Redhat Enterprise Linux7已经默认使用firewalld作为防火墙，其使用方式已经变化。基于iptables的防火墙被默认不启动，但仍然可以继续使用。

RHEL7中有几种防火墙共存：firewalld、iptables、ebtables等，默认使用firewalld作为防火墙，管理工具是firewall-cmd。RHEL7的内核版本是3.10，在此版本的内核里防火墙的包过滤机制是firewalld，使用firewalld来管理netfilter,不过底层调用的命令仍然是iptables等。因为这几种daemon是冲突的，所以建议禁用其他几种服务

[root@sunday-test ~]# systemctl status {firewalld,iptables,ip6tables,ebtables}

```
[root@srv1 ~]# systemctl status {firewalld,iptables,ip6tables,ebtables}
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled)
   Active: active (running) since Wed 2015-04-29 22:16:02 CST; 3min 14s ago
 Main PID: 1057 (firewalld)
   CGroup: /system.slice/firewalld.service
           ¥ì1057 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

Apr 29 22:16:02 srv1.benet.com systemd[1]: Started firewalld - dynamic firewall daemon.

iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled)
   Active: inactive (dead)


ip6tables.service - IPv6 firewall with ip6tables
   Loaded: loaded (/usr/lib/systemd/system/ip6tables.service; disabled)
   Active: inactive (dead)


ebtables.service - Ethernet Bridge Filtering tables
   Loaded: loaded (/usr/lib/systemd/system/ebtables.service; disabled)
   Active: inactive (dead)
```

例如若要禁用iptables、ip6tables、ebtables防火墙，方法如下图

[root@sunday-test ~]# systemctl mask {iptables,ip6tables,ebtables}

```
[root@sunday-test ~]# systemctl mask {iptables,ip6tables,ebtables}
Created symlink from /etc/systemd/system/iptables.service to /dev/null.
Created symlink from /etc/systemd/system/ip6tables.service to /dev/null.
Created symlink from /etc/systemd/system/ebtables.service to /dev/null.
[root@sunday-test ~]#
```

或

```
[root@server1 ~]# for service in iptables ip6tables ebtables;do
> systemctl mask ${service}.service
> done
```

查看这几种服务是否正在运行

```
[root@srv1 ~]# systemctl status {firewalld,iptables,ip6tables,ebtables}
firewalld.service
   Loaded: masked (/dev/null)
   Active: active (running) since Wed 2015-04-29 22:16:02 CST; 6min ago
 Main PID: 1057 (firewalld)
   CGroup: /system.slice/firewalld.service
           ¥1057 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

Apr 29 22:16:02 srv1.benet.com systemd[1]: Started firewalld - dynamic firewall daemon.

iptables.service
   Loaded: masked (/dev/null)
   Active: inactive (dead)


ip6tables.service
   Loaded: masked (/dev/null)
   Active: inactive (dead)


ebtables.service
   Loaded: masked (/dev/null)
   Active: inactive (dead)
```

或

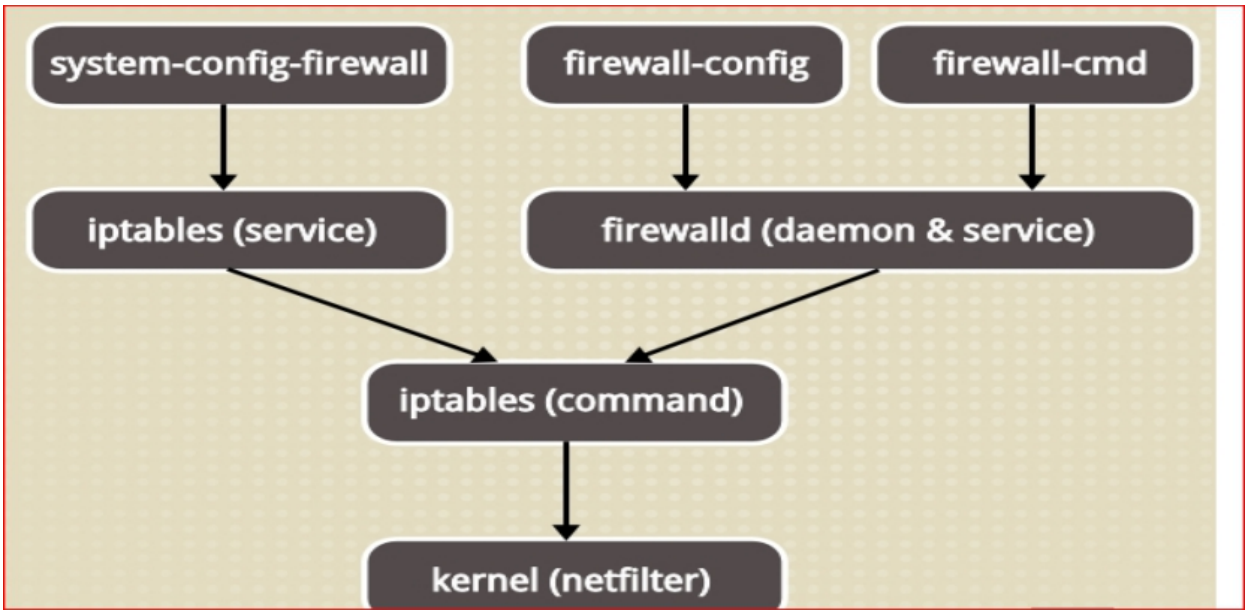
```
[root@server1 ~]# systemctl is-active firewalld.service
active
[root@server1 ~]# systemctl is-active iptables.service
inactive
[root@server1 ~]# systemctl is-active ip6tables.service
inactive
[root@server1 ~]# systemctl is-active ebtables.service
inactive
```

