海莲花APT组织最新攻击样本分析

2018-04-27 15:47:32

海莲花（OceanLotus）也叫APT32或APT-C-00，是一个长期针对中国及其他东亚国家（地区）政府、科研机构、海运企业等领域进行攻击的APT组织。近日腾讯御见威胁情报中心捕获到了一个该组织的最新攻击样本。在本次攻击事件中，该组织使用了CVE-2017-11882漏洞并结合白签名利用程序...

0x1 概况

海莲花（OceanLotus）也叫APT32或APT-C-00，是一个长期针对中国及其他东亚国家（地区）政府、科研机构、海运企业等领域进行攻击的APT组织。近日腾讯御见威胁情报中心捕获到了一个该组织的最新攻击样本。在本次攻击事件中，该组织使用了CVE-2017-11882漏洞并结合白签名利用程序来最大化隐藏自己的后门木马。后门木马将常驻用户电脑，并根据云控指令伺机窃取机密信息或进行第二阶段攻击。

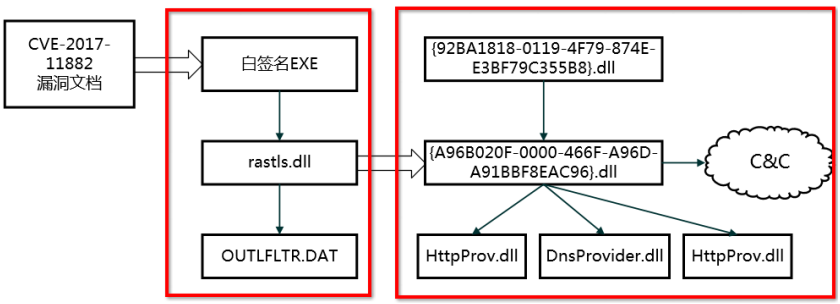


图1

（攻击流程图）

0x2 载荷投递

本次攻击时使用了CVE-2017-11882漏洞文档，诱饵文档文件名为《Document_GPI Invitation-UNSOOC China.doc》，内容为一模糊的图片。

最新资讯

医院遭勒索病毒攻击—腾讯企业安速响应成功防御获感谢

企业未修复Apache Struts 2漏洞 至服务器被批量入侵

黑产竞争激烈：一台服务器遭遇两客进攻

PhotoMiner木马挖矿收入8900万门罗币"黄金矿工"

腾讯2018年Q1季度互联网安全报基础设施或成重点攻击对象

以Windows服务器为攻击目标 或病毒新趋势

最新APT组织"寄生兽"活动披露

针对企业定向攻击的Xtreme、Tes

AVC安卓杀软测评最佳名额独占三讯TAV杀毒引擎大放异彩

Weblogic 反序列化命令执行 (CVE-2628)

图2

(诱饵文档)

漏洞触发后，公式编辑器进程会在“C:\Program Files\Microsoft-Windows-DiskDiagnosticResolver_2021325962”目录下释放MicrosoftWindowsDiskDiagnosticResolver.exe、OUTLFLTR.DAT、rastls.dll 3个文件。其中MicrosoftWindowsDiskDiagnosticResolver.exe有有效签名，原始名为dot1xtra.exe，这是典型的白加黑利用技术,带有效签名的exe运行后会自动加载木马文件rastls.dll。

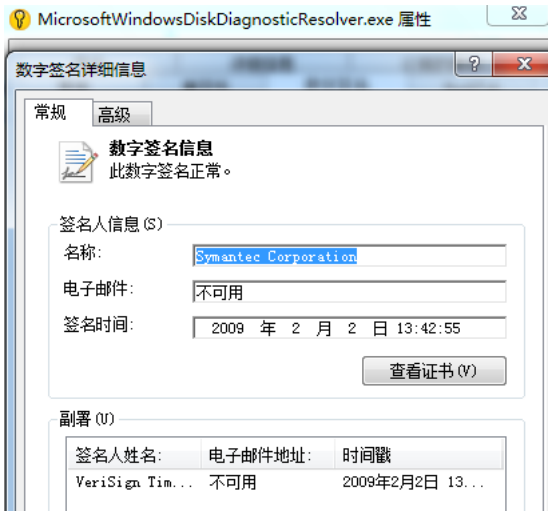


图3

(签名信息)

