# 36 统计网络，安全防火墙

更新时间：2019-08-01 11:34:58

> 只有在那崎岖的小路上不畏艰险奋勇攀登的人，才有希望达到光辉的顶点。
>
> ——马克思



**内容简介**

1. 前言
2. netstat：网络统计
3. iptables / nftables ：防火墙
4. 总结
5. 第四部分第七课预告

**1. 前言**

上一课 带你玩转Linux和Shell编程 | 第四部分第五课：IP地址和分析网络 中，我们了解了 IP 地址和域名的知识，还学习了 ifconfig 命令。

这一课我们接着来学习 netstat 和 iptables 这两个很强大的命令。

**2. netstat：网络统计**

netstat 命令很好记，它由两部分组成：net 和 stat。

net 是 network 的缩写，表示“网络”。stat 是 statistics 的缩写，表示“统计”。所以顾名思义就是“对网络信息进行统计”啦。

假如你没有一些网络方面的知识，那么 netstat 命令的输出可能难以理解，但是也没那么难。假如你要了解你的电脑正在网络上做什么，那么 netstat 是不二选择。

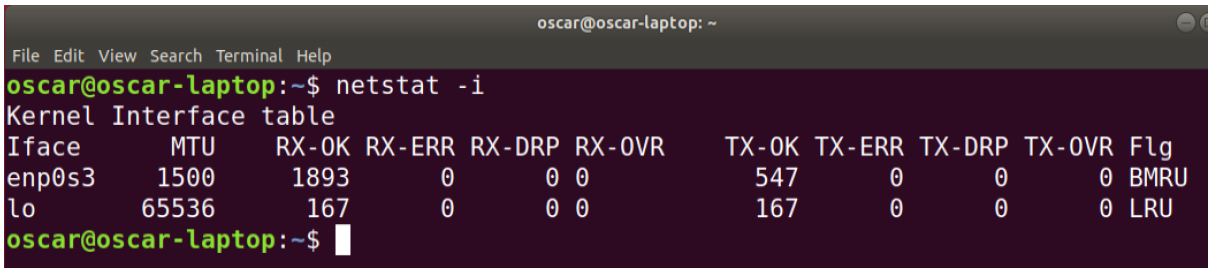
netstat 可以显示很多信息，但是我们可以用参数来控制显示信息的种类和样式。下面介绍几个常用的参数吧。

## netstat -i：网络接口的统计信息

首先，试试 i 参数吧：

```
netstat -i
```

会显示一张统计列表，列出你电脑的所有网络接口的一些统计信息：




可以看到，列出了两行信息：enp0s3 和 lo 。

上一课中我们在讲解 ifconfig 时，已经分析过，其中 enp0s3 是新版本的 Ethernet（以太网）接口的名字，不再是 eth0。lo 则和旧版一样，还是表示 Local Loopback（本地回环）。

RX 是 receive（表示“接收”）的缩写，TX 是 transmit（表示“发送”）的缩写。

- RX-OK：在此接口接收的包中正确的包数。OK 表示“没问题，好的”；
- RX-ERR：在此接口接收的包中错误的包数。ERR 是 error 的缩写，表示“错误”；
- RX-DRP：在此接口接收的包中丢弃的包数。DRP 是 drop 的缩写，表示“丢掉”；
- RX-OVR：在此接口接收的包中没能接收的包数。OVR 是 over 的缩写，表示“结束”。



