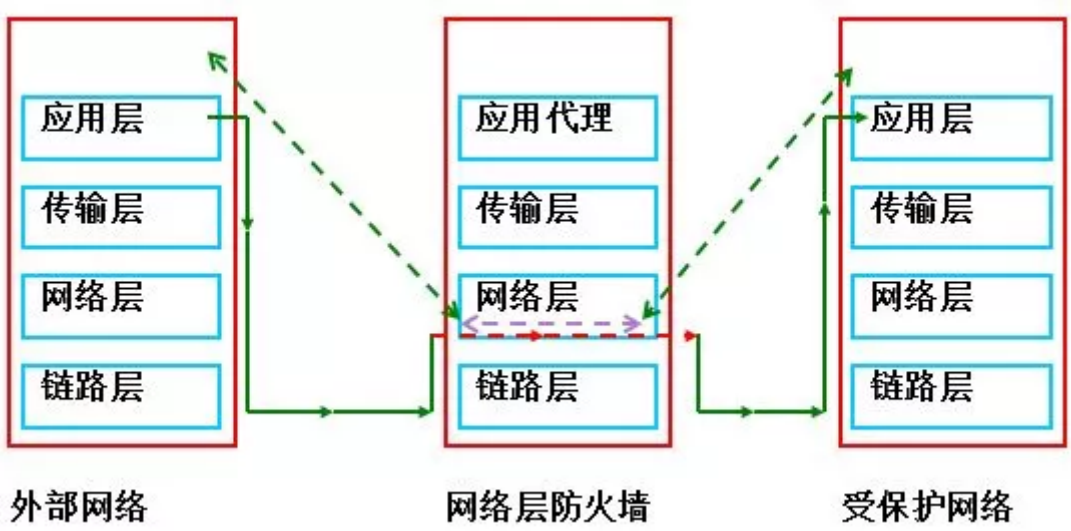
防火墙

iptables简介

iptables命令是Linux上常用的防火墙软件，是netfilter项目的一部分。可以直接配置，也可以通过许多前端和图形界面配置。所谓防火墙，实质上是指由软硬件组合成的一个在内外网之间构造的一种保护屏障，它是一种隔离技术。因此从物理上区分，可以分为软件和硬件防火墙，从逻辑上区分，可以分为主机和网络防火墙，而我们现在要讲到的iptables，是属于防火墙中的软件防火墙的范畴，但它只是一个命令行工具或者说是一种客户端代理，并不是真正的防火墙，用户通过这个代理，把安全设定执行到真正的防火墙框架中，这个框架叫做Netfilter。



原理

Netfilter是Linux中的一个子项目，它的主要功能是进行数据包过滤、连接跟踪、地址转换等，而iptables则是netfilter提供的对用户数据包进行过滤、修改等操作的一种命令行工具，当数据包通过网卡进入内核时，它得先经过iptables的五条链，这些链都有相应的处理数据包的规则，而我们正是通过在这些链上设置规则来控制管理数据包，从而达到防火墙的功能。每当数据包到达一个链时，iptables就会从链中的所有规则逐一开始校验该数据包是否符合规则中限定的条件。若满足，系统就会根据每个规则定义的方法来处理该数据包；若不满足，iptables则继续检查下一条规则，如果该数据包不符合链中的任意规则，iptables则会该链的默认策略去处理该数据包。

