2016 年第 16 题

大家好，我是胡一米。这期视频是关于 2016 年看雪 CTF 第 16 题解题资料。

一、简介

1.1 录制说明

视频是基于其他大佬文字 Writeup 录制的，所以在视频操作之前就已知了该题中的各种坑，如反调试等，所以不再演示踩坑过程。

1.2 录制参考

第 16 题主要参看了 HighHand、风间仁以及 daiweizhi 的 Writeup，链接如下：
HighHand:
https://bbs.pediy.com/thread-214418.htm
daiweizhi:
https://bbs.pediy.com/thread-214359.htm
风间仁：
https://bbs.pediy.com/thread-214358.htm

1.3 内容简介

在本期中，主要讨论 3 个问题。这 3 个问题可能与最终的 Key 没有太大的联系，但题目中既然出现了，还是说一说。此 3 个问题分别是：
a. 调试虚拟机保护
b run trace 用法
c. nor 门和 xor 门

二、视频操作

2.1 静态分析定位关键点

结合其他大佬的分析过程，可以直接定位到关键点。.

2.2 动态调试其加密过程

通过动态调试，理解程序内存加载过程。
vm_code: ESI

```
0043E565   1B 01 00 00 00 1B 00 00 00 00 26 01 04 00 18 1B
0043E575   18 00 00 00 17 17 11 FC 0D 0F 18 12 05 17 16 1F   ...?
0043E585   1B 01 00 00 00 1B 00 00 00 00 26 01 04 00 27 FC   .'?
0043E595   1B 01 00 00 00 17 17 11 F4 1B 01 00 00 00 1B 00   ...?....
0043E5A5   00 00 00 26 01 04 00 27 F8 1B 00 00 00 00 17 17   .'?....
0043E5B5   11 F8 1B 01 00 00 00 1B 00 00 00 00 26 01 04 00   .
0043E5C5   27 FC 1B 00 00 00 00 17 17 0E 1E F8 24 01 40 24   '?....-?@$
0043E5D5   02 84 C9 1B F5 E5 43 00 1B B7 E5 43 00 27 00 14   劻蹂 C.峰 C.'.
0043E5E5   19 1B 40 00 00 00 14 19 19 1B 06 00 00 00 02 1C   @.....
0043E5F5   27 FC 27 F4 05 09 16 11 FC 27 FC 0F 0D 12 05 09   '??.??..
0043E605   16 01 04 00 1B 26 E6 43 00 1B D5 EA 43 00 27 00   贞 C.'.
0043E615   14 19 1B 40 00 00 00 14 19 19 1B 06 00 00 00 02   @.....
0043E625   1C 1B 01 00 00 00 1B 00 00 00 00 26 01 04 00 18
```

Dispatcher : 421224

```
004210BD    主    xor    ax, 0BF77          FL=S, EAX=0000E851
004210C2    主    jmp    short 004210C6
004210C6    主    lea    eax, dword ptr [edi+40]          EAX=0012FAF8
004210C9    主    jmp    short 004210CC
004210CC    主    cmp    ebp, eax          FL=PA
004210CE    主    jmp    short 004210D1
004210D1    主    ja     00421208
00421208    主    movzx  eax, byte ptr [esi]          EAX=0000001B
0042120B    主    jmp    0042123A
0042123A    主    lea    esi, dword ptr [esi+1]          ESI=0043E5D9
0042123D    主    jmp    0042121E
0042121E    主    popfd          FL=A, ESP=0012FA78
0042121F    主    jmp    short 00421224
00421224    主    jmp    dword ptr [eax*4+42271B] ;这里跳转到每个 handler
```

2.3 完成分析

理解每个 handler，完成分析。

1B :

```
00421224      jmp dword ptr ds:[eax*4+0x42271B]
00421928      and eax,0xED825741                    ; FL=0, EAX=00000001
0042192E      mov ax,word ptr ds:[edi+0x8]          ; EAX=0000C5EC
00421936      jmp short CrackMe_.0042193B
0042193B      sbb eax,0xED82D59D                    ; FL=CA, EAX=127DF04F
00421941      jmp CrackMe_.00421D12
00421D12      add eax,dword ptr ds:[esi+0x2]        ; FL=0, EAX=1298F04F
00421D19      jmp CrackMe_.004219F6
004219F6      mov eax,dword ptr ds:[esi]            ; EAX=00000001
004219F8      jmp short CrackMe_.004219FE
004219FE      add esi,0x4                           ; FL=P, ESI=0043E56A
00421A01      jmp short CrackMe_.00421A05
00421A05      sub ebp,0x4                           ; FL=A, EBP=0018FBBC
00421A08      jmp short CrackMe_.00421A0B
00421A0B      mov dword ptr ss:[ebp],eax
00421A0E      jmp CrackMe_.00421810
00421810      jmp CrackMe_.0042126B
```