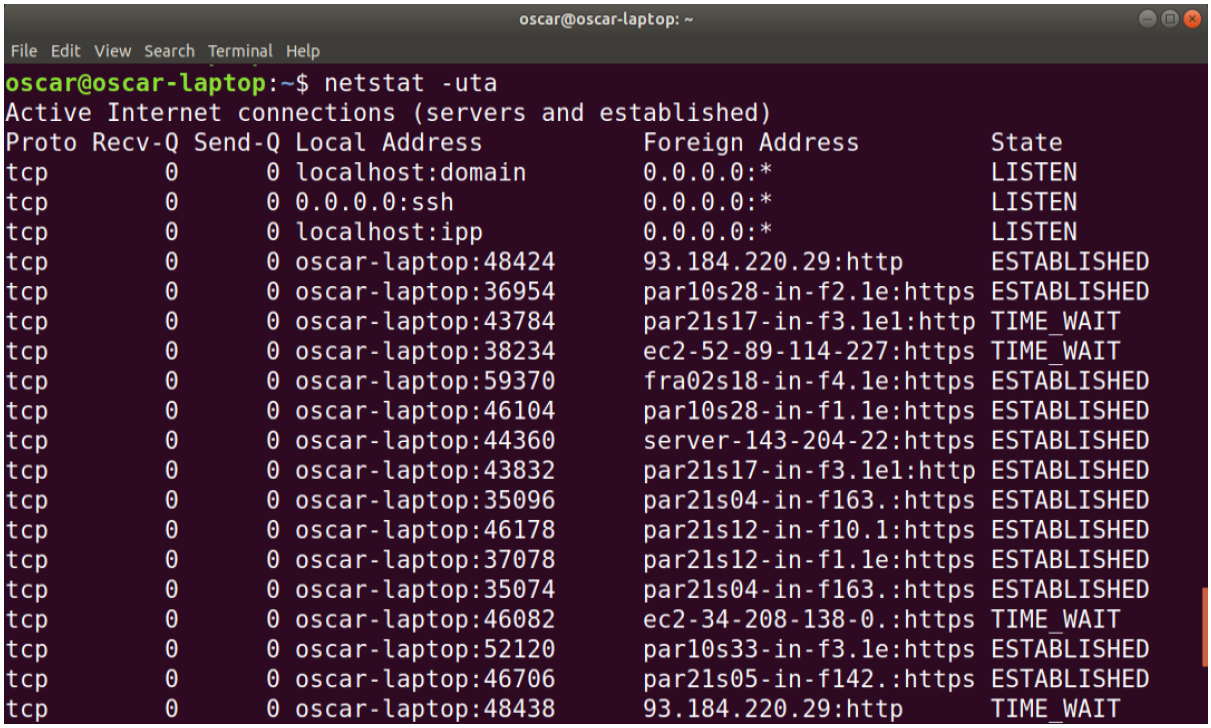
类似的，TX-OK、TX-ERR、TX-DR 和 TX-OVR 则表示在此接口放送的包中对应的包数。

MTU 是 Maximum Transmission Unit 的缩写，表示“最大传输单元”，是指一种通信协议的某一层上面所能通过的最大数据包大小（以字节为单位）。

## netstat -uta：列出所有开启的连接

运行：

```
netstat -uta
```




参数 uta 分别表示：

- -u : 显示 UDP 连接（u 是 udp 的首字母）
- -t : 显示 TCP 连接（t 是 tcp 的首字母）
- -a : 不论连接的状态如何，都显示（a 是 all 的首字母）



TCP 和 UDP 是两种不同的协议，用于在网络上传输数据。

UDP（User Datagram Protocol，“用户数据报协议”）一般用于网络游戏，音频通讯（例如 Skype）。除此之外，一般来说 TCP（Transmission Control Protocol，“传输控制协议”）是最常用的。一般在互联网上都是用 TCP/IP 协议。

我们也可以只显示 TCP 连接的信息：

```
netstat -ta
```

或者只显示 UDP 连接的信息（不常用）：

```
netstat -ua
```

再来看看上面图片中 state（“状态”）那一列的信息，有好几种不同状态：

- ESTABLISHED：与远程电脑的连接已建立，establish 是英语“建立”的意思；
- TIME_WAIT : 连接正在等待网络上封包的处理，一旦处理完毕就开始关闭连接。time 是英语“时间”的意思，wait 是英语“等待”的意思；
- CLOSE_WAIT ：远程服务器中止了连接（也许你太久没什么动作，处在不活跃状态）。close 是英语“关闭”的意思；
- CLOSED ：连接没有被使用，关闭了；
- CLOSING ：连接正在关闭，但有些数据还没有发送完毕；
- LISTEN ：监听着可能进入的连接。此时连接还没有被使用。listen 是英语“听”的意思。