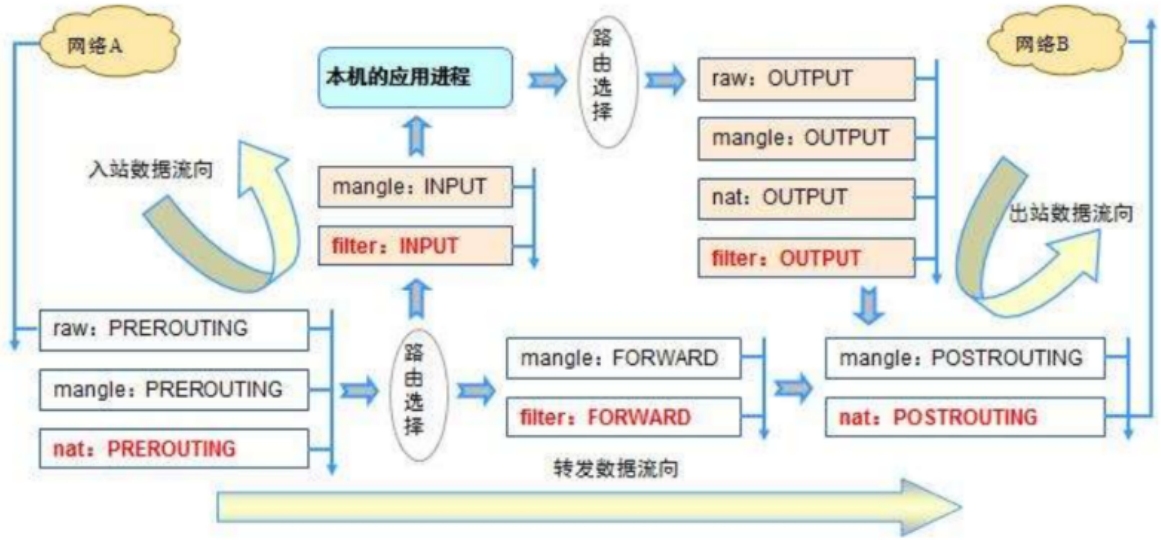
四个表

iptables的结构是由tables组成，而tables是由链组成，链又是由具体的规则组成。因此我们在编写iptables的规则时，要先指定表，再指定链。tables的作用是区分不同功能的规则，并且存储这些规则。

tables的类型分别有：

表名	功能	优先级	内建链
filter	数据包的过滤功能	4	INPUT OUTPUT FORWARD
nat	网络地址转换功能	3	PREROUTING POSTROUTING OUTPUT
mangle	数据包的修改功能，拆解数据包，进行修改，然后重新组装成数据包	2	PREROUTING OUTPUT FORWARD INPUT POSTROUTING
raw	数据跟踪	1	PREROUTING OUTPUT

iptables过滤封包流程



```
-->PREROUTING-->[ROUTE]-->FORWARD-->POSTROUTING-->
   mangle      |     mangle                    ^ mangle
    nat        |     filter          |  nat
               |                     |
               |                     |
               v                     |
             INPUT                 OUTPUT
               | mangle                ^ mangle
               | filter                |  nat
               v ---------------->local---------->| filter
```

总结： 整体数据包分两类： 1、发给防火墙本身的数据包； 2、需要经过防火墙的数据包

① 当一个数据包进入网卡时，它首先进入PREROUTING链，内核根据数据包目的IP判断是否需要转送出去。

② 如果数据包就是进入本机的，它就会沿着图向下移动，到达INPUT链。数据包到了INPUT链后，任何进程都会收到它。

本机上运行的程序可以发送数据包，这些数据包会经过OUTPUT链，然后到达POSTROUTING链输出。

③ 如果数据包是要转发出去的，且内核允许转发，数据包就会如图所示向右移动，经过FORWARD链，然后到达POSTROUTING链输出。

总结： 整体数据包分两类： 1、发给防火墙本身的数据包； 2、需要经过防火墙的数据包

注意：规则的次序非常关键，谁的规则越严格，应该放的越靠前，而检查规则的时候，是按照从上往下的方式进行检查的。

表间的优顺序：

filter：一般的过滤功能
nat:用于nat功能（端口映射，地址映射等）
mangle:用于对特定数据包的修改
raw:优先级最高，设置raw时一般是为了不再让iptables做数据包的链接跟踪处理，提高性能

五条链

链，也称为钩子函数，它是一系列规则的一个组合，当数据包经过这些钩子函数时，它必须完全匹配每一个钩子函数中的所有规则，方能进入下一个钩子函数。
钩子函数的类型分别有：

链类型	作用域
PREROUTING	数据包进入路由表之前
INPUT	通过路由表后目的地为本机
OUTPUT	由本机产生，向外转发
FORWARD	通过路由表后，目的地不为本机
POSTROUTING	发送到网卡接口之前

语法

```
iptables(选项)(参数)
选项
-t <表>：指定要操纵的表；
-A：向规则链中添加条目；
-D：从规则链中删除条目；
-I：向规则链中插入条目；
-R：替换规则链中的条目；
-L：显示规则链中已有的条目；
-F：清除规则链中已有的条目；
-Z：清空规则链中的数据包计算器和字节计数器；
-N：创建新的用户自定义规则链；
-P：定义规则链中的默认目标；
-h：显示帮助信息；
-p：指定要匹配的数据包协议类型；
-s：指定要匹配的数据包源ip地址；
```

```
-j<目标>：指定要跳转的目标；
-i<网络接口>：指定数据包进入本机的网络接口；
-o<网络接口>：指定数据包要离开本机所使用的网络接口。
```

iptables命令选项输入顺序：

iptables -t 表名 <-A/I/D/R> 规则链名 [规则号] <-i/o 网卡名> -p 协议名 <-s 源IP/源子网> --sport 源端口 <-d 目标IP/目标子网> --dport 目标端口 -j 动作

表名包括：

> raw：高级功能，如：网址过滤。
> mangle：数据包修改（QOS），用于实现服务质量。
> net：地址转换，用于网关路由器。
> filter：包过滤，用于防火墙规则。