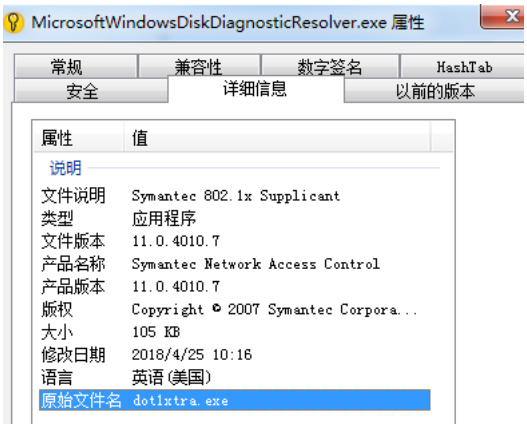


图4

（文件详细信息）

0x3 RAT分析

1.rastls.dll 行为分析

该组织在本次攻击中使用的关键模块文件都加入了大量混淆，混淆代码如下所示。此dll会在DLLMain函数中加载OUTLFLTR.DAT，进行解密后得到一段shellcode。接着会将宿主exe即MicrosoftWindowsDiskDiagnosticResolver.exe 0x401000（默认基址为0x400000）开始的一大片代码修改为无实际作用的指令，宿主exe oep位置的指令也被修改了。当宿主exe从OEP位置开始执行时，就会跳转到木马的shellcode部分。

```
.text:10005940                mov     ebp, [esp+4]
.text:10005943                pushf
.text:10005944                push    ecx
.text:10005945                neg     cx
.text:10005948                push    edx
.text:10005949                neg     dx
.text:1000594C                sahf
.text:1000594D                push    eax
.text:1000594E                aas
.text:1000594F                push    ebx
.text:10005950                bts     bx, 2
.text:10005955                not     bl
.text:10005957                aas
.text:10005958                not     dx
.text:1000595B                sahf
.text:1000595C                daa
.text:1000595D                das
.text:1000595E                nop
.text:1000595F                aam
.text:10005961                inc     ecx
.text:10005962                bsf     edx, ebx
.text:10005965                not     ebx
.text:10005967                or      cx, ax
.text:1000596A                xor     bh, 49h
.text:1000596D                cdq
.text:1000596E                bsf     ebx, ecx
.text:10005971                das
.text:10005972                mov     eax, [esp+10h+var_C]
.text:10005976                lea     ecx, [edx+ebx-62B8h]
.text:1000597D                rol     edx, 0
.text:10005980                mov     edx, [esp+10h+var_8]
.text:10005984                not     ecx
.text:10005986                mov     ecx, [esp+10h+var_4]
.text:1000598A                inc     bx
.text:1000598C                mov     ebx, [esp+10h+var_s0]
.text:10005990                push    ebx
.text:10005991                popf
```

图5

（混淆代码）

```
00401000 ┌$  52          push edx
00401001 |.  41          inc ecx
00401002 |?  48          dec eax
00401003 |.  43          inc ebx
00401004 |?  43          inc ebx
00401005 |?  59          pop ecx
00401006 |.  42          inc edx
00401007 |?  57          push edi
00401008 |?  43          inc ebx
00401009 |?  4A          dec edx
0040100A |?  43          inc ebx
0040100B |?  90          nop
0040100C |.  40          inc eax
0040100D |.  40          inc eax
0040100E |.  40          inc eax                ┌pModule
0040100F |?  59          pop ecx
00401010 |.  49          dec ecx                └GetModuleHandleW
```

图6

（宿主exe 0x401000处被修改后的指令）

```
0040C195 |.  51          push ecx
0040C196 |.  41          inc ecx
0040C197 |>  59          pop ecx
0040C198 |?  48          dec eax
0040C199 |?  90          nop
0040C19A |.  43          inc ebx
0040C19B |.  B8 D01E0010 mov eax,0x10001ED0
0040C1A0 |?  FFD0        call eax
0040C1A2 |?  C3          retn
```

图7

（宿主exe 被修改后OEP附近的代码）

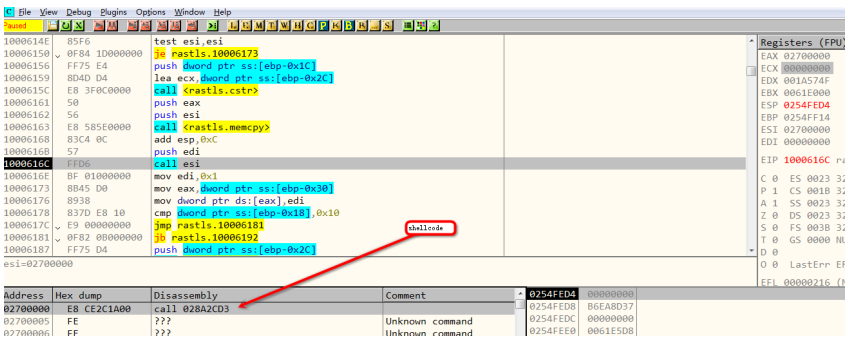


图8

（rastls.dll 跳转到shellcode执行）

OUTLFLTR.DAT中的shellcode被执行后，会自加载shellcode中存储的一个名为{92BA1818-0119-4F79-874E-E3BF79C355B8}.dll。接着执行此dll的导出函数DllEntry。

```
'data:10011508 ;
'data:10011508 ; Export Address Table for {92BA1818-0119-4F79-874E-E3BF79C355B8}.dll
'data:10011508 ;
'data:10011508 off_10011508    dd rva DllEntry         ; DATA XREF: .rdata:100114FC↑o
'data:1001150C ;
'data:1001150C ; Export Names Table for {92BA1818-0119-4F79-874E-E3BF79C355B8}.dll
'data:1001150C ;
'data:1001150C off_1001150C    dd rva aDllentry        ; DATA XREF: .rdata:10011500↑o
'data:1001150C                                         ; "DllEntry"
'data:10011510 ;
'data:10011510 ; Export Ordinals Table for {92BA1818-0119-4F79-874E-E3BF79C355B8}.dll
'data:10011510 ;
'data:10011510 word_10011510   dw 0                    ; DATA XREF: .rdata:10011504↑o
'data:10011512 a92ba181801194f db '{92BA1818-0119-4F79-874E-E3BF79C355B8}.dll',0
'data:10011512                                         ; DATA XREF: .rdata:100114EC↑o
'data:1001153D aDllentry       db 'DllEntry',0         ; DATA XREF: .rdata:off_1001150C↑o
```

