通过explorer.exe的后渗透权限维持

Fay / 2021-05-08 14:18:59 / 浏览数 281

对windows explorer的一点探究

> 这里记录的是过程...有点乱，发现什么就写了什么，没啥顺序，建议直接看利用思路

想法来自于用clsid修改文件夹从而得到一个新图标的应用时，比如 `回收站.{645FF040-5081-101B-9F08-00AA002F954E}` 会得到一个回收站，但查看文件夹属性的时候还是可以看到它的名称为我们命名的名称，开始很迷惑，最近com接触的比较多，进行了点分析，大概懂发生了什么。

automaticDestinations-ms

“跳转列表”在Windows 7中引入的文件类型，包含最近打开的项目的快捷方式，可能是一个文字处理器最近访问的文档，用于图像编辑器或其他最近使用的项目的图像文件，可以从访问固定的应用程序在任务栏上近期部分（右键单击固定项目和浏览近期部分）。
AUTOMATICDESTINATIONS -MS文件被保存到 `C:\Users\用户名\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations`

explorer的运行似乎很依赖于注册表，

先是查找了这两个注册表项
`HKCU\Software\Classes\CLSID\{645FF040-5081-101B-9F08-00AA002F954E}`
`计算机\HKEY_CLASSES_ROOT\CLSID\{645FF040-5081-101B-9F08-00AA002F954E}`

判断是文件夹后还用了以下命令去提前获取目录下所有文件

- queryopen 文件
- querydirectory 文件夹 列出了文件夹下文件

还有个 `679F85CB-0220-4080-B29B-5540CC05AAB6` 似乎是固定到主菜单的com组件

然后有一段复读queryopenkey去找不存在的键值，找不到就queryclosekey，然后隔一段又继续复读，真受不了windows程序员了...

然后到处找相关的内容，比如name，instance，command，inprocserver32之类的，像是爆破目录一样，确定这个clsid有什么作用以及执行的方式

顺藤摸瓜找到了它在寻找的东西

- {a015411a-f97d-4ef3-8425-8a38d022aebc} ---clsid find-executable ----我的电脑
- {48527bb3-e8de-450b-8910-8c4099cb8624} ---Empty Recycle Bin verb invocation ----回收站

根据名字猜测应该是某种寻找欲执行的clsid的东西

打开则与这个有关

- `计算机\HKEY_CLASSES_ROOT\Folder\shell\opennewprocess\command`
- `计算机\HKEY_CLASSES_ROOT\Folder\shell\opennewtab\command`

都指向这个com组件

- {11DBB47C-A525-400B-9E80-A54615A090C0} --CLSID_ExecuteFolder ---explorer

然后在注册表中explorer的功能点

- `计算机\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\MyComputer\NameSpace\` 这是在我的电脑中显示的内容
- 找到个百度网盘的clsid，{679F137C-3162-45da-BE3C-2F9C3D093F64}
- `计算机\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderTypes\` 存了文件类型
- ::{031E4825-7B94-4dc3-B131-E946B44C8DD5} ParsingName
- Master Control.{ED7BA470-8E54-465E-825C-99712043E01C}
- `计算机\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Desktop\NameSpace\` 桌面显示

在看回收站的注册表时，惊奇的发现回收站的注册表里有第三方软件的东西？！要知道回收站等系统组件的权限到了Trustinstaller，即使提到system人家也不认，于是力求百度，找到了前人写的修改Trustinstaller注册表的代码，读了一遍，才想起拿到SeBackupPrivilege，SeRestorePrivilege就具备了修改本计算机上所有文件的权限...毕竟注册表也算是文件目录。不能活学活用就会忘记...