当然，状态还不止这几种，其它的可以在 netstat 的命令手册中找到（用 `man netstat` 来查看）。

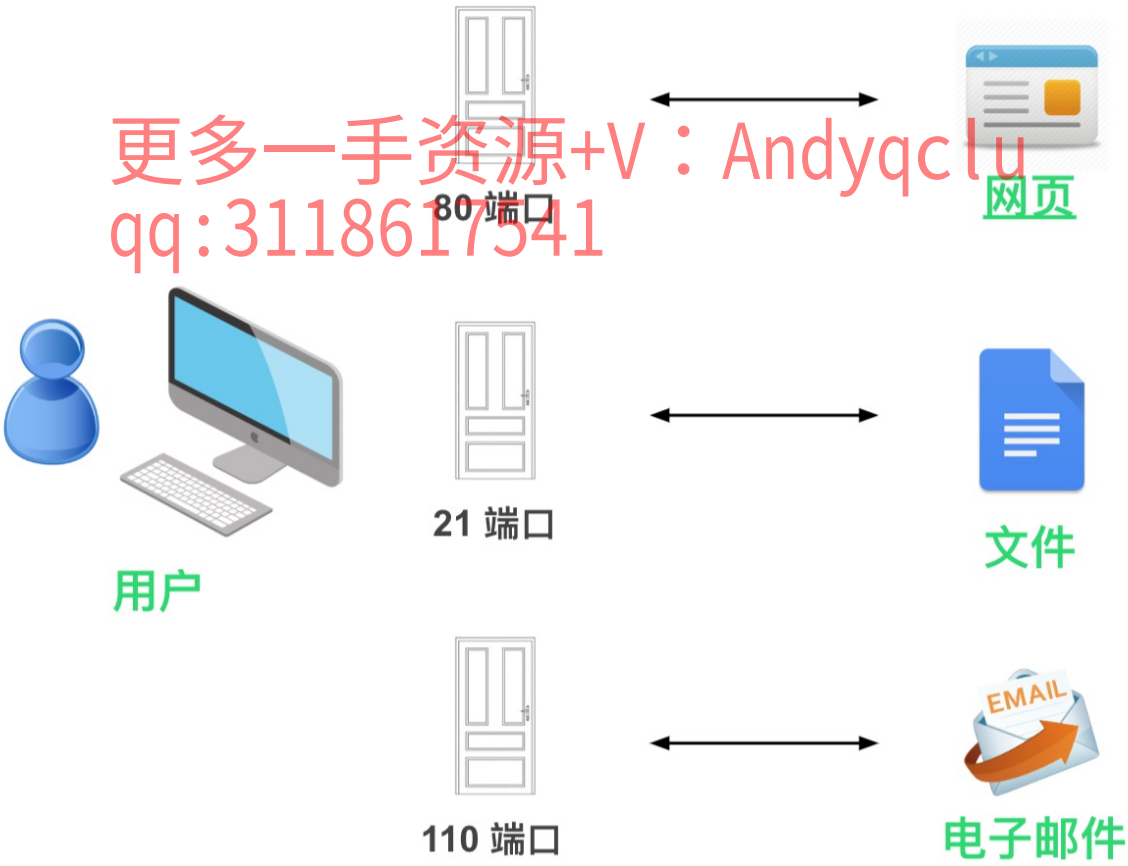
我们再来看看端口的信息，就是上面图片中冒号（:）之后的数据。

事实上，我们连接其它电脑，可以透过不同的端口（port），有点类似门户。比如我去朋友家，可能进他们的厨房门、书房门、地下室门等等。

不同的端口用处不同。进厨房门可能看看做菜如何，进书房门可能一窥书香，进地下室门可能去品品葡萄酒。反正卧室门是不可以随便进的~

> 摘自百度百科：
> “端口”是英文 port 的意译，可以认为是设备与外界通讯交流的出口。
> 端口可分为虚拟端口和物理端口。
> 其中虚拟端口指计算机内部或交换机路由器内的端口，不可见。例如计算机中的 80 端口、21 端口、23 端口等。
> 物理端口又称为接口，是可见端口，例如计算机背板的 RJ45 网口，交换机路由器集线器等 RJ45 端口。电话使用的 RJ11 插口也属于物理端口的范畴。

如下图所示：





- 80 端口是为 HTTP（HyperText Transport Protocol，“超文本传输协议”）开放的，此为上网冲浪使用次数最多的协议，主要用于 WWW（World Wide Web，“万维网”）传输信息的协议。可以通过 HTTP 地址（即常说的“网址”）加 `:80` 来访问网站，因为浏览网页服务默认的端口号都是 80，因此只需输入网址即可，不用输入`:80`了；
- 21 端口用于 FTP（File Transfer Protocol，“文件传输协议”）服务，FTP 服务主要是为了在两台计算机之间实现文件的上传与下载。上一课我们学习过 FTP 相当的知识；
- 110 端口是为 POP3（Post Office Protocol version 3 的缩写，表示“邮件协议 第三版”）服务开放的，用于收发