图9

（dll内部名称及导出函数）

{92BA1818-0119-4F79-874E-E3BF79C355B8}.dll 又会从资源中解密出木马功能文件{A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll，并自加载此dll，执行此dll的导出函数DllEntry。

```
.rdata:10089E38 ;
.rdata:10089E38 ; Export Address Table for {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll
.rdata:10089E38 ;
.rdata:10089E38 off_10089E38    dd rva DllEntry         ; DATA XREF: .rdata:10089E2C↑o
.rdata:10089E3C ;
.rdata:10089E3C ; Export Names Table for {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll
.rdata:10089E3C ;
.rdata:10089E3C off_10089E3C    dd rva aDllentry        ; DATA XREF: .rdata:10089E30↑o
.rdata:10089E3C                                         ; "DllEntry"
.rdata:10089E40 ;
.rdata:10089E40 ; Export Ordinals Table for {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll
.rdata:10089E40 ;
.rdata:10089E40 word_10089E40   dw 0                    ; DATA XREF: .rdata:10089E34↑o
.rdata:10089E42 aA96b020f000046 db '{A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll',0
.rdata:10089E42                                         ; DATA XREF: .rdata:10089E1C↑o
.rdata:10089E6D aDllentry       db 'DllEntry',0         ; DATA XREF: .rdata:off_10089E3C↑o
.rdata:10089E76                 align 200h
.rdata:10089E76 _rdata          ends
```

图10

（木马功能文件导出函数及内部文件名）

2. {A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll 行为分析

该dll执行后，会解密资源得到木马的c2等配置信息及3个通信相关的dll，名称分别为HttpProv.dll、DnsProvider.dll、HttpProv.dll。每个通信dll都导出一个"CreateInstance"函数。

```
.text:1004F65A                  mov     [ebp+LockResource], eax
.text:1004F65D                  test    ebx, ebx
.text:1004F65F                  jz      loc_1004F727
.text:1004F665                  cmp     [ebp+LoadResource], 0
.text:1004F669                  jz      loc_1004F727
.text:1004F66F                  cmp     [ebp+SizeofResource], 0
.text:1004F673                  jz      loc_1004F727
.text:1004F679                  test    eax, eax
.text:1004F67B                  jz      loc_1004F727
.text:1004F681                  mov     eax, [ebp+arg_4]
.text:1004F684                  mov     ecx, [ebp+arg_0]
.text:1004F687                  push    eax
.text:1004F688                  push    ecx
.text:1004F689                  push    10000000h
.text:1004F68E                  call    ebx             ; ebx=77E25517 (kernel32.FindResourceW)
.text:1004F690                  mov     esi, eax
.text:1004F692                  test    esi, esi
.text:1004F694                  jz      loc_1004F727
.text:1004F69A                  push    esi
.text:1004F69B                  push    10000000h
.text:1004F6A0                  call    [ebp+LoadResource] ; Stack ss:[0012FE6C]=77E29CBA (kernel32.LoadResource)
.text:1004F6A3                  mov     ebx, eax
```

