【2020腾讯游戏安全技术竞赛】PC方向初赛 WriteUp

原创

古月浪子 于 2020-04-13 20:53:42 发布 1851 收藏 3

文章标签： 安全 安全漏洞 游戏 腾讯 windows

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/tqydyqt/article/details/105497000

版权

> 很遗憾没有进入复赛，不过还是记录一下自己做题的过程叭
> PC方向的初赛题目有2道，一道Ring0一道Ring3，由于技术不到位 我只做了简单的Ring3，无缘复赛QAQ
> Rlng0的题目为：给了一个加了vmp壳的驱动，该驱动无法正常加载，要求在不修改驱动文件的情况下尝试让驱动成功加载，并设法让驱动成功执行print
> Ring3的题目为：给了一个扫雷exe和一份dmp文件，要求通过分析得出外挂修改了程序的哪些地方、实现了什么功能

首先拿到winmine.exe于winmine.dmp

尝试windbg加载dmp文件，指定image file为exe，输入指令 !analyze -v ，分析得到结果，崩溃是由int3指令产生（不过这并不重要，重要的是内存转储文件中的程序数据）

```
CONTEXT:  (.ecxr)
eax=002c8000 ebx=00000000 ecx=773eb3b0 edx=773eb3b0 esi=773eb3b0 edi=773eb3b0
eip=773b2790 esp=044cff44 ebp=044cff70 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000244
ntdll!DbgBreakPoint:
773b2790 cc              int     3
Resetting default scope

FAULTING_IP:
ntdll!DbgBreakPoint+0
773b2790 cc              int     3

EXCEPTION_RECORD:  (.exr -1)
ExceptionAddress: 773b2790 (ntdll!DbgBreakPoint)
   ExceptionCode: 80000003 (Break instruction exception)
  ExceptionFlags: 00000000
NumberParameters: 1
   Parameter[0]: 00000000
```

使用exepeinfo查看该exe的pe结构，找到text段偏移和大小

Sections viewer : [winmine.exe] 3 sections - alignment : 1000h

Nr	Virtual ...	Virtual s...	RAW Data offset	RAW size	Flags	Name	First bytes (hex)
01 e...	00001000	00003A56	00000400	00003C00	60000020	.text	65 1B DA 77 0B 58 DA 77 EA
02	00005000	00000B98	00004000	00000200	C0000040	.data	18 00 00 00 8F 00 00 00 8D
03 rs	00006000	00019160	00004200	00019200	40000040	.rsrc	00 00 00 00 00 00 00 00 04

使用winhex打开exe文件，将其text段裁剪下来，在这里我采用了text段偏移的基础上再加0x1d4的偏移，故裁剪后得到的文件大小为0x3a2c字节

再使用winhex打开dmp文件，根据特征码搜索定位到text段被映射到的地址偏移，采用同样的方法将0x3a2c大小的块裁剪下来

分析比对2个裁剪后的文件的不同

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

搜索不同

1. C:\Users\古月浪子\Desktop\winmine.exe: 14,892 字节
2. C:\Users\古月浪子\Desktop\winmine.dmp: 14,892 字节
Offsets: 十六进制

1E21:	FF	90
1E22:	05	90
1E23:	9C	90
1E24:	57	90
1E25:	00	90
1E26:	01	90
23BD:	6A	EB
23BE:	00	1D

8 区别 被发现.

可以发现，有6个字节的代码被填充为了nop，在IDA上找到该处代码

```
.text:01002FE0
.text:01002FE0 sub_1002FE0     proc near               ; CODE XREF: sub_1
.text:01002FE0                 cmp     dword_1005164, 0
.text:01002FE7                 jz      short locret_1003007
.text:01002FE9                 cmp     dword_100579C, 3E7h
.text:01002FF3                 jge     short locret_1003007
.text:01002FF5                 inc     dword_100579C
.text:01002FFB                 call    sub_10028B5
.text:01003000                 push    1
.text:01003002                 call    sub_10038ED
.text:01003007
.text:01003007 locret_1003007:                         ; CODE XREF: sub_1
```