规则链名包括：

INPUT链：处理输入数据包。

> OUTPUT链：处理输出数据包。
> PORWARD链：处理转发数据包。
> PREROUTING链：用于目标地址转换（DNAT）。
> POSTOUTING链：用于源地址转换（SNAT）。

动作包括：

> ACCEPT：接收数据包。
> REJECT：拒绝数据包。
> DROP：丢弃数据包。
> REDIRECT：重定向、映射、透明代理。
> SNAT：源地址转换。
> DNAT：目标地址转换。
> MASQUERADE：IP伪装（NAT），用于ADSL。
> LOG：日志记录。

实例

清除已有iptables规则

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

-F 是清空指定某个 chains 内所有的 rule 设定。比方 iptables -F -t filter，那就是把 filter table 内所有的INPUT/OUTPUT/FORWARD chain 设定的规则都清空。

-X 是删除使用者自订 table 项目，一般使用 iptables -N xxx 新增自订 chain 后，可以使用 iptables -X xxx 删除。

-Z：清空规则链中的数据包计算器和字节计数器。

开放指定的端口

```
iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
            #允许本地回环接口(即运行本机访问本机)
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT    #允许已建立的或相关连的通行
iptables -A OUTPUT -j ACCEPT          #允许所有本机向外的访问
iptables -A INPUT -p tcp --dport 22 -j ACCEPT    #允许访问22端口
iptables -A INPUT -p tcp --dport 80 -j ACCEPT    #允许访问80端口
iptables -A INPUT -p tcp --dport 21 -j ACCEPT    #允许FTP服务的21端口
iptables -A INPUT -p tcp --dport 20 -j ACCEPT    #允许FTP服务的20端口
iptables -A INPUT -j REJECT       #禁止其他未允许的规则访问
iptables -A FORWARD -j REJECT     #禁止其他未允许的规则访问
```

屏蔽IP

```
iptables -I INPUT -s 123.45.6.7 -j DROP        #屏蔽单个IP的命令
iptables -I INPUT -s 123.0.0.0/8 -j DROP       #封整个段即从123.0.0.1到123.255.255.254的命令
iptables -I INPUT -s 124.45.0.0/16 -j DROP     #封IP段即从123.45.0.1到123.45.255.254的命令
iptables -I INPUT -s 123.45.6.0/24 -j DROP     #封IP段即从123.45.6.1到123.45.6.254的命令
```

查看已添加的iptables规则

```
iptables -L -n -v
Chain INPUT (policy DROP 48106 packets, 2690K bytes)
 pkts bytes target     prot opt in     out     source               destination
 5075  589K ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
 191K   90M ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
1499K  133M ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
4364K 6351M ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
 6256  327K ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 3382K packets, 1819M bytes)
 pkts bytes target     prot opt in     out     source               destination
 5075  589K ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0
```

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

iptables固化操作

```
iptables的配置文件 /etc/sysconfig/iptables
iptables-save > /etc/sysconfig/iptables #iptables-save是将规则追加到一个文件，主要是配合iptables-restore命令
```

测试操作：

```
iptables -I INPUT -p tcp --dport 1000 -j ACCEPT
iptables-save > /etc/sysconfig/iptables
iptables -F
iptables-restore < /etc/sysconfig/iptables
```

再查看 /etc/sysconfig/iptables，新增刚刚添加的那条规则

```
echo "iptables-restore < /etc/sysconfig/iptables" >> /etc/profile
```

做reboot操作，iptables 规则仍然存在
不做 iptables-save操作，/etc/sysconfig/iptables文件中是没有你刚刚进行的iptales规则添加的，所以重启时/etc/init.d/iptables restart , 配置文件中没有相应的规则，重启后，规则消失

补充：

RHEL 7 和 CentOS 7 管理 iptables 服务的方法

```
[root@centos7 ~]# yum install -y iptables-services
```

已加载插件：fastestmirror
Loading mirror speeds from cached hostfile
正在解决依赖关系
--> 正在检查事务
---> 软件包 iptables-services.x86_64.0.1.4.21-28.el7 将被 安装
--> 解决依赖关系完成

依赖关系解决

==========

Package 架构 版本 源 大小

正在安装:
 iptables-services x86_64 1.4.21-28.el7 CentOS7 52 k

事务概要

安装 1 软件包

总下载量：52 k
安装大小：26 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  正在安装   : iptables-services-1.4.21-28.el7.x86_64   1/1
警告：/etc/sysconfig/iptables 已建立为 /etc/sysconfig/iptables.rpmnew
  验证中   : iptables-services-1.4.21-28.el7.x86_64   1/1