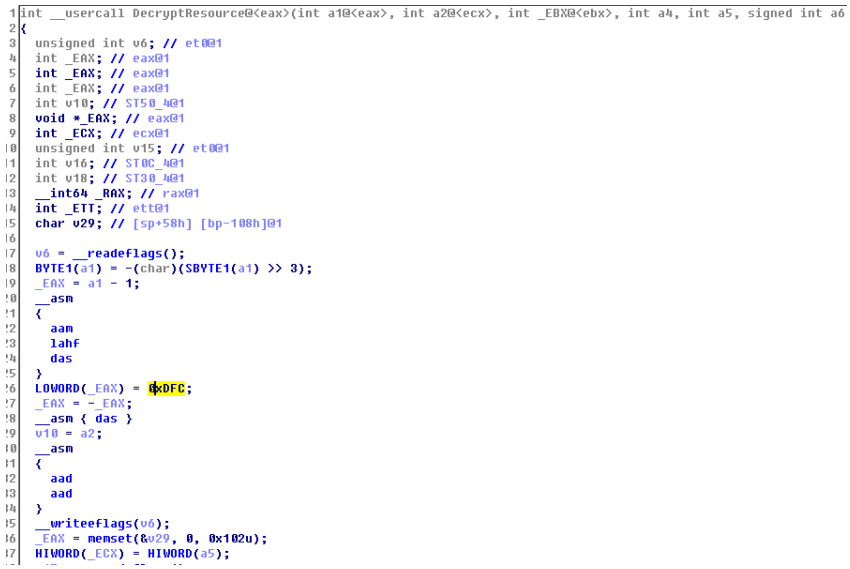
图11

(加载资源)

```
.text:10053D51                  cwd
.text:10053D53                  push    ecx
.text:10053D54                  aam
.text:10053D56                  bt      ax, 0
.text:10053D5B                  inc     ecx
.text:10053D5C                  or      ax, ax
.text:10053D5F                  mov     eax, 0EFCh
.text:10053D64                  mov     ecx, 52A4h
.text:10053D69                  mul     ecx
.text:10053D6B                  mov     eax, 15B7h
.text:10053D70                  mov     ecx, 284Ch
.text:10053D75                  mul     ecx
.text:10053D77                  mov     eax, [esp+18h]
.text:10053D7B                  push    eax
.text:10053D7C                  popf
.text:10053D7D                  mov     eax, [esp+14h]  ; eax=res
.text:10053D81                  mov     ecx, [esp+0C8h+var_C4]
.text:10053D85                  mov     edx, [esp+0C8h+var_BC]
.text:10053D89                  lea     esp, [esp+1Ch]
.text:10053D8D                  mov     [esp+0], eax
.text:10053D90                  push    80h
.text:10053D95                  push    offset unk_1008A0F8
.text:10053D9A                  call    DecryptResource
.text:10053D9F                  mov     eax, [ebp-1Ch]
.text:10053DA2                  mov     ecx, [ebp+outbuf]
.text:10053DA5                  add     esp, 14h
```

图12

（解密资源）

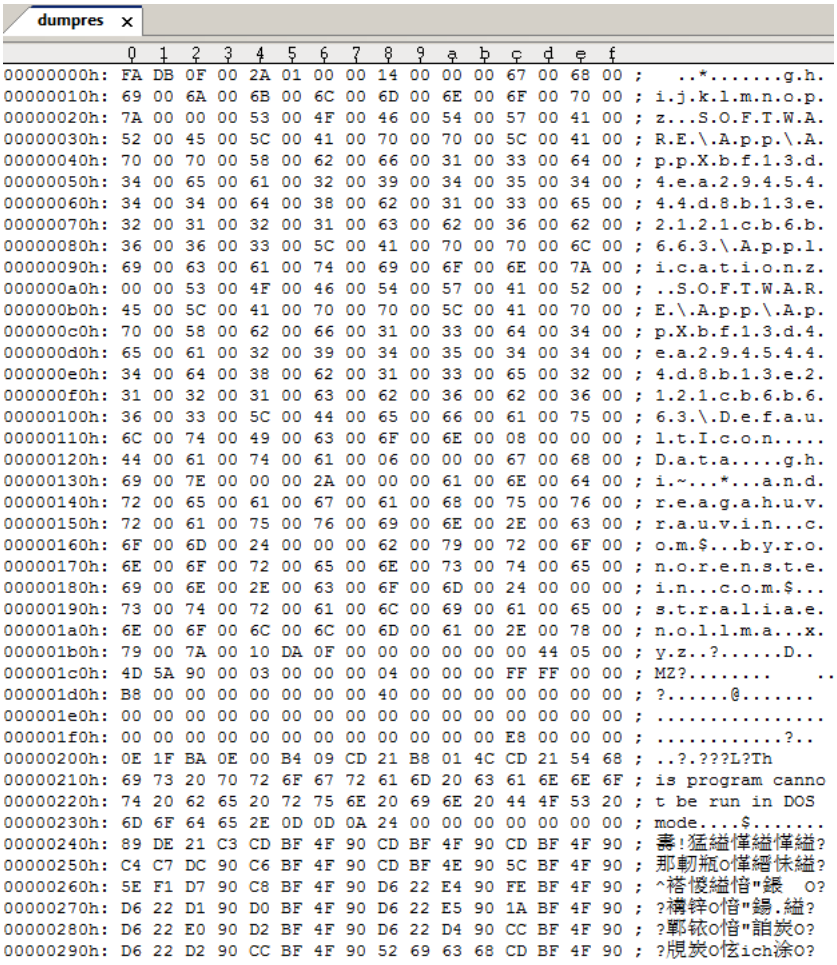


```
int __usercall DecryptResource@<eax>(int a1@<eax>, int a2@<ecx>, int _EBX@<ebx>, int a4, int a5, signed int a6
{
  unsigned int v6; // et0@1
  int _EAX; // eax@1
  int _EAX; // eax@1
  int _EAX; // eax@1
  int v10; // ST50_4@1
  void *_EAX; // eax@1
  int _ECX; // ecx@1
  unsigned int v15; // et0@1
  int v16; // ST0C_4@1
  int v18; // ST30_4@1
  __int64 _RAX; // rax@1
  int _ETT; // ett@1
  char v29; // [sp+58h] [bp-108h]@1

  v6 = __readeflags();
  BYTE1(a1) = ~(char)(SBYTE1(a1) >> 3);
  _EAX = a1 - 1;
  __asm
  {
    aam
    lahf
    das
  }
  LOWORD(_EAX) = 0xDFC;
  _EAX = -_EAX;
  __asm { das }
  v10 = a2;
  __asm
  {
    aad
    aad
  }
  __writeeflags(v6);
  _EAX = memset(&v29, 0, 0x102u);
  HIWORD(_ECX) = HIWORD(a5);
```