vm_push_imm32

EBP : stack

26 :

```
00421224     jmp dword ptr ds:[eax*4+0x42271B]
004213F9     nop dword ptr ss:[esp+0xc]
00421401     jmp CrackMe_.00421CFB
00421CFB     bt ax,cx
00421CFF     jmp CrackMe_.004216A9
004216A9     bswap eax                             ; EAX=26000000
004216AB     mov eax,dword ptr ss:[ebp+0x4]        ; EAX=00000001
004216AE     btr dx,0x4B
004216B3     jmp CrackMe_.00421675
00421675     movzx edx,word ptr ss:[ebp+0x4]       ; EDX=00000001
0042167D     jmp short CrackMe_.0042167F
0042167F     mov dx,sp                             ; EDX=0000FA58
00421682     jmp CrackMe_.0042136B
0042136B     movsx edx,byte ptr ss:[esp+0x6]       ; EDX=FFFFFFE3
00421373     jmp short CrackMe_.00421375
00421375     bt dx,cx                              ; FL=CP
00421379     btr edx,0x3E                          ; EDX=BFFFFFE3
0042137D     rol edx,0xc3                          ; EDX=FFFFFF1D
00421380     movzx edx,dl                          ; EDX=0000001D
00421383     jmp CrackMe_.0042170C
0042170C     ror dl,0xd4                           ; EDX=000000D1
```

```
0042170F    jmp short CrackMe_.00421713
00421713    mov edx,dword ptr ss:[ebp]              ; EDX=00000000
00421716    imul edx                                  ; FL=P, EAX=00000000
00421718    jmp short CrackMe_.0042171D
0042171D    mov dword ptr ss:[ebp+0x4],eax
00421720    mov dword ptr ss:[ebp],edx
00421723    jmp CrackMe_.0042157A
0042157A    jmp CrackMe_.0042126B
```

vm_imul

```
01：
00421224    jmp dword ptr ds:[eax*4+0x42271B]
00421582    movzx eax,word ptr ds:[esi]             ; EAX=00000004
00421585    add esi,0x2                               ; FL=0, ESI=0043E573
00421588    jmp short CrackMe_.0042158A
0042158A    add ebp,eax                               ; EBP=0018FBBC
0042158C    jmp short CrackMe_.0042158F
0042158F    jmp CrackMe_.0042126B
```

vm_add_stk

```
18：
00421224   jmp dword ptr ds:[eax*4+0x42271B]
004219D0    shr ax,cl                                 ; FL=PZ, EAX=00000000
004219D3    jmp CrackMe_.0042194C
0042194C    xor al,byte ptr ss:[ebp+0xA]
00421953   jmp short CrackMe_.00421957
00421957   lea eax,dword ptr ss:[ebp+0x4]          ; EAX=0018FBC0
0042195A    jmp CrackMe_.00421553
00421553   sub ebp,0x4                               ; FL=P, EBP=0018FBB8
00421556   mov dword ptr ss:[ebp],eax
00421559   jmp short CrackMe_.0042155C
0042155C    jmp CrackMe_.0042126B
```

