24       push dword ptr ss:[esp+0x24]
12. 00CCB0BE    E8 05ECFFFF     call 00CC9CC8                            ; enter进入
13. 00CCB0C3    61              popad
14. 00CCB0C4    9D              popfd
15. 00CCB0C5    C3              retn                                     ; 此处地址为patchVA，记下。。
17. ........
19. 00CC9CC8    A1 F4A8CD00     mov eax,dword ptr ds:[0xCDA8F4]
20. 00CC9CCD    80B8 BB000000 0>cmp byte ptr ds:[eax+0xBB],0x0
21. 00CC9CD4    74 57           je short 00CC9D2D
22. 00CC9CD6    FF15 F010CC00   call dword ptr ds:[0xCC10F0]             ; kernel32.GetTickCount
23. 00CC9CDC    8BC8            mov ecx,eax
24. 00CC9CDE    2B0D A0A7CD00   sub ecx,dword ptr ds:[0xCDA7A0]          ; 大牛辅助.00523B93
25. 00CC9CE4    81F9 88130000   cmp ecx,0x1388
26. 00CC9CEA    76 41           jbe short 00CC9D2D
27. 00CC9CEC    FF35 A4A7CD00   push dword ptr ds:[0xCDA7A4]
28. 00CC9CF2    A3 A0A7CD00     mov dword ptr ds:[0xCDA7A0],eax
29. 00CC9CF7    FF15 4010CC00   call dword ptr ds:[0xCC1040]             ; kernel32.ResumeThread
30. 00CC9CFD    833D 60AFCD00 0>cmp dword ptr ds:[0xCDAF60],0x3          ; CDAF60 为zeroVA
31. 00CC9D04    7C 08           jl short 00CC9D0E
32. 00CC9D06    6A 00           push 0x0

复制代码

    
  好,记录下  
    
  IATstart  0049B000  
  IATend    0049B708  
  patchVA   00CCB0C5  
  zeroVA    00CDAF60  
    
  注意，后面的两个VA，在各个电脑上都是不同的，要自己找。  
    
    
  打开ZPfixer  
  process ID： 自己看你们的  
  IATstart= code start  
  IATend= code end  
    
  其他都填好，点击 start。  
  返回 New Address:  1000000  
  OD 跟随此地址，在此处新建EIP，末尾下断。

1. 01000000    BE 00B04900     mov esi,0x49B000                         ; 新建EIP
2. 01000005    BF 08B74900     mov edi,0x49B708
3. 0100000A    B9 C5B0CC00     mov ecx,0xCCB0C5
4. 0100000F    83C1 05         add ecx,0x5
5. 01000012    83C7 04         add edi,0x4
6. 01000015    8B06            mov eax,dword ptr ds:[esi]
7. 01000017    8931            mov dword ptr ds:[ecx],esi
8. 01000019    8A10            mov dl,byte ptr ds:[eax]
9. 0100001B    80FA 68         cmp dl,0x68
10. 0100001E    74 02           je short 01000022
11. 01000020    EB 0A           jmp short 0100002C
12. 01000022    8A50 05         mov dl,byte ptr ds:[eax+0x5]
13. 01000025    80FA E9         cmp dl,0xE9
14. 01000028  ^ 75 F6           jnz short 01000020
15. 0100002A    FFD0            call eax
16. 0100002C    83C6 04         add esi,0x4
17. 0100002F    C605 60AFCD00 0>mov byte ptr ds:[0xCDAF60],0x0
18. 01000036    3BF7            cmp esi,edi
19. 01000038    74 0B           je short 01000045
20. 0100003A    8B06            mov eax,dword ptr ds:[esi]
21. 0100003C    85C0            test eax,eax
22. 0100003E  ^ 75 D5           jnz short 01000015
23. 01000040    83C6 04         add esi,0x4
24. 01000043  ^ EB D0           jmp short 01000015
25. 01000045    33C0            xor eax,eax                              ; F2下断。

复制代码

  取消 之前的ESP硬件断点，F9运行

1. 01000043  ^\EB D0           jmp short 01000015
2. 01000045    33C0            xor eax,eax                              ; ole32.OleInitialize
3. 01000047    0000            add byte ptr ds:[eax],al

复制代码

  末尾出现 ole32.OleInitialize，此时importrec 点自动获取，有3个无效的，直接用等级1修复。  
  全部有效，保存下树文件，然后转存刚才dump的，此时肯定不能运行，因为我们没有修复代码。  
    
  三、代码的修复。  
    
  OD重载程序，要保证第一步中  “00477DB7 地址，下断硬件写入断点“  这个断点存在。不在的，再做一次。  
  F8，下esp断点，运行，来到处理dll的地方，刚修改值为0（这里一定要改！！），运行

1. 00CC79B6  ^\75 F2           jnz short 00CC79AA                       ; 停在此处
2. 00CC79B8    8975 08         mov dword ptr ss:[ebp+0x8],esi
3. 00CC79BB    33F6            xor esi,esi
4. 00CC79BD    46              inc esi
5. 00CC79BE    EB 12           jmp short 00CC79D2
6. 00CC79C0    8B45 EC         mov eax,dword ptr ss:[ebp-0x14]

复制代码

  ctrl + g 跟随 VirtualAlloc ，分别在开始跟结尾下断

1. 7C809A61 >  8BFF            mov edi,edi                              ; 下断
2. 7C809A63    55              push ebp
3. 7C809A64    8BEC            mov ebp,esp
4. 7C809A66    FF75 14         push dword ptr ss:[ebp+0x14]
5. 7C809A69    FF75 10         push dword ptr ss:[ebp+0x10]
6. 7C809A6C    FF75 0C         push dword ptr ss:[ebp+0xC]
7. 7C809A6F    FF75 08         push dword ptr ss:[ebp+0x8]
8. 7C809A72    6A FF           push -0x1
9. 7C809A74    E8 09000000     call kernel32.VirtualAllocEx
10. 7C809A79    5D              pop ebp
11. 7C809A7A    C2 1000         retn 0x10                                ; 下断

复制代码

  运行到 retn 看寄存器，此时EAX值为 00F70000 ，记下来，取消断点，运行到OEP，alt + m查看内存镜像。  
  由00F70000开始，一直到00FF0000，这些地址全部记下来，保存在一个文本文档中。  
    
  00F70000  
  00F80000  
  00F90000  
  00FA0000  
  00FB0000  
  00FC0000  
  00FD0000  
  00FE0000  
  00FF0000  
    
  打开Zpdump，开始补区段。ID自己填，填好后点load 选择刚才保存的文档，右边窗口出现数值，对他们点右键，GetVirtualSize,  
  然后点 Analysis，最后start，完毕后点dump保存。  
  OD载入脱好后的文件，发现最后一段(ImportREC补得段)的起始地址为 00633000  
  计算器算一下，  
  00F70000 - 00633000 = 93D000  
  将 dump过出错的文件载入 LoadPe PE编辑器。  
  点区段，右键编辑最后一段，虚拟大小改为 93D000 确定，再右键点击最后一段，从磁盘载入区段，载入刚保存的.dmp段  
  提示成功，保存，确定，重建一下PE。  
  可以运行，所有按钮都无错。  
  脱壳成功。  
    
--------------------------------------------------------------------------------  
【经验总结】  
  再再次感谢cektop，分享这么多工具。  
  Zp1.6会自动抽取，这让难度增加了很多，不过无key解码才是最高境界。  
  对ZP要有爱啊。  
  我不讲如何[破解](https://www.52pojie.cn/)这个挂，至于为何，大家猜？