图13

（解密资源的函数代码片断）



dumpres

```
            0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: FA DB 0F 00 2A 01 00 00 14 00 00 00 67 00 68 00 ;   ..*.......g.h.
00000010h: 69 00 6A 00 6B 00 6C 00 6D 00 6E 00 6F 00 70 00 ; i.j.k.l.m.n.o.p.
00000020h: 7A 00 00 00 53 00 4F 00 46 00 54 00 57 00 41 00 ; z...S.O.F.T.W.A.
00000030h: 52 00 45 00 5C 00 41 00 70 00 70 00 5C 00 41 00 ; R.E.\.A.p.p.\.A.
00000040h: 70 00 70 00 58 00 62 00 66 00 31 00 33 00 64 00 ; p.p.X.b.f.1.3.d.
00000050h: 34 00 65 00 61 00 32 00 39 00 34 00 35 00 34 00 ; 4.e.a.2.9.4.5.4.
00000060h: 34 00 34 00 64 00 38 00 62 00 31 00 33 00 65 00 ; 4.4.d.8.b.1.3.e.
00000070h: 32 00 31 00 32 00 31 00 63 00 62 00 36 00 62 00 ; 2.1.2.1.c.b.6.b.
00000080h: 36 00 36 00 33 00 5C 00 41 00 70 00 70 00 6C 00 ; 6.6.3.\.A.p.p.l.
00000090h: 69 00 63 00 61 00 74 00 69 00 6F 00 6E 00 7A 00 ; i.c.a.t.i.o.n.z.
000000a0h: 00 00 53 00 4F 00 46 00 54 00 57 00 41 00 52 00 ; ..S.O.F.T.W.A.R.
000000b0h: 45 00 5C 00 41 00 70 00 70 00 5C 00 41 00 70 00 ; E.\.A.p.p.\.A.p.
000000c0h: 70 00 58 00 62 00 66 00 31 00 33 00 64 00 34 00 ; p.X.b.f.1.3.d.4.
000000d0h: 65 00 61 00 32 00 39 00 34 00 35 00 34 00 34 00 ; e.a.2.9.4.5.4.4.
000000e0h: 34 00 64 00 38 00 62 00 31 00 33 00 65 00 32 00 ; 4.d.8.b.1.3.e.2.
000000f0h: 31 00 32 00 31 00 63 00 62 00 36 00 62 00 36 00 ; 1.2.1.c.b.6.b.6.
00000100h: 36 00 33 00 5C 00 44 00 65 00 66 00 61 00 75 00 ; 6.3.\.D.e.f.a.u.
00000110h: 6C 00 74 00 49 00 63 00 6F 00 6E 00 08 00 00 00 ; l.t.I.c.o.n.....
00000120h: 44 00 61 00 74 00 61 00 06 00 00 00 67 00 68 00 ; D.a.t.a.....g.h.
00000130h: 69 00 7E 00 00 00 2A 00 00 00 61 00 6E 00 64 00 ; i.~...*...a.n.d.
00000140h: 72 00 65 00 61 00 67 00 61 00 68 00 75 00 76 00 ; r.e.a.g.a.h.u.v.
00000150h: 72 00 61 00 75 00 76 00 69 00 6E 00 2E 00 63 00 ; r.a.u.v.i.n...c.
00000160h: 6F 00 6D 00 24 00 00 00 62 00 79 00 72 00 6F 00 ; o.m.$...b.y.r.o.
00000170h: 6E 00 6F 00 72 00 65 00 6E 00 73 00 74 00 65 00 ; n.o.r.e.n.s.t.e.
00000180h: 69 00 6E 00 2E 00 63 00 6F 00 6D 00 24 00 00 00 ; i.n...c.o.m.$...
00000190h: 73 00 74 00 72 00 61 00 6C 00 69 00 61 00 65 00 ; s.t.r.a.l.i.a.e.
000001a0h: 6E 00 6F 00 6C 00 6C 00 6D 00 61 00 2E 00 78 00 ; n.o.l.l.m.a...x.
000001b0h: 79 00 7A 00 10 DA 0F 00 00 00 00 00 00 44 05 00 ; y.z..?......D..
000001c0h: 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 ; MZ?........    ..
000001d0h: B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ; ?.......@.......
000001e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000001f0h: 00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00 ; ............?..
00000200h: 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 ; ..?.???L?Th
00000210h: 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F ; is program canno
00000220h: 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 ; t be run in DOS
00000230h: 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 ; mode....$.......
00000240h: 89 DE 21 C3 CD BF 4F 90 CD BF 4F 90 CD BF 4F 90 ; 壽!猛縊惲縊惲縊?
00000250h: C4 C7 DC 90 C6 BF 4F 90 CD BF 4E 90 5C BF 4F 90 ; 那軔瓶o惲繙怵縊?
00000260h: 5E F1 D7 90 C8 BF 4F 90 D6 22 E4 90 FE BF 4F 90 ; ^褡惾縊愔"鎵  O?
00000270h: D6 22 D1 90 D0 BF 4F 90 D6 22 E5 90 1A BF 4F 90 ; ?褥锌o愔"鍚.縊?
00000280h: D6 22 E0 90 D2 BF 4F 90 D6 22 D4 90 CC BF 4F 90 ; ?鄞铱o愔"誼炭o?
00000290h: D6 22 D2 90 CC BF 4F 90 52 69 63 68 CD BF 4F 90 ; ?覞炭o恆ich涂o?
```

图14

（资源解密后的明文内容）

将解密后的资源进行分析后得到资源中各字段的值及部分含义

字段序号	字段内容
1	Unicode 字符串“ghijklmnop”
2	Unicode字符串，注册表键值“SOFTWARE\App\AppXbf13d4ea2945444d8b13e2121cb6b663\Application”

