前言

iptables与firewalld防火墙管理工具在linux发行版Redhat7系列使用较为广泛。
UFW则是在linux发行版Ubuntu下进行管理防火墙的一款管理工具。

在选用防火墙工具的时候，运维或者是开发人员往往会纠结使用哪个。这里给出建议，使用iptables工具管理就禁用firewalld，使用firewalld工具管理就禁用iptables，二者选其一即可，避免产生混乱。

此篇文章不会在原理上做深究，主要以实用性为主，原理可以阅读相关书籍慢慢品味。

正文

一、Netfilter内核模块

无论是使用`iptables`还是`firewalld`，不妨先了解一下**Netfilter内核模块**
什么是Netfilter：linux操作系统核心层内部的一个数据包处理模块。
Hook point：数据包在Netfilter中的挂载点。（PRE_ROUTING INPUT OUTPUT FORWORD POST_ROUTING）

1、netfilter的体系结构

网络数据包的的统信主要通过以下相关步骤，对应netfilter定义的钩子函数，具体可以参考源码介绍。

- **NF_IP_PRE_ROUTING**：

  网络数据包进入系统，经过简单检测后，数据包转交给改函数进行处理，然后根据系统设置的规则对数据包进行处理，如果数据包不被丢弃则交给路由函数进行处理。**在该函数中可以替换IP包的目的地址，及DNAT。**