电子邮件。

你可以加上 -n 参数，假如你想让端口信息以数字的形式显示，而不是像前面的截图中那样有点看不懂的状态，比如 http、https、nfs、mysql 等等。

### netstat -lt：列出状态是 LISTEN 的统计信息

```
netstat -lt
```

### netstat -s：列出总结性的统计信息

```
netstat -s
```

还有更多命令选项就用 `man netstat` 来查看吧。

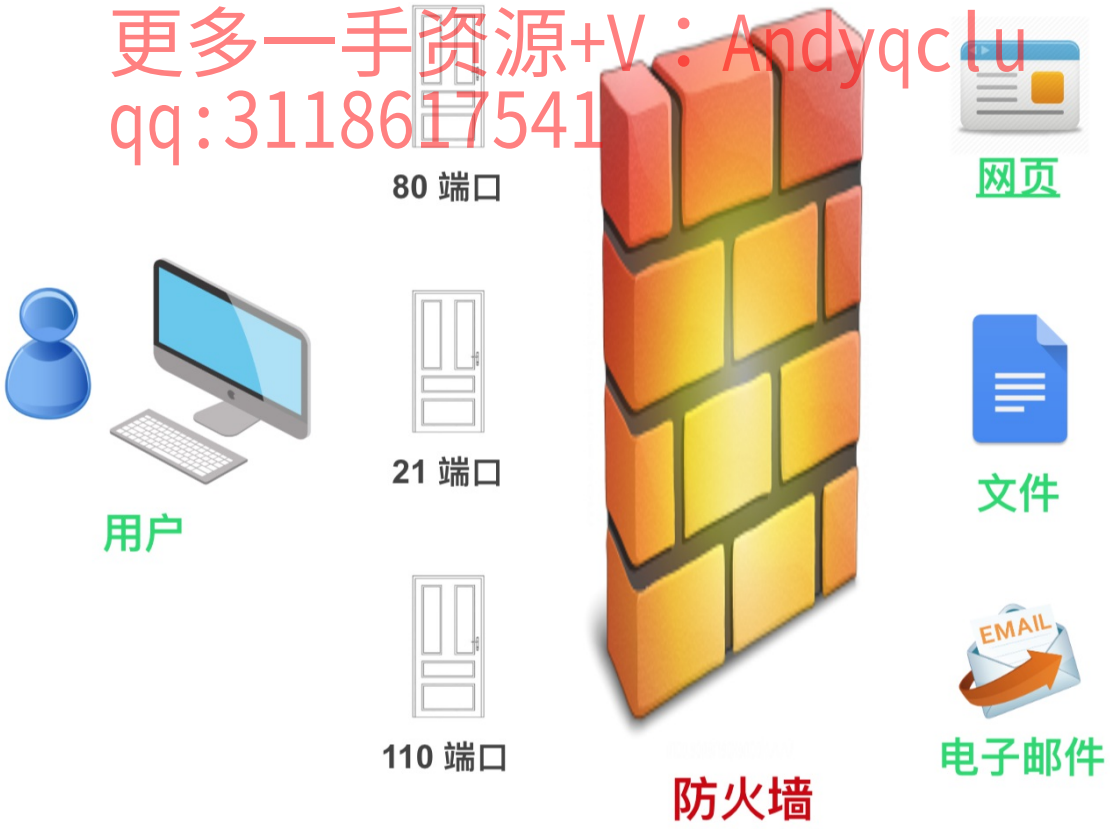
**3. iptables / nftables：防火墙**

现在既然我们已经知道如何分析网络传输，我们就“趁热打铁”，学习如何用防火墙来过滤网络传输吧。

Linux 下比较著名的防火墙是 iptables。它有点年纪了，已经服役十几年。

iptables 命令可以制定一些规则，规定其它电脑可以使用哪些端口来连接你的电脑（对应“入”），以及你的电脑可以连接哪些端口（对应“出”）。也可以通过 IP 地址来过滤。类似下图所示：