所以其实提到admin，开启两个特权，应该就能开始胡作非为了。

利用思路

explorer能自动执行是猜测的，因为之前在修改文件名为那个有问题的clsid后（dlna，貌似在win10被阉割了，所以在启动时会导致崩溃），不仅是打开会崩溃，加载当前目录，或者上级目录，也会导致崩溃（具体几级目录会崩溃不一定），所以猜测是为了提升加载速度提前加载目录下的文件

其次，它原本想自动执行是因为这些clsid对应的其实算是特殊文件夹，通过 `::{645FF040-5081-101B-9F08-00AA002F954E}` 这种格式就能执行，具体参照微软文档说明 微软文档传送门。但是我们也能这样去执行我们的dll。

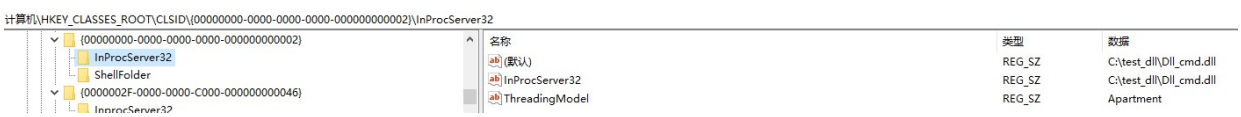
所以思路就回到了如何执行我们的dll身上，也就是要让它能解析我们的恶意dll的clsid，我稍微查了一下 `::{clsid}` 这种格式的命令有什么用，它来自于IShellFolder的ParseDisplayName，用于解析这种::{clsid}路径。

我们说是需要调用函数自己造一个shellfolder，但其实我们对比注册表的同异，发现多的就是clsid下一个shellfolder的项，通用的是里面有个attributes的值，虽然每个shellfolder的attributes值都不同，但我们随便选了一个填了个相同的，就能够解析了。

总结步骤：

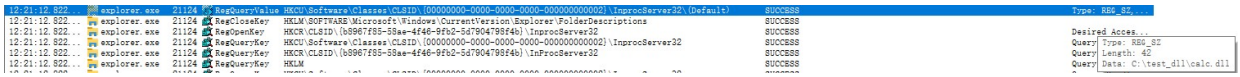
1. 在 `计算机\HKEY_CLASSES_ROOT\CLSID\` 下创建一个独一无二的clsid(就相当于regsvr32注册clsid，但是用regsvr32注册还得自己找)
2. 创建InProcServer32(注册表貌似没有大小写区别，但是最好还是按标准来)，默认值填我们dll的路径，再生成一个字符串值 `ThreadingModel` 填 `Apartiment` (还有个值是 `Both` ，暂不清楚区别)
3. 创建ShellFolder项，和一个 `Attributes` 值。 `Attributes` 随便填一个
4. 复制这些注册表项到 `HKCU\Software\Classes\CLSID\` 下(好像会默认复制过去，修改一边另一边也会同步修改保持一致，如果没有还是自己手动改一下，搜索键值的时候会优先搜索这个注册表目录)
5. 通过几个方式执行clsid来执行我们的恶意dll

注册表项



几种执行方式

- 新建文件夹，后缀改为 `.{对应clsid}` ，就会在点击文件夹的时候用rundll32执行我们的dll



找到了我们指定的clsid

用rundll32执行dll

- explorer路径栏输入 `shell:::{clsid}` 或者直接在启动中输入 `explorer.exe /e,::{clsid}`
- library-ms,windows库文件