- NF_IP_LOCAL_IN：

  所有发送给本机的数据包都要通过该函数进行处理，该函数根据系统设置的系统规则对数据包进行处理，如果数据包不被丢弃则交给本地的应用程序。

- NF_IP_FORWARD：

  所有不是发送给本机的数据包都要通过该函数进行处理，该函数会根据系统设置的规则对数据包进行处理，如数据包不被丢弃则转给

  NF_IP_POST_ROUTING处理。

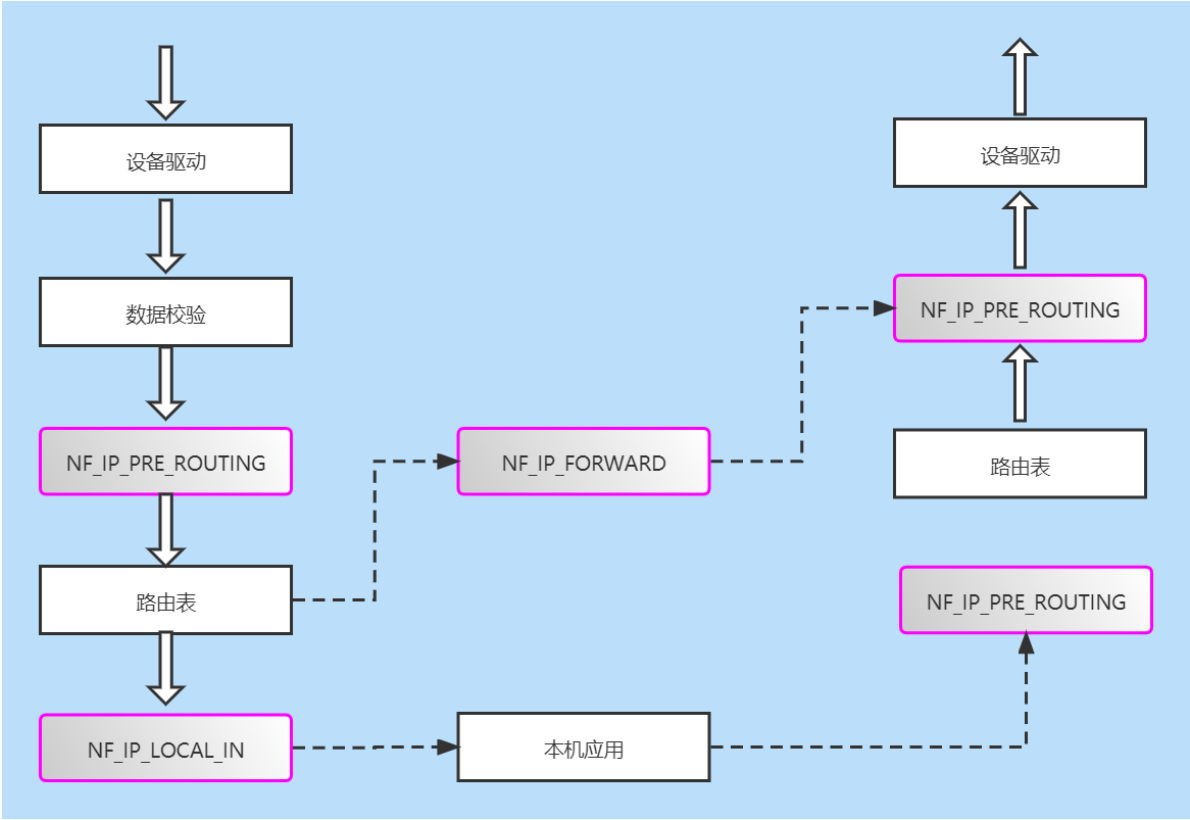
- NF_IP_LOCAL_OUT：

  所有从本地应用程序出来的数据包必须通过该函数进行处理，该函数会根据系统设置的规则对数据包进行处理，如数据包不被丢弃则交给路由函数进行处理。

- **NF_IP_POST_ROUTING**：

  所有数据包在发送给其它主机之前需要通过该函数进行处理，该函数会根据系统设置的规则对数据包进行处理，如数据包不被丢弃，将数据包发给数据链路层。**在该函数中可以替换IP包的源地址，即SNAT。**

数据包通过linux防火墙的处理过程如下图



2、包过滤

每个函数都可以对数据包进行处理，最基本的操作是对数据包进行过滤。系统管理员可以通过iptables工具来向内核模块注册多个过滤规则，并且指明过滤规则的优先权。设置完以后每个钩子按照规则进行匹配。如果规则匹配，函数就会进行一些过滤操作，这些操作主要如下：

- NF_ACCEPT：继续正常的传递包。
- NF_DROP：丢弃包，阻止传送。
- SF_STOLEN：已经接管了包，不需要继续传送。
- NF_QUEUE：排列包。
- NF_REPEAT：再次使用钩子。

篇幅受限，介绍过多反而不好。关于包过滤就介绍这么多，后续会进一步完善。

二、firewalld防火墙工具

以Redhat系列为例子做简单的介绍，熟悉centos的基本上可以套用。

firewalld简介

> 在RHEL7之前的版本中，iptables和ip6tables作为防火墙配置管理工具。在RHEL7中防火墙管理工具变成了firewalld，它是一个支持自定义网络区域（zone）及接口安全等级的动态防火墙管理工具。利用firewalld，用户可以实现许多强大的网络功能，例如防火墙、代理服务器以及网络地址转换。

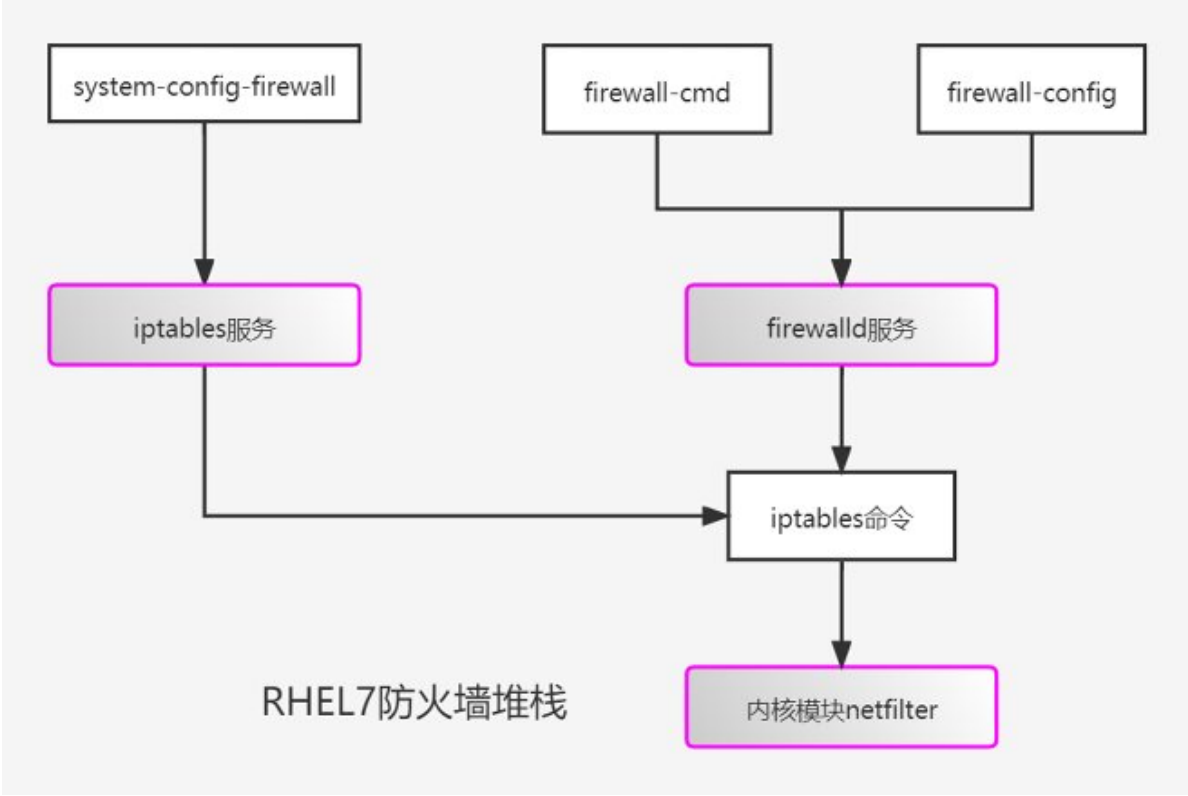
> 之前版本的system-config-firewall和lokkit防火墙模型是静态的，每次修改防火墙规则都需要完全重启。在此过程中包括提供防火墙的内核模块netfilter需要卸载和重新加载。而卸载会破坏已建立的连接和状态防火墙。与之前的静态模型有区别，firewalld将动态地管理防火墙，不需要重新启动防火墙，也不需重新加载内核模块。但firewalld服务要求所有关于防火墙的变更都要通过守护进程来完成，从而确保守护进程中的状态与内核防火墙之间的一致性。