例如，我想要拦截所有 FTP 的连接，那么我可以用 iptables 封锁 21 端口。

### 安装 iptables 防火墙

如果没有安装 iptables，需要先安装（我的 Ubuntu 系统已经自带了 iptables 命令）：

```
# CentOS 执行：
sudo yum install iptables
```

```
# Debian / Ubuntu 执行：
sudo apt install iptables
```

## iptables 的使用需要 root 身份

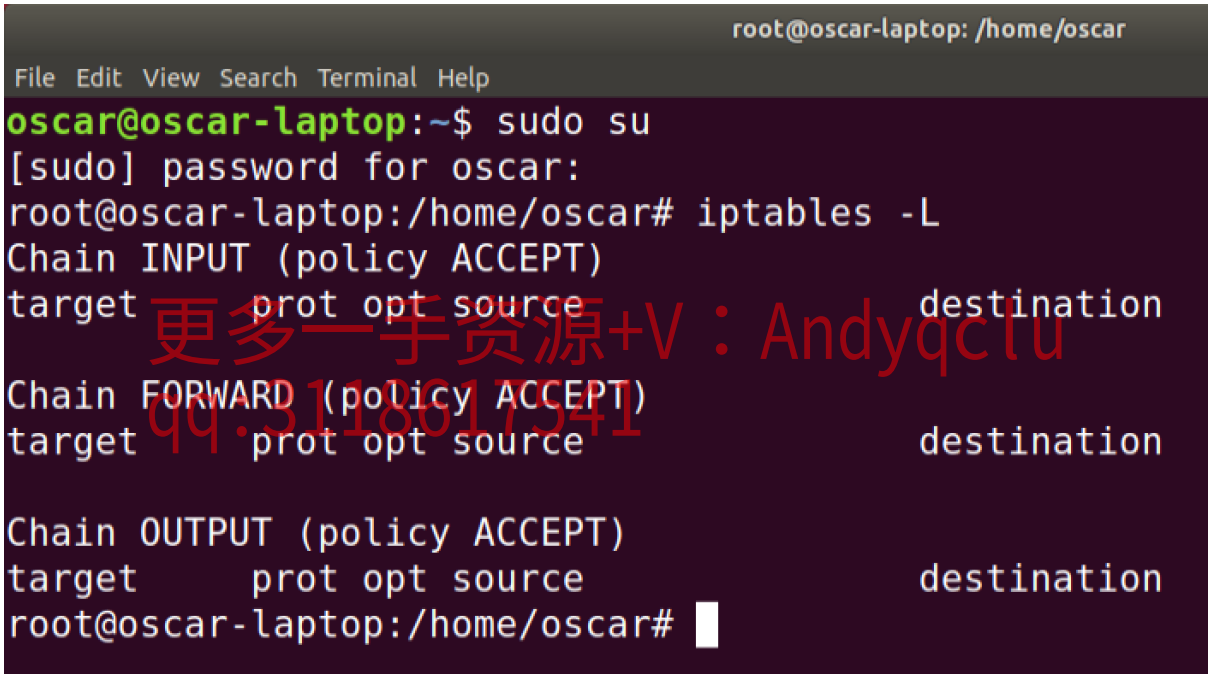
为了使用 iptables，你需要切换到 root 身份：

```
sudo su
```

或者你在下面我们执行的那些命令前每次加 sudo 也是可以的。

## iptables -L：显示所有规则

```
iptables -L
```





可以看到三个区域：

- `Chain INPUT`：对应控制“进入”的网络传输的规则，input 是英语“输入”的意思。
- `Chain FORWARD`：对应控制“转发”的网络传输的规则，forward 是英语“转发”的意思。
- `Chain OUTPUT`：对应控制“出去”的网络传输的规则，output 是英语“输出”的意思。

暂时我们还没有制定任何规则，我们慢慢来学习。

1、清除已有 iptables 规则（慎用）：

```
iptables -F
iptables -X
iptables -Z
```

2、开放指定的端口：

```
# 允许本地回环接口（即运行本机访问本机）
iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
```

```
# 允许已建立的或相关连的通行
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# 允许所有本机向外的访问
iptables -A OUTPUT -j ACCEPT
```

```
# 允许访问 22 端口
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
# 允许访问 80 端口
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

```
# 允许 FTP 服务的 21 和 20 端口
iptables -A INPUT -p tcp --dport 21 -j ACCEPT
iptables -A INPUT -p tcp --dport 20 -j ACCEPT
# 如果有其它端口的话，规则也类似，稍微修改上述语句就行。
```