RHEL7虽然有iptables但是不建议使用了，使用新的firewalld服务。

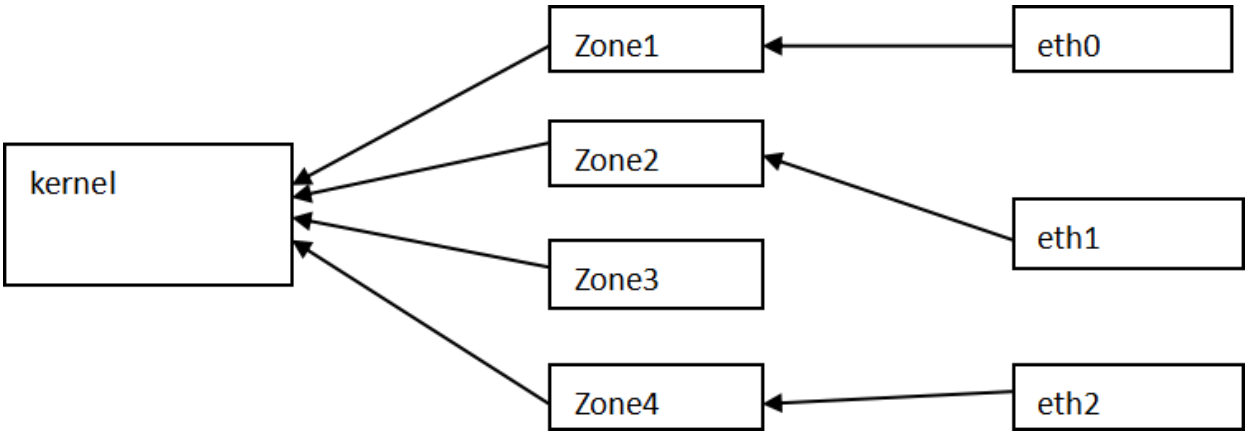
查看firewalld软件包是否安装

```
[root@server1 ~]# rpm -q firewalld
firewalld-0.3.9-7.el7.noarch
```

Firewalld提供了支持网络/防火墙区域(zone)定义网络链接以及接口安全等级的防火墙管理工具。拥有运行时配置和永久配置选项。它也支持允许服务或者应用程序直接添加防火墙规则的接口。以前的 system-config-firewall防火墙模型是静态的，每次修改都要求防火墙完全重启。这个过程包括内核 netfilter 防火墙模块的卸载和新配置所需模块的装载等。相反，firewall daemon 动态管理防火墙，不需要重启整个防火墙便可应用更改。因而也就没有必要重载所有内核防火墙模块了。



什么是**区域**：网络区域定义了网络连接的可信等级。



数据包要进入到内核必须要通过这些zone中的一个，而不同的zone里定义的规则不一样（即信任度不一样，过滤的强度也不一样）。可以根据网卡所连接的网络的安全性来判断，这张网卡的流量到底使用哪个zone，比如上图来自eth0的流量全部使用zone1的过滤规则，eth1的流量使用zone2。一张网卡同时只能绑定到一个zone