已安装:
  iptables-services.x86_64 0:1.4.21-28.el7

完毕!

```
[root@centos7 ~]# systemctl start iptables

[root@centos7 ~]# systemctl status iptables
```

● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; vendor preset: disabled)
   Active: active (exited) since 二 2020-04-21 15:53:36 CST; 9s ago
  Process: 7326 ExecStart=/usr/libexec/iptables/iptables.init start (code=exited, status=0/SUCCESS)
 Main PID: 7326 (code=exited, status=0/SUCCESS)

4月 21 15:53:36 centos7 systemd[1]: Starting IPv4 firewall with iptables...
4月 21 15:53:36 centos7 iptables.init[7326]: iptables: Applying firewall ...]
4月 21 15:53:36 centos7 systemd[1]: Started IPv4 firewall with iptables.
Hint: Some lines were ellipsized, use -l to show in full.

```
[root@centos7 ~]# systemctl stop iptables
[root@centos7 ~]# systemctl status iptables
```

● iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; vendor preset: disabled)
   Active: inactive (dead) since 二 2020-04-21 15:54:16 CST; 1s ago
  Process: 7348 ExecStop=/usr/libexec/iptables/iptables.init stop (code=exited, status=0/SUCCESS)
  Process: 7326 ExecStart=/usr/libexec/iptables/iptables.init start (code=exited, status=0/SUCCESS)
 Main PID: 7326 (code=exited, status=0/SUCCESS)

4月 21 15:53:36 centos7 systemd[1]: Starting IPv4 firewall with iptables...
4月 21 15:53:36 centos7 iptables.init[7326]: iptables: Applying firewall ...]
4月 21 15:53:36 centos7 systemd[1]: Started IPv4 firewall with iptables.
4月 21 15:54:16 centos7 systemd[1]: Stopping IPv4 firewall with iptables...
4月 21 15:54:16 centos7 iptables.init[7348]: iptables: Setting chains to ...]
4月 21 15:54:16 centos7 iptables.init[7348]: iptables: Flushing firewall ...]
4月 21 15:54:16 centos7 iptables.init[7348]: iptables: Unloading modules:...]
4月 21 15:54:16 centos7 systemd[1]: Stopped IPv4 firewall with iptables.
Hint: Some lines were ellipsized, use -l to show in full.

```
[root@centos7 ~]# systemctl enable iptables
```

Created symlink from /etc/systemd/system/basic.target.wants/iptables.service to /usr/lib/systemd/system/iptables.service.

```
[root@centos7 ~]# iptables -L
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  localhost            localhost
ACCEPT     all  --  anywhere             anywhere             state
RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp-data
REJECT     all  --  anywhere             anywhere             reject-with icmp-
port-unreachable
DROP       all  --  192.168.100.0/24     anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere             reject-with icmp-
port-unreachable

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
```

```
[root@centos7 ~]# reboot
```

```
Last login: Mon Apr 11 21:29:33 2022 from 192.168.10.1
```

```
[root@centos7 ~]# iptables -L
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  localhost            localhost
ACCEPT     all  --  anywhere             anywhere             state
RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp-data
REJECT     all  --  anywhere             anywhere             reject-with icmp-
```

port-unreachable
DROP   all -- 192.168.100.0/24   anywhere

Chain FORWARD (policy ACCEPT)
target   prot opt source   destination
REJECT   all -- anywhere   anywhere   reject-with icmp-port-unreachable

Chain OUTPUT (policy ACCEPT)
target   prot opt source   destination
ACCEPT   all -- anywhere   anywhere

```
[root@centos7 ~]#
[root@centos7 ~]# systemctl stop iptables
[root@centos7 ~]# iptables -L
```

Chain INPUT (policy ACCEPT)
target   prot opt source   destination

Chain FORWARD (policy ACCEPT)
target   prot opt source   destination

Chain OUTPUT (policy ACCEPT)
target   prot opt source   destination

```
[root@centos7 ~]# systemctl start iptables
[root@centos7 ~]# iptables -L
```

Chain INPUT (policy ACCEPT)
target   prot opt source   destination
ACCEPT   all -- localhost   localhost
ACCEPT   all -- anywhere   anywhere   state RELATED,ESTABLISHED
ACCEPT   tcp -- anywhere   anywhere   tcp dpt:ssh
ACCEPT   tcp -- anywhere   anywhere   tcp dpt:http
ACCEPT   tcp -- anywhere   anywhere   tcp dpt:ftp
ACCEPT   tcp -- anywhere   anywhere   tcp dpt:ftp-data
REJECT   all -- anywhere   anywhere   reject-with icmp-

```
port-unreachable
DROP       all  --  192.168.100.0/24     anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere             reject-with icmp-
port-unreachable

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
```