```
# 禁止其它未允许的规则访问（注意：如果 22 端口未加入允许规则，SSH 链接会直接断开。）
## 1）. 用 DROP 方法
iptables -A INPUT -p tcp -j DROP
## 2）. 用 REJECT 方法
iptables -A INPUT -j REJECT
iptables -A FORWARD -j REJECT
```

3、屏蔽 IP：

```
# 屏蔽单个 IP 的命令是
iptables -I INPUT -s 123.45.6.7 -j DROP
```



```
# 封整个段，即从 123.0.0.1 到 123.255.255.254 的命令
iptables -I INPUT -s 123.0.0.0/8 -j DROP
```

```
# 封 IP 段从 123.45.0.1 到 123.45.255.254 的命令
iptables -I INPUT -s 124.45.0.0/16 -j DROP
```

```
# 封 IP 段从 123.45.6.1 到 123.45.6.254 的命令是
iptables -I INPUT -s 123.45.6.0/24 -j DROP
```

4、查看已添加的 iptables 规则：

```
iptables -L -n
```

5、删除已添加的 iptables 规则：

```
# 将所有 iptables 以序号标记显示，执行：
iptables -L -n --line-numbers
```

```
# 要删除 INPUT 里序号为 8 的规则，执行：
iptables -D INPUT 8
```

6、iptables 的开机启动及规则保存：

CentOS 上可能会存在安装好 iptables 后，iptables 并不开机自动启动，可以执行一下：

```
# 将其加入开机启动
chkconfig --level 345 iptables on
```

CentOS 上可以执行：

```
# 保存规则
service iptables save
```

> Debian / Ubuntu 上 iptables 是不会一直保存规则的。需要按如下步骤进行，让网卡关闭时保存 iptables 规则，启动时加载 iptables 规则：

1. 如果当前用户不是 root，即使使用了 sudo，也会提示你没有权限，无法保存。所以执行本命令，你必须使用 root 用户；
2. 可以使用 `sudo su` 转到 root 用户；
3. 为了重启服务器后，规则自动加载，我们创建如下文件:

```
sudo nano /etc/network/if-pre-up.d/iptables
```

这个 iptables 文件里的初始内容是：

```
#!/bin/bash
iptables-save > /etc/iptables.rules
```

添加执行权限：

```
chmod +x /etc/network/if-pre-up.d/iptables
```



附上基础规则：

```
*filter
:INPUT ACCEPT [106:85568]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [188:168166]
:RH-Firewall-1-INPUT - [0:0]