> 许多不了解的人，认为RHEL7中的防火墙从iptables变成了firewalld。其实不然，无论是iptables还是firewalld都无法提供防火墙功能。他们都只是linux系统中的一个防火墙管理工具，负责生成防火墙规则与内核模块netfilter进行“交流”，真正实现防火墙功能的是内核模块netfilter。

> firewalld提供了两种管理模式：其一是firewall-cmd命令管理工具，其二是firewall-config图形化管理工具。在之前版本中的
> iptables将规则保存在文件/etc/sysconfig/iptables中，现在firewalld将配置文件保存在/usr/lib/firewalld和/etc/firewalld
> 目录的xml文件中。

虽然RHEL7中将默认的防火墙管理工具从iptables换成了firewalld，但在RHEL7中仍然可以使用iptables的，只需要通过yum命令进行安装启用iptables服务即可。换句话说，红帽将这个选择权交给了用户。

下面给出RHEL7的防火墙堆栈



1、firewalld命令行模式

1.1、区域选择

当前操作系统安装完成后，防火墙会设置一个默认区域，将接口加入到默认区域中。用户配置防火墙的第一步是获取默认区域并修改，关于操作如下：

查看当前系统中所有区域

```
firewall-cmd --get-zones
```

查看当前默认的区域

```
firewall-cmd --get-default-zone
```

查看当前已激活的区域

```
firewall-cmd --get-active-zones
```

获取接口ens33所属区域

```
firewall-cmd --get-zone-of-interface=ens33
```

修改接口所属区域

```
firewall-cmd --permanent --zone=internal --change-interface=ens33
```

1.2、firewalld服务重载、重启、停止

重新加载防火墙配置

```
firewall-cmd --reload
```

重启防火墙(redhat系列)

```
systemctl restart firewalld.service
```

临时关闭防火墙

```
systemctl stop firewalld.service
```

开机启用防火墙

```
systemctl enable firewalld.service
```

开机禁止防火墙

```
systemctl disable firewalld.service
```

查看firewalld的运行状态

```
firewall-cmd --state
```

1.3、firewalld开放端口（public）

公共区域设置开放21端口永久生效并写入配置文件（参数：--permanent）

```
#参数：--permanent，设置即立刻生效并且写入配置文件
firewall-cmd --zone=public --add-port=21/tcp --permanent
```

查询防火墙端口21是否开放

```
firewall-cmd --zone=public --query-port=21/tcp
```

移除开放的端口21

```
firewall-cmd --zone=public --remove-port=21/tcp --permanent
```

1.4、区域规则修改

查询防火墙规则列表

```
firewall-cmd --zone=public --list-all
```

新增一条区域规则httpd服务

```
firewall-cmd --permanent --zone=internal --add-service=http
```