预定义的服务：服务是端口和/或协议入口的组合。

端口和协议：定义了 tcp 或 udp 端口，端口可以是一个端口或者端口范围。

ICMP 阻塞：可以选择 Internet 控制报文协议的报文。这些报文可以是信息请求亦可是对信息请求或错误条件创建的响应。

伪装：私有网络地址可以被映射到公开的IP地址。这是一次正规的地址转换。

端口转发：端口可以映射到另一个端口以及/或者其他主机。

在进行firewalld配置之前，先来了解区域（zones）这个概念。默认情况就有一些有效的区域。由firewalld 提供的区域按照从不信任到信任的顺序排序。

•丢弃区域（Drop Zone）：如果使用丢弃区域，任何进入的数据包将被丢弃。这个类似与我们之前使用iptables -j drop。使用丢弃规则意味着将不存在响应。

•阻塞区域（Block Zone）：阻塞区域会拒绝进入的网络连接，返回icmp-host-prohibited，只有服务器已经建立的连接会被通过即只允许由该系统初始化的网络连接。

•公共区域（Public Zone）：只接受那些被选中的连接，默认只允许 ssh 和 dhcpv6-client。这个 zone 是缺省 zone

•外部区域（External Zone）：这个区域相当于路由器的启用伪装（masquerading）选项。只有指定的连接会被接受，即ssh，而其它的连接将被丢弃或者不被接受。

•隔离区域（DMZ Zone）：如果想要只允许给部分服务能被外部访问，可以在DMZ区域中定义。它也拥有只通过被选中连接的特性，即ssh。

•工作区域（Work Zone）：在这个区域，我们只能定义内部网络。比如私有网络通信才被允许，只允许ssh，ipp-client和 dhcpv6-client。

•家庭区域（Home Zone）：这个区域专门用于家庭环境。它同样只允许被选中的连接，即ssh，ipp-client，mdns，samba-client和 dhcpv6-client。

•内部区域（Internal Zone）：这个区域和工作区域（Work Zone）类似，只有通过被选中的连接，和home区域一样。

•信任区域（Trusted Zone）：信任区域允许所有网络通信通过。记住：因为trusted是最被信任的，即使没有设置任何的服务，那么也是被允许的，因为trusted是允许所有连接的

以上是系统定义的所有的 zone，但是这些 zone 并不是都在使用。只有活跃的 zone 才有实际操作意义。

Firewalld的原则：

如果一个客户端访问服务器，服务器根据以下原则决定使用哪个 zone 的策略去匹配

1.如果一个客户端数据包的源IP地址匹配zone的sources，那么该zone的规则就适用这个客户端；一个源只能属于一个zone，不能同时属于多个zone。

2.如果一个客户端数据包进入服务器的某一个接口（如eth0）区配zone的interfaces，则该zone 的规则就适用这个客户端；一个接口只能属于一个zone，不能同时属于多个zone。

3.如果上述两个原则都不满足，那么缺省的zone将被应用

你可以使用任何一种 firewalld 配置工具来配置或者增加区域，以及修改配置。工具有例如 firewall-config 这样的图形界面工具， firewall-cmd 这样的命令行工具，或者你也可以在

配置文件目录中创建或者拷贝区域文件，/usr/lib/firewalld/zones 被用于默认和备用配置，/etc/firewalld/zones 被用于用户创建和自定义配置文件。

使用图形化 firewall-config工具和通过编辑/etc/firewalld/services/中的XML文件，服务可以被增加和删除。如果服务没有被用户增加或者改变，那么/etc/firewalld/services/中不会发现相应的XML文件。如果您希望增加或者改变服务，/usr/lib/firewalld/services/文件可以作为模板使用。以 root 身份执行以下命令：

~]# cp /usr/lib/firewalld/services/[service].xml /etc/firewalld/services/[service].xml

然后您可以编辑最近创建的文件。firewalld优先使用/etc/firewalld/services/里的文件，如果一份文件被删除且服务被重新加载后，会切换到 /usr/lib/firewalld/services/。

管理firewalld

可以通过以下三种方式来管理firewalld:

- Ø 使用命令行工具firewall-cmd
- Ø 使用图形工具firewall-config
- Ø 使用/etc/firewalld/中的配置文件

在大部分情况下，不建议直接编辑配置文件，但是在使用配置管理工具时，以这种方法复制配置会很有用。

在 CentOS7 中，默认安装firewalld 和图形化用户接口配置工具firewall-config。作为 root 用户运行下列命令可以检查：

~]# yum install firewalld firewall-config

要禁用 firewalld，则作为 root 用户运行下列命令：

~]# systemctl disable firewalld.service

~]# systemctl stop firewalld.service

要用iptables和ip6tables服务代替firewalld则以 root 身份运行以下命令，先禁用firewalld,然后安装 iptables-services程序包,以root身份输入以下命令:

~]# yum install iptables-services

iptables-services 程序包包含了iptables服务和ip6tables服务。然后以 root身份运行 iptables 和 ip6tables 命令：

~]# systemctl start iptables

~]# systemctl start ip6tables

~]# systemctl enable iptables

~]# systemctl enable ip6tables

要启动 firewalld，则以root用户身份输入以下命令：

~]# systemctl start firewalld.service

开机启动firewalld，则以root用户身份输入以下命令：

~]# systemctl enable firewalld.service

如果firewalld在运行，输入以下命令检查：

[root@localhost ~]# systemctl status firewalld.service

● firewalld.service - firewalld - dynamic firewall daemon

Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)

Active: active (running) since 一 2017-05-15 21:55:26 CST; 12min ago

Docs: man:firewalld(1)

Main PID: 2565 (firewalld)

CGroup: /system.slice/firewalld.service

└─2565 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

另外，还可以通过firewall-cmd命令来连接后台程序进行检查：

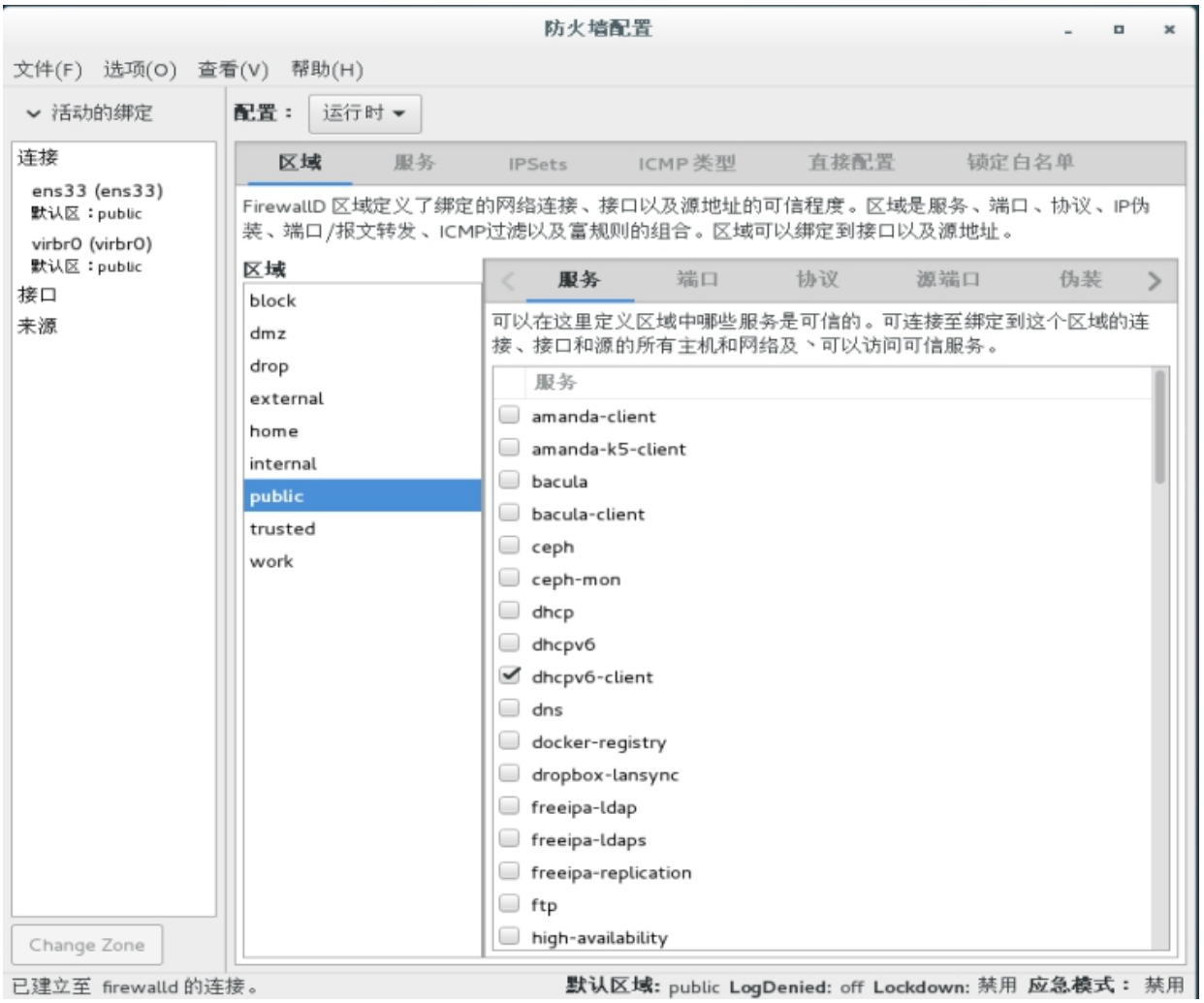
[root@localhost ~]# firewall-cmd --state