3	Unicode字符串，注册表键值“SOFTWARE\App\AppXbf13d4ea2945444d8b13e2121cb6b663\DefaultIcon”，存储发包时用的guid
4	Unicode字符串“Data”
5	“ghi” 此值与计算机用户名进行运算后得到互斥体名称
6	andreagahuvrauvin.com c2
7	byronorenstein.com c2
8	straliaenollma.xyz c2
9	0x54400字节的pe文件，名称为HttpProv.dll
10	0x35c00字节的pe文件，名称为DnsProvider.dll
11	0x073a00字节的pe文件，名称也为HttpProv.dll
12	4字节，值为0x88a36523，可以理解为版本号
13	4字节，值为0x13
14	Unicode字符串“46405”，即c2端口号
15	8字节，Hex值为”2B CA 18 AF D7 11 EA 59”，发包时会带上此值
16	8字节，Hex值为“80 A9 03 00 30 75 00 00”，根据此值Sleep一定时间后再执行木马功能

```
rdata:1002C708 ;
rdata:1002C708 ; Export Address Table for DnsProvider.dll
rdata:1002C708 ;
rdata:1002C708 off_1002C708    dd rva CreateInstance   ; DATA XREF: .rdata:1002C6FC↑o
rdata:1002C70C ;
rdata:1002C70C ; Export Names Table for DnsProvider.dll
rdata:1002C70C ;
rdata:1002C70C off_1002C70C    dd rva aCreateinstance  ; DATA XREF: .rdata:1002C700↑o
rdata:1002C70C                                         ; "CreateInstance"
rdata:1002C710 ;
rdata:1002C710 ; Export Ordinals Table for DnsProvider.dll
```

图15

（通信dll导出函数）

接着此dll会根据资源中的配置信息Sleep一定的时间后再开始后续的操作。后续会再利用自加载的方式将3个通信相关的dll全部加载起来。自加载时用到的函数主要有VirtualAlloc、

RtlMoveMemory。木马还会创建名为“Microsoft-Windows-DiskDiagnosticResolver“的任务计划，达到常驻的目的。

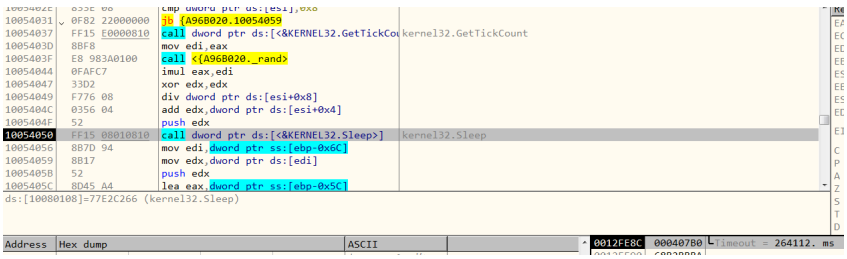


图16

```
rdata:1002C708 ;
rdata:1002C708 ; Export Address Table for DnsProvider.dll
rdata:1002C708 ;
rdata:1002C708 off_1002C708    dd rva CreateInstance   ; DATA XREF: .rdata:1002C6FC↑o
rdata:1002C70C ;
rdata:1002C70C ; Export Names Table for DnsProvider.dll
rdata:1002C70C ;
rdata:1002C70C off_1002C70C    dd rva aCreateinstance  ; DATA XREF: .rdata:1002C700↑o
rdata:1002C70C                                         ; "CreateInstance"
rdata:1002C710 ;
rdata:1002C710 ; Export Ordinals Table for DnsProvider.dll
```