验证规则

```
firewall-cmd  --zone=internal --list-all
```

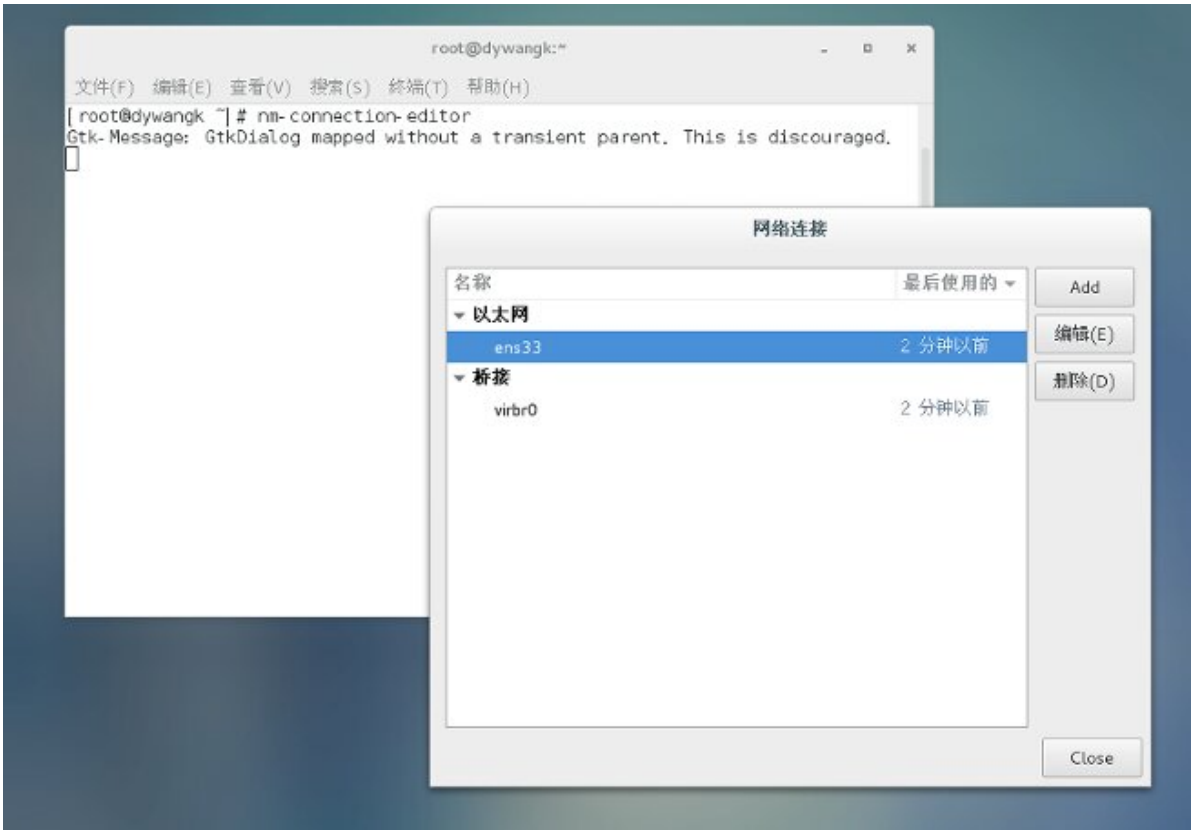
2、firewalld图形化界面

上面的简介也介绍到了firewalld提供了两种管理模式：其一是`firewall-cmd`命令管理工具，其二是`firewall-config`图形化管理工具。在之前版本中的iptables将规则保存在文件`/etc/sysconfig/iptables`中，现在firewalld将配置文件保存在`/usr/lib/firewalld`和`/etc/firewalld`目录的xml文件中。

图形化界面中修改接口区域可以使用NetworkManager，也可以使用firewall-config工具。NetworkManager使用方法：打开终端执行命令`nm-connection-editor`，如下图弹出对话框：