# 允许本地回环接口(即运行本机访问本机)
-A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT

# 允许已建立的或相关联的通行
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# 允许所有本机向外的访问
-A OUTPUT -j ACCEPT

# 允许 PPTP 拨号到外网
-A INPUT -p tcp -m tcp --dport 1723 -j ACCEPT

# 仅特定主机访问 Rsync 数据同步服务
-A INPUT -s 8.8.8.8/32 -p tcp -m tcp --dport 873 -j ACCEPT

# 仅特定主机访问 WDCP 管理系统
-A INPUT -s 6.6.6.6/32 -p tcp -m tcp --dport 8080 -j ACCEPT

# 允许访问 SSH
-A INPUT -p tcp -m tcp --dport 1622 -j ACCEPT

# 允许访问 FTP
-A INPUT -p tcp -m tcp --dport 21 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 20 -j ACCEPT

# 允许访问网站服务
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT

# 禁止所有未经允许的连接
-A INPUT -p tcp -j DROP

#注意：如果 22 端口未加入允许规则，SSH 链接会直接断开。
#-A INPUT -j REJECT
#-A FORWARD -j REJECTCOMMIT
```



上面的步骤有点麻烦，可以使用以下方法直接载入：

1. 用文本编辑器来创建文件 `sudo nano /etc/iptables.test.rules`，复制上面的规则粘贴到文件中，保存文件；
2. 加载规则，使之生效。注意，iptables 不需要重启，加载一次规则就可以。`sudo iptables-restore < /etc/iptables.test.rules`；
3. 查看最新的配置，应该所有的设置都生效了。`sudo iptables -L -n`；
4. 保存生效的配置，让系统重启的时候自动加载有效配置（iptables 提供了保存当前运行的规则功能）`iptables-save > /etc/iptables.rules`。

上面的操作看着都很复杂，因为我们还没学习脚本语言。第五部分我们就会学习 Shell 脚本。

如果你想提前试试，也可以看 Ubuntu 官方的关于保存和配置开机加载 iptables 规则的文章（英文的）：

https://help.ubuntu.com/community/IptablesHowTo#Saving_iptables 。

> 我们也见识到了，iptables 的配置相当繁复，普通用户简直望而却步。
> 幸好，有一些软件可以帮助我们减轻痛苦。

## UFW - Uncomplicated Firewall

UFW 是 Uncomplicated Firewall 的缩写，uncomplicated 是英语“不复杂的，简单的”的意思，firewall 是“防火墙”的意思。

顾名思义 UFW 这个软件是“简单的防火墙”，比 iptables 简单很多。但 UFW 并不是在每个 Linux 发行版中都有的，幸好 Ubuntu 中自带了。

运行 UFW 需要 root 身份：

```
sudo ufw xxx
```

其中 xxx 表示参数。

- Ubuntu 官方 UFW 文档：https://help.ubuntu.com/community/UFW ；
- 中文 Ubuntu 官方 UFW 文档：UFW使用指南 。

当然了，还有更好的图形界面的 UFW：**GUFW**。

Ubuntu 官方 GUFW 文档：https://help.ubuntu.com/community/Gufw 。

GUFW 一般 Ubuntu 中没有自带，需要安装：

```
sudo apt install gufw
```

安装完之后，运行以下命令来启动：

```
sudo gufw
```




- Status ： 表示“状态”。如果是 OFF，则防火墙没有激活。ON 则防火墙已激活。
- Incoming ： 表示“进来”。
- Outgoing ： 表示“出去”。

其它的选项，可以参考使用手册。

### nftables

> 新的防火墙子系统 / 包过滤引擎 nftables 在 Linux 3.13 中替代了有十多年历史的 iptables。iptables / netfilter 是在 2001 年加入到 2.4 内核中。

诞生于 2008 年的 nftables 设计替代 iptables，它提供了一个更简单的 Kernel ABI (Application Binary Interface)，减少重复代码，改进错误报告，更有效支持过滤规则。

除了 iptables，nftables 还将替代 ip6tables、arptables 和 ebtables。Linux 内核的第一代包过滤机制是 ipfwadm（1.2.1 内核，1995 年），之后是 ipchains（1999 年），iptables。nftables 是第四代。

如果你的 Ubuntu 里面没有 nftables，那就运行下面的命令来安装：

```
sudo apt install nftables
```

nftables 引入了一个新的命令行工具 nft。nft 是 iptables 及其衍生指令（ip6tables，arptables）的超集。nft 的运行也需要 root 权限。

同时，nft 拥有完全不同的语法。如果你习惯于 iptables，这是个不好的消息。但是有一个兼容层允许你使用 iptables，而过滤是由内核中的 nftables 完成的。

但是基本的原理是类似的，nftables 比 iptables 更方便，使用更有效率，可以把一些命令合并。

例如，你想用 iptables 记录并丢弃一个包，你必须写两条规则，一条记录，一条丢弃：

```
iptables -A FORWARD -p tcp --dport 22 -j LOG
iptables -A FORWARD -p tcp --dport 22 -j DROP
```



使用 nft，你可以把两个目标合并到一起：

```
nft add rule filter forward tcp dport 22 log drop
```

所以，假如你的 Linux 内核版本是 3.13 之前的，那就继续使用 iptables；如果 Linux 内核版本是 3.13 版之后，那就用 nftables吧（其实 nftables 要从 Linux 内核版本 3.15 版才开始比较成熟）。

nft 还有更多命令选项，就用 `man iptables` 和 `man nftables` / `man nft` 来查看吧。

**4. 总结**

1. netstat 命令会列出你电脑上打开的连接，说明当下哪些端口正打开着，一个端口就好比引导出入你电脑的门户。
2. 可以用 iptables 命令来拦截进入某些端口的连接，它是一个很不错的防火墙。但是配置比较复杂。iptables 配置很繁琐，可以用 UFW 软件来减轻压力。从 Linux 3.13 开始，nftables 命令替代了 iptables。

今天的课就到这里，一起加油吧！

35 IP地址和分析网络

37 源码编译，安装便利