（根据资源中的配置的信息，sleep一段时间）

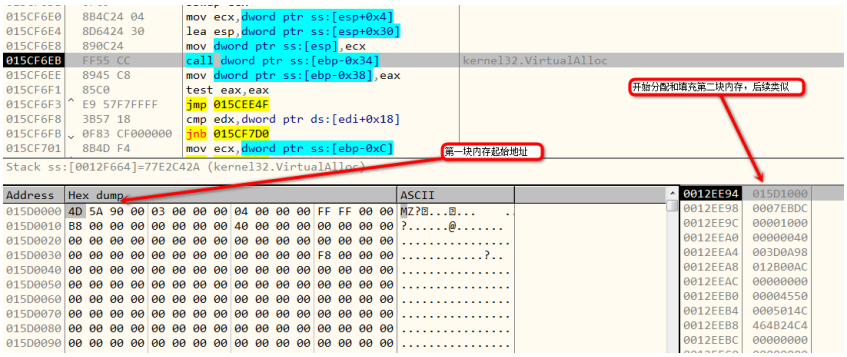


图17

（自加载时分配内存）

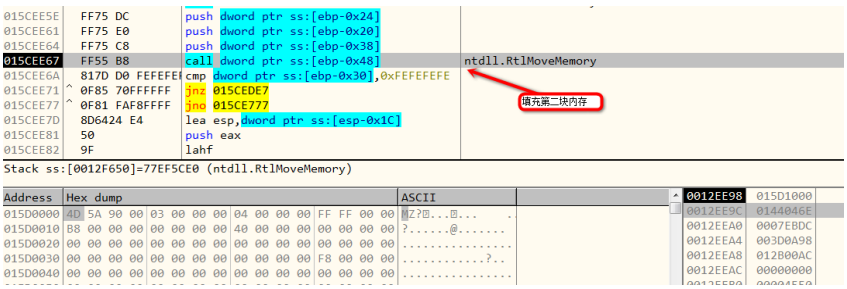


图18

（自加载时填充内存）

木马会根据资源中的c2，使用DGA（域名生成算法）生成通信时的域名信息，木马上线时会将用户名、计算机名、操作系统等信息加密后上报给c2。

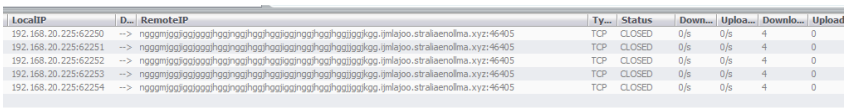


图19

（DGA得到的通信域名）

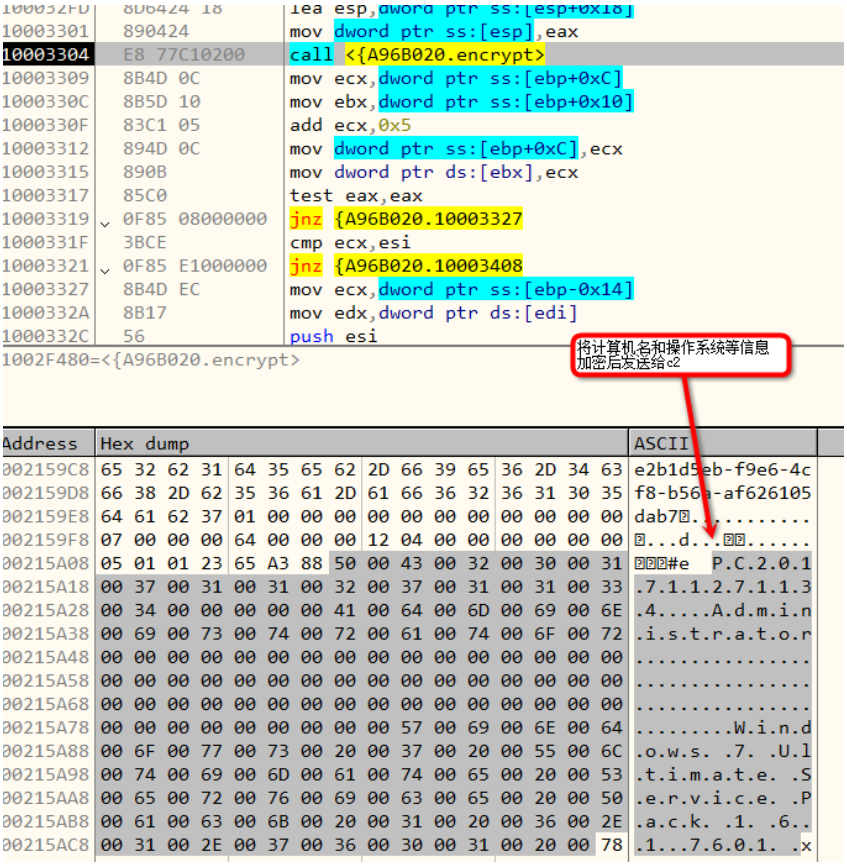


图20

(上线时的第一个明文包)