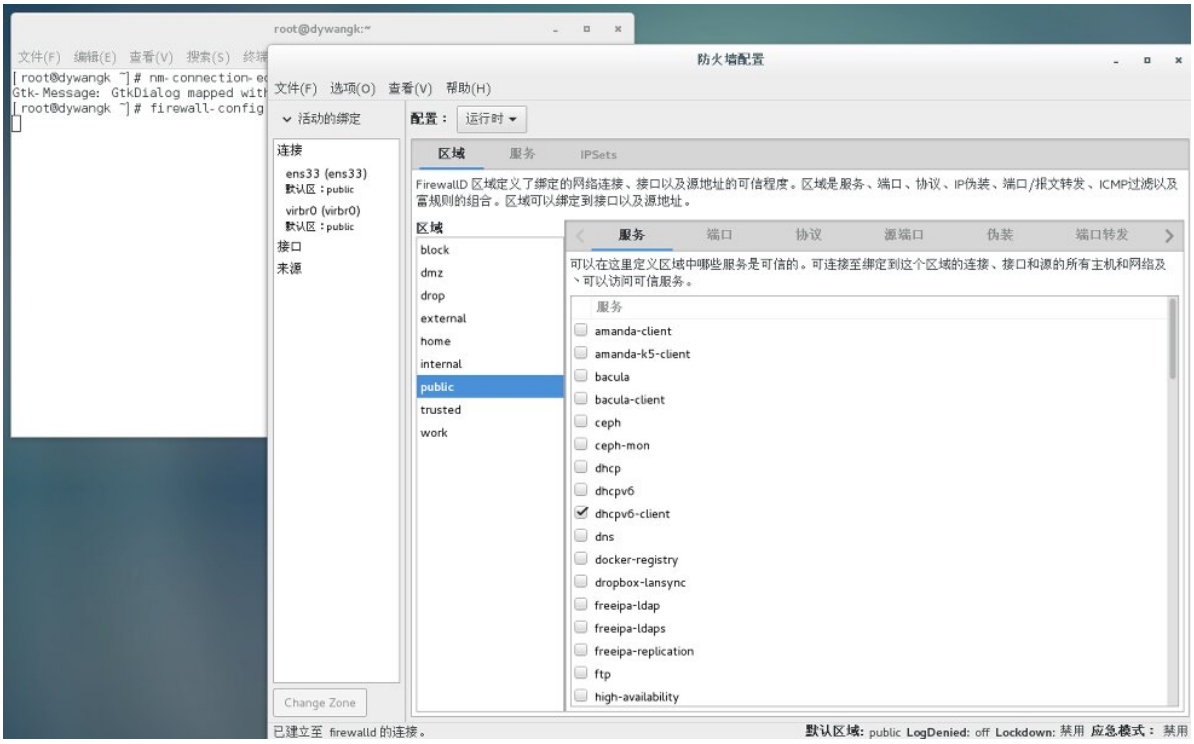
2.1、图形化界面NetworkManager

终端执行：`nm-connection-editor`



2.2、图形化界面firewall-config

终端执行：`firewall-config`



三、iptables防火墙工具

1、安装iptables

假如是centos6，默认是安装了iptables。
如果是centos7或者Redhat7系列，默认没有安装iptables。你需要关闭默认启动的firewalld，二选一即可。

临时关闭firewalld

```
systemctl stop firewalld
```

开机禁用firewalld

```
systemctl disable firewalld
```

开机启用firewalld

```
systemctl enable firewalld
```

通过yum在线安装iptables，检查是否安装了iptables

```
systemctl status iptables.service
service iptables status
```

安装iptables

```
yum -y install iptables
```

升级iptables

```
yum update iptables
```

安装iptables-services

```
yum -y install iptables-services.x86_64
```

设置iptables为开机自启

```
systemctl enable iptables.service
```

iptables规则组成

- ACCEPT(接收，允许通过)
- DROP(丢弃数据包不做任何反馈)
- REJECT(丢弃数据包，客户端有对应消息返回)

查询已经设置的规则：-L命令

```
#一般配合-n命令使用
iptables -L
#不显示主机地址
iptables -nL
```

清除原来设置的规则：-F命令

```
iptables -F
```

删除某一条已经设置的规则：-D命令

```
iptables -D INPUT -p tcp --dport 80 -j ACCEPT
```

2、场景一放通端口

插入一些规则：-I命令，**放通80、22、10~21（一段）这些端口**

```
iptables -I INPUT -p tcp --dport 80 -j ACCEPT
```

设置某一个固定的IP访问80端口：-s 192.168.xxx.xxx

```
iptables -I INPUT -p tcp -s (你的IP地址) --dport 80 -j ACCEPT
```

ssh远程连接本地服务器或者云服务器需要默认启用的端口

```
iptables -I INPUT -p tcp --dport 22 -j ACCEPT
```