vm_push_esp

```
11：
00421224   jmp dword ptr ds:[eax*4+0x42271B]
00421B83    xor dx,0x5A                               ; EDX=0000005A
00421B87   jmp short CrackMe_.00421B8C
00421B8C    mov dx,dx
00421B8F   jmp CrackMe_.00421B2F
00421B2F   sbb dx,0xFFBF                              ; FL=CA, EDX=0000009B
```

```
00421B33   jmp short CrackMe_.00421B39
00421B39   movzx edx,bx                          ; EDX=0000E000
00421B3C   movsx eax,byte ptr ds:[esi]           ; EAX=FFFFFFFC
00421B3F   and dx,cx                             ; FL=PZ, EDX=00000000
00421B42   shl dx,0xe4
00421B46   jmp CrackMe_.00421510
00421510   shr dx,cl
00421513   jmp short CrackMe_.00421518
00421518   add esi,0x1                          ; FL=P, ESI=0043E57D
0042151B   or dx,word ptr ss:[esp+0xB]           ; FL=PS, EDX=0000E277
00421523   jmp short CrackMe_.00421525
00421525   movzx dx,byte ptr ss:[esp+0x2]       ; EDX=000000E8
0042152E   or dx,cx                              ; FL=P, EDX=000013F9
00421531   jmp CrackMe_.0042200F
0042200F   or edx,0x23                           ; FL=0, EDX=000013FB
00422012   jmp CrackMe_.00421E42
00421E42   mov edx,dword ptr ss:[ebp]            ; EDX=0018FBD8
00421E45   jmp CrackMe_.00421C2E
00421C2E   add ebp,0x4                           ; FL=PA, EBP=0018FBC0
00421C31   mov dword ptr ds:[edi+eax],edx
00421C34   jmp short CrackMe_.00421C36
00421C36   jmp CrackMe_.0042126B
```

vm_pop_reg32
EDI: context

19：

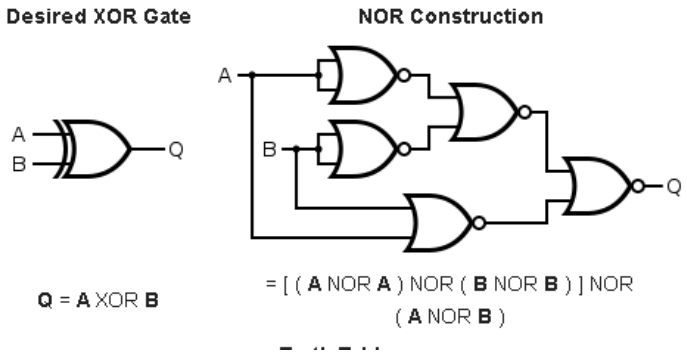
```
00421224   jmp dword ptr ds:[eax*4+0x42271B]
00421C3B   sub edx,0xED9C6B39                    ; EDX=126392C4
00421C41   btc ax,bx                            ; FL=C, EAX=00000018
00421C45   jmp CrackMe_.00421F7E
00421F7E   dec dx                                ; FL=CPS, EDX=126392C3
00421F81   jmp short CrackMe_.00421F85
00421F85   xor dx,0xBBE                          ; FL=PS, EDX=1263997D
00421F8A   jmp CrackMe_.00421AC8
00421AC8   sub al,0xC3                           ; FL=CP, EAX=00000055
00421ACB   jmp short CrackMe_.00421AD0
00421AD0   ???
00421AD8   jmp CrackMe_.0042185F
0042185F   ???
00421868   jmp short CrackMe_.0042186A
0042186A   movsx ax,byte ptr ss:[esp+0x9]        ; EAX=00000000
00421873   jmp short CrackMe_.00421875
00421875   mov eax,dword ptr ss:[ebp]            ; EAX=00000080
```

```
00421878   jmp short CrackMe_.0042187C
0042187C   or dx,0xFFF2                        ; FL=PS, EDX=1263FFFF
00421880   jmp short CrackMe_.00421885
00421885   mov edx,dword ptr ss:[ebp+0x4]     ; EDX=00000080
00421888   jmp CrackMe_.004216C6
004216C6   not eax                            ; EAX=FFFFFF7F
004216C8   jmp short CrackMe_.004216CE
004216CE   not edx                            ; EDX=FFFFFF7F
004216D0   jmp short CrackMe_.004216D2
004216D2   add ebp,0x4                        ; FL=P, EBP=0018FACC
004216D5   jmp short CrackMe_.004216DB
004216DB   and eax,edx                        ; FL=S
004216DD   jmp short CrackMe_.004216E0
004216E0   mov dword ptr ss:[ebp],eax
004216E3   jmp CrackMe_.00421D66
00421D66   jmp CrackMe_.0042126B
```

vm_nor32

XOR [edit]

An XOR gate is made by connecting the output of 3 NOR gates (connected as an AND gate) and the output of a NOR gate to the respective inputs of a NOR gate. This construction entails a propagation delay three times that of a single NOR gate and uses five gates.



三、小结

题目很好。

附录：

录制顺序：29-28-7-19-22-16-？