结构如下，是个xml格式文件

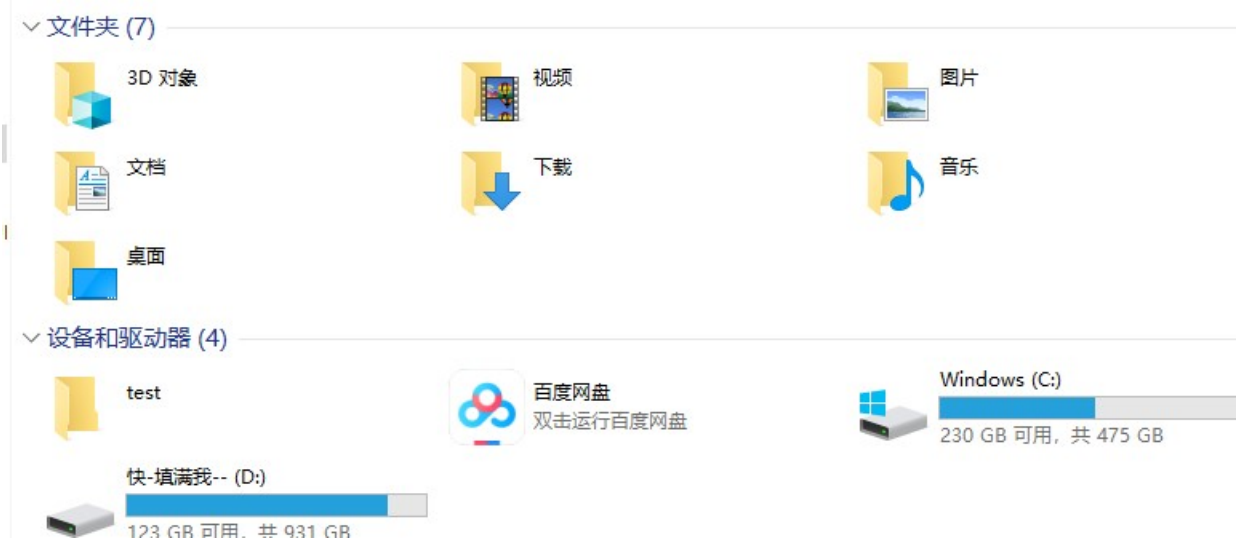
修改或者增加一个这个区域

```
<searchConnectorDescription>
      <description>@shell32.dll,-34577</description>
      <isDefaultSaveLocation>true</isDefaultSaveLocation>
      <isSupported>true</isSupported>
      <simpleLocation>
        <url>shell:::{00000000-0000-0000-0000-000000000002}</url>
      </simpleLocation>
    </searchConnectorDescription>
```

放在目录下就能自动加载了

- 添加
  到 `计算机\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\MyComputer\NameSpace` 下，
  它就会在explorer首页出现这个目录

效果如下，但是不能和shellfolder重合，也就是没有shellfolder这个项才会显示

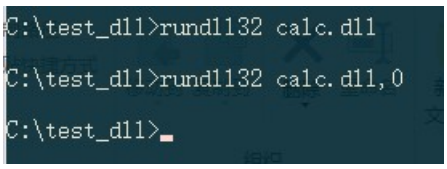


- 再狠一点可以调用`IKnownFolderManager::RegisterFolder`造一个knownfolder

这是个com组件，提供了`RegisterFolder`调用方法，可以造一个目录放在explorer的侧边栏，同样也可以指向我们的dll，我猜也可以自己改注册表达到效果，但还没研究过程。

综上，我觉得这种这种权限维持的方式还是有点意思的，一是能自动执行，二是图标和目录名都能自己控制，三是能放在一般人不上心或者不太警惕的位置。

> 注意，由于explorer默认权限是用户权限，所以explorer加载后门dll时用的也是用户权限，还是需要一步提权步骤。
>
> 虽然rundll32执行了，但是我自己在`dll_process_attach`和`dll_thread_attach`中写的winexec执行cmd和calc都没有弹窗出来..不知道发生了啥，但是确实能够执行，还能找到由explorer.exe生成的dllhost.exe进程。
>
> rundll32直接执行我的dll也不会弹窗...要跟一个参数才能弹窗，感觉是自己编写的dll的问题，暂时不知道怎么修改。rundll32还有`-sta {clsid}`参数可以自动搜索注册表中对应的clsid的localserver32和InProcServer32中的dll地址执行。



其他想法

有一说一这个思路有点像dll劫持，但是也不同，毕竟是创建自己的目录，利用它来加载。

但我做的时候看到其他文件就在想，劫持已存在的不也可以么

事实证明确实可以，修改对应的com组件的InProcServer32为我们的就行了。但是最好不要劫持常用组件，比如上文经常用到的回收站recycle bin，最好选择一些冷门的组件来劫持。

所需要的com组件都在HKCU下修改即可，所以本机用户都有权限修改，更别说可以提权到system来执行系列操作了。

想法plus：既然explorer加载的com可以劫持，那么其他应用加载的com组件也可以劫持，比如我看到有项目劫持outlook的，也就是任务栏常驻的那个邮箱。还是那句话，脚本是死的，方法是活的；懂得变通。

上一篇： 记一次对Tp二开的源码审计（Php审计）

下一篇： Stowaway 2.0来了～

0 条回复

动动手指，沙发就是你的了！

登录 后跟帖