设置10~21端口开放访问

```
iptables -I INPUT -p tcp --dport 10:21 -j ACCEPT
```

设置icmp规则允许访问

```
iptables -I INPUT -p icmp -j ACCEPT
```

注意：允许本机可以访问本机，本机访问外网
解决本机可以访问本机（telnet 127.0.0.1 22），添加-i lo（网卡）规则。

设置规则

```
iptables -I INPUT -i lo -p tcp  -j ACCEPT
```

本机测试访问外网

```
curl https://www.baidu.com
```

设置规则

```
iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

在设置的最后追加一条规则：-A命令
拒绝xx规则

```
iptables -A INPUT -j REJECT
```

3、场景二

ftp主动模式下iptables的规则配置（不建议）
ftp被动模式下iptables的规则配置（实际应用，推荐）

4、场景三

工作中的一些常用配置，设置好规则后保存到配置文件。chkconfig iptables on 设置开机启动规则。
snat（对原地址，发起地址）规则设置，dnat（目标地址，发往的地址）规则设置。

5、iptables防攻击企业应用

iptables防攻击企业应用(根据实际业务设置)
利用iptables防CC攻击
connlimit模块
作用： 用于限制每一个客户端IP的并发连接数。
参数： `--connlimit-above n`（次数），限制并发数

例如，限制次数为100

```
iptables -I INPUT -p tcp --syn --dport 80 -m connlimit --connlimit-above 100 -j
REJECT
```

测试，限制某一固定IP并发次数

```
iptables -I INPUT -p tcp --dport 80 -s [ip地址] -m connlimit --connlimit-above 10
-j REJECT
```

6、limit模块

作用： 限速，控制流量
例如

```
iptable -A INPUT -m limit --limit 3/hour
```

`--limit-burst 5`，默认值为5

在设置最后追加一条过滤规则

```
iptables -A INPUT -p icmp -m limit --limit 1/m --limit-burst 10 -j ACCEPT
```

拒绝其它规则访问

```
iptables -A INPUT -p icmp -j DROP
```

设置完，测试接限制的IP地址：

```
#测试受限IP地址
ping 192.168.245.139
```

2、配置文件新增规则

文中介绍过iptables安装后的配置文件所在目录。

```
/etc/sysconfig/iptables
```

```
[root@dywangk ~]# ls /etc/sysconfig/
atd                ip6tables-config   network-scripts                  rsyncd
authconfig         iptables           nfs                              rsyslog
autofs             iptables-config    ntpd                             run-parts
cbq                irqbalance         ntpdate                          samba
cgred              kdump              oracle-database-preinstall-19c   saslauthd
console            kernel             oracledb_ORCLCDB-19c.conf        selinux
cpupower           ksm                pluto                            smartmontools
crond              libvirtd           qemu-ga                          sshd
ebtables-config    libvirt-guests     radvd                            svnserve
fcoe               man-db             raid-check                       sysstat
firewalld          modules            rdisc                            sysstat.ioconf
grub               netconsole         readonly-root                    virtlockd
init               network            rpcbind                          virtlogd
ip6tables          network.orabackup  rpc-rquotad                      wpa_supplicant
```

安装iptables后，生成配置文件

修改配置文件

```
vim  /etc/sysconfig/iptables
```

```
# sample configuration for iptables service
# you can edit this manually or use system-config-firewall
# please do not ask us to add additional ports/services to this default configuration
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
~
~
~
~
~
~
~
~
~
~
~
"/etc/sysconfig/iptables" 14L, 550C                                  1,1           全部
```

四、UFW防火墙工具

> Uncomplicated Firewall
>
> 简称UFW，是Ubuntu系统上默认的防火墙组件。UFW是为轻量化配置iptables而开发的一款工具。
> UFW 提供一个非常友好的界面用于创建基于IPV4，IPV6的防火墙规则。UFW 在 Ubuntu 8.04 LTS 后的所有发行版中默认可用。
> UFW 的图形用户界面叫Gufw。

1、开启与关闭防火墙

开启防火墙

```
ufw enable
```

关闭防火墙

```
ufw disable
```

2、显示防火墙状态

2.1、显示防火墙状态

```
ufw status
```

2.2、查看防火墙详细状态

```
ufw status verbose
```

3、允许与阻止

3.1、增加一条表示允许的规则

```
ufw allow
```

3.2、允许通过 21 连接端口使用 tcp 和 udp 协议连线本机

```
ufw allow 21
```

3.3、增加一条表示阻止的规则

```
ufw deny
```

阻止通过 21 连接端口使用 tcp 协议连线本机

```
ufw deny 21/tcp
```

3.4、增加一条表示拒绝的规则

```
ufw reject
```

4、以服务名称代表连接端口

可以采用

```
less /etc/services
```