可以发现，该外挂将inc dword_100579c这条命令跳过了

```
285       ShowWindow(::hWnd, 0);
286     }
287     SendMessageW(::hWnd, 0x112u, 0xF060u, 0);
288     return 0;
289   case 0x112u:
290     v7 = wParam & 0xFFF0;
291     if ( v7 == 61472 )
292     {
293       sub_100341C(lParam);
294       dword_1005000 |= 0xAu;
295     }
296     else if ( v7 == 61728 )
297     {
298       dword_1005000 &= 0xF5u;
299       sub_100344C(lParam);
300       dword_1005148 = 0;
301     }
302     return DefWindowProcW(hWnd, Msg, wParam, lParam);
303   case 0x113u:
304     sub_1002FE0();
305     return 0;
306 }
307 return DefWindowProcW(hWnd, Msg, wParam, lParam);
308}
0000116C sub_1001BC9:304 (1001D6C)
```

在Windows消息循环中有一个sub_1002fe0函数，这条被nop的命令就在该函数中，该函数会将dword_100579c与999进行比较，如果小于999，则自增dword_100579c

```
1 void sub_1002FE0()
2 {
3   if ( dword_1005164 )
4   {
5     if ( dword_100579C < 999 )
6     {
7       ++dword_100579C;
8       sub_10028B5();
```

```
      sub_10036ED(1);
    }
  }
}
```

根据进一步调试，我确定了这是与计时有关的函数，恰好扫雷的时间显示框只能容纳3位数(最高表示数999)，自增命令被nop后扫雷的计时功能将停止，数字不会随着时间递增，时间会永远显示为001

```
.text:01003588
.text:01003588 loc_1003588:                            ; CODE XREF: sub_1003512+24↑j
.text:01003588                 push    4Ch
.text:0100358A                 push    eax
.text:0100358B                 push    esi
.text:0100358C                 call    sub_1002EAB
.text:01003591                 push    0
.text:01003593                 jmp     short loc_10035AB
.text:01003595 ; ---------------------------------------------------------------------------
.text:01003595
.text:01003595 loc_1003595:                            ; CODE XREF: sub_1003512+1B↑j
```

第二个被修改的指令为push 0，被修改后的代码为jmp short loc_10035b0，也就是直接跳转至函数尾

```
.text:01003588
.text:01003588 loc_1003588:                            ; CODE XREF: sub_1003512+24↑j
.text:01003588                 push    4Ch
.text:0100358A                 push    eax
.text:0100358B                 push    esi
.text:0100358C                 call    sub_1002EAB
.text:01003591                 jmp     short loc_10035B0
.text:01003593 ; ---------------------------------------------------------------------------
.text:01003593                 jmp     short loc_10035AB
.text:01003595 ; ---------------------------------------------------------------------------
.text:01003595
```

正常的执行流程为

```
void __stdcall sub_1003512(int a1, int a2)
{
  char *v2; // edx
  signed int v3; // eax
  char *v4; // edi
  signed int v5; // ecx

  v2 = &byte_1005340[32 * a2 + a1];
  if ( *v2 >= 0 )
  {
    sub_1003084(a1, a2);
    if ( dword_10057A4 == dword_10057A0 )
      sub_100347C(1);
  }
  else if ( dword_10057A4 )
  {
    sub_1002EAB(a1, a2, 76);
    sub_100347C(0);
  }
  else
  {
    v3 = 1;
    if ( dword_1005338 > 1 )
    {
      v4 = (char *)&unk_1005360;
      while ( 1 )
      {
        v5 = 1;
        if ( dword_1005334 > 1 )
          break;
LABEL_8:
        ++v3;
        v4 += 32;
        if ( v3 >= dword_1005338 )
          return;
      }
      while ( v4[v5] < 0 )
      {
        if ( ++v5 >= dword_1005334 )
          goto LABEL_8;
      }
      *v2 = 15;
      byte_1005340[32 * v3 + v5] |= 0x80u;
```

```
44        sub_1003084(a1, a2);
45      }
46    }
47  }
```

而修改后代码的第18行的call sub_100347c(0)没有被执行，该函数的作用是点到雷后的逻辑操作，不执行该函数相当于点到雷以后游戏并不会结束，依然可以继续扫雷

综上所述，该外挂通过patch了2处代码实现了如下功能：

（1）时间锁定为001

FF 05 9C 57 00 01 → 90 90 90 90 90 90

（2）点到雷以后游戏并不会结束

6A 00 → EB 1D

> 不敢说这是正确答案，只能说是我的解题报告，不足之处还望各位师傅斧正