木马上线后，根据服务器下发的指令执行相应的功能，主要功能有：

l 文件操作，比如创建文件或目录、删除文件或目录、查找文件

l 注册表读写

l 远程执行代码，比如创建进程、执行dll等

l 设置环境变量

0x4 溯源

从该RAT通信的C&C地址154.16.138.89在腾讯御见威胁情报中心平台进行反查，得到下列结果：

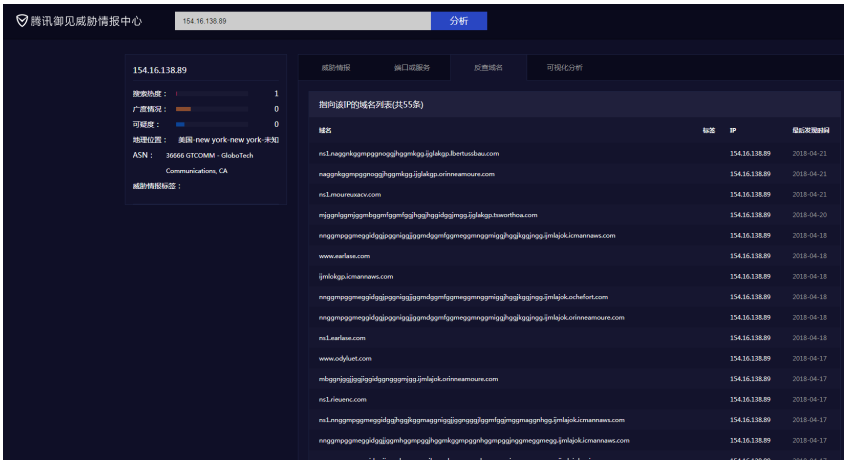


图21

随便选一个域名orinneamoure.com继续反查：

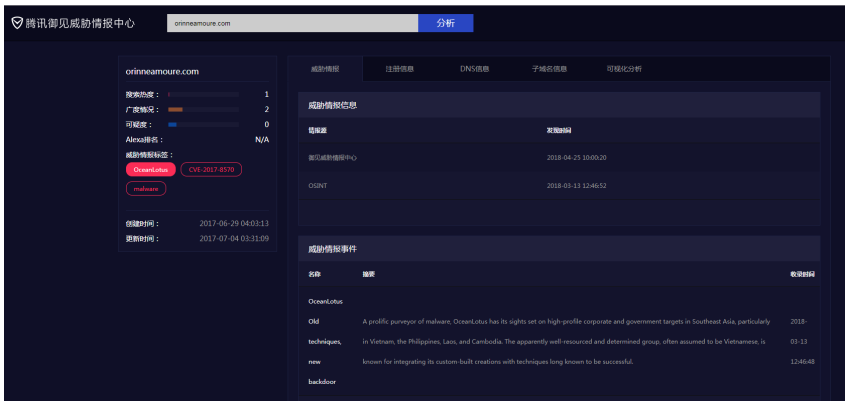


图22

可以发现，该域名被腾讯御见威胁情报平台为标注为海莲花。而该域名也在之前友商对海莲花的报告中披露过。此外，该攻击使用的技术、网络通信协议等和以往的海莲花的攻击样本进行比对也完全一致。因而我们可以确认，该次攻击属于海莲花APT团队所为。

0x5 总结

从上文的分析可以看出该组织在漏洞利用、白加黑利用技术、代码混淆等方面都有着很深的技术积累。后门木马不落地直接内存执行、签名程序白利用、shellcode隐藏可执行文件、多变的网络通信等技术手段大大增加了杀软的查杀难度。因此，我们提醒政府、企业等广大用户，切勿随意打开来历不明的文档，同时安装安全软件。

目前，腾讯御界高级威胁检测系统已经可以检测该轮攻击的连接行为。御界高级威胁检测系统，是基于腾讯反病毒实验室的安全能力、依托腾讯在云和端的海量数据，研发出的独特威胁情报和恶意检测模型系统。

凭借基于行为的防护和智能模型两大核心能力，御界高级威胁检测系统可高效检测未知威胁，并通过对企业内外网边界处网络流量的分析，感知漏洞的利用和攻击。通过部署御界高级威胁检测系统，及时感知恶意流量，检测钓鱼网址和远控服务器地址在企业网络中的访问情况，保护企业网络安全。

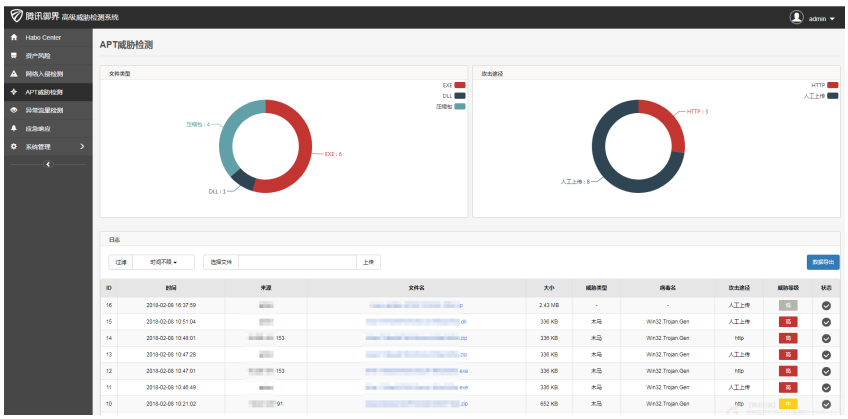


图23

0x6 附录(IOCs):

Hash:

02AE075DA4FB2A6D38CE06F8F40E397E (Document_GPI Invitation-UNSOOC China.doc)

D7C172D4A88573B7E373F2B666C011AC(GPI Invitation-UNSOOC China.doc)

72A5AD375401F33A5079CAEE18884C9D ({92BA1818-0119-4F79-874E-E3BF79C355B8}.dll)

79D06DD20768FD8CD4A043833C1F2D4B ({A96B020F-0000-466F-A96D-A91BBF8EAC96}.dll)\

EC505565E4CB5A22BFD3F63E4AD83FF3(HttpProv.dll)

2559738D1BD4A999126F900C7357B759(HttpProv.dll)

2DFAEDD9265642E430E6635F210FABB4(DnsProvider.dll)

F775CC387A55831386E44DD00EF9723E(rastls.dll)

B10F93CDBCDF43D4C5C5770872E239F4(OUTLFLTR.DAT)

C&C:

andreagahuvrauvin.com

byronorenstein.com

straliaenollma.xyz

dieordaunt.com
christienoll.xyz
illagedrivestralia.xyz
154.16.99.85
154.16.47.41
154.16.138.89

注册表：

“SOFTWARE\App\AppXbf13d4ea2945444d8b13e2121cb6b663\DefaultIcon”

“SOFTWARE\App\AppXbf13d4ea2945444d8b13e2121cb6b663\Application”

相关文章

企业未修复Apache Struts 2 漏洞 致Web服务器被批量入

黑产竞争激烈：一台服务器遭遇两拨黑客进攻

PhotoMiner木马挖矿收入8900万 已成门罗币“黄金矿工”