running

使用图形化用户接口配置防火墙

点击“应用程序”中的“杂项”，选择“防火墙”，打开firewall-config工具

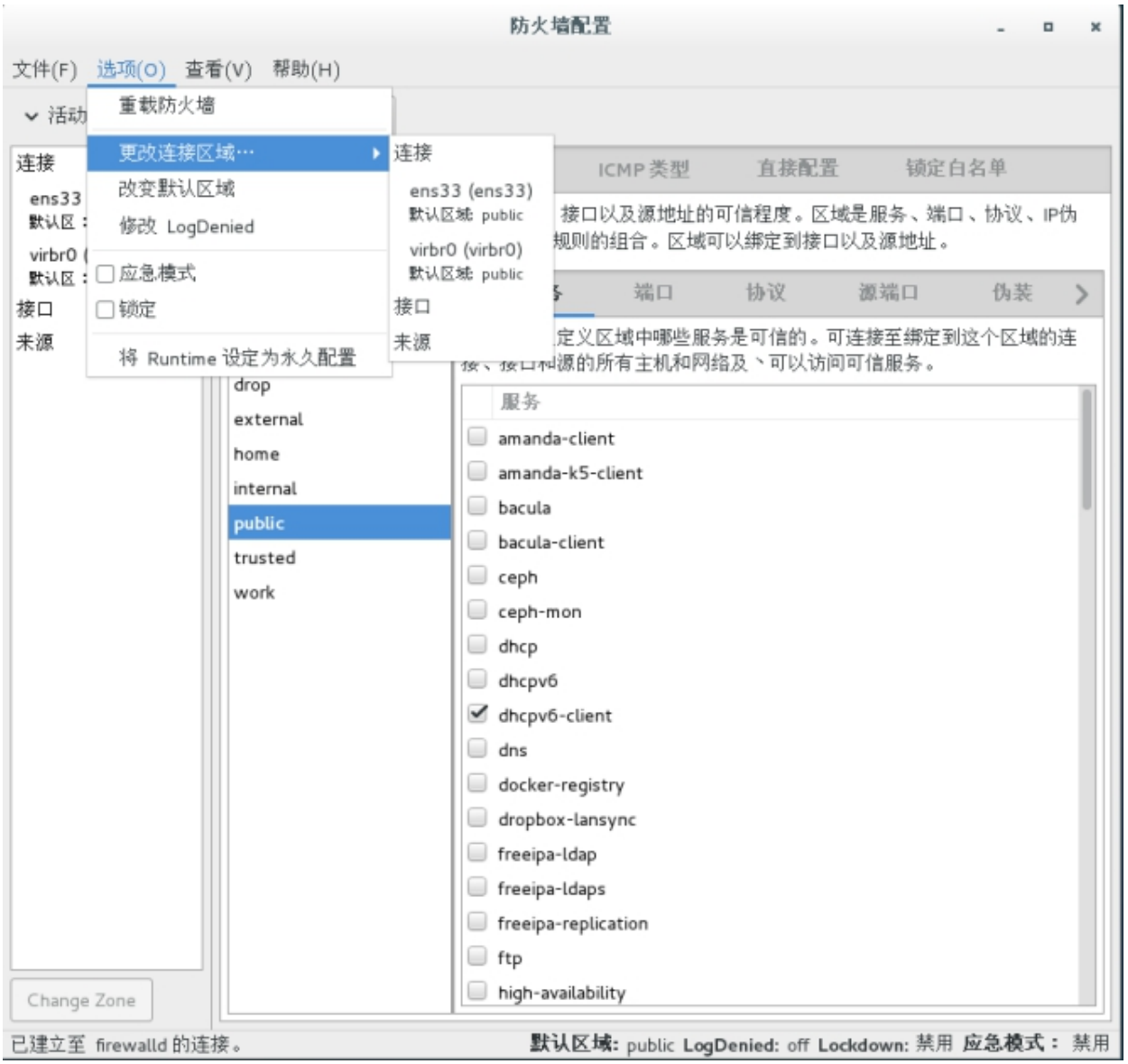
或者直接以root身份在终端中输入：firewall-config命令也可以打开。注意，这个命令可以由普通用户运行，但随后您会被反复提示输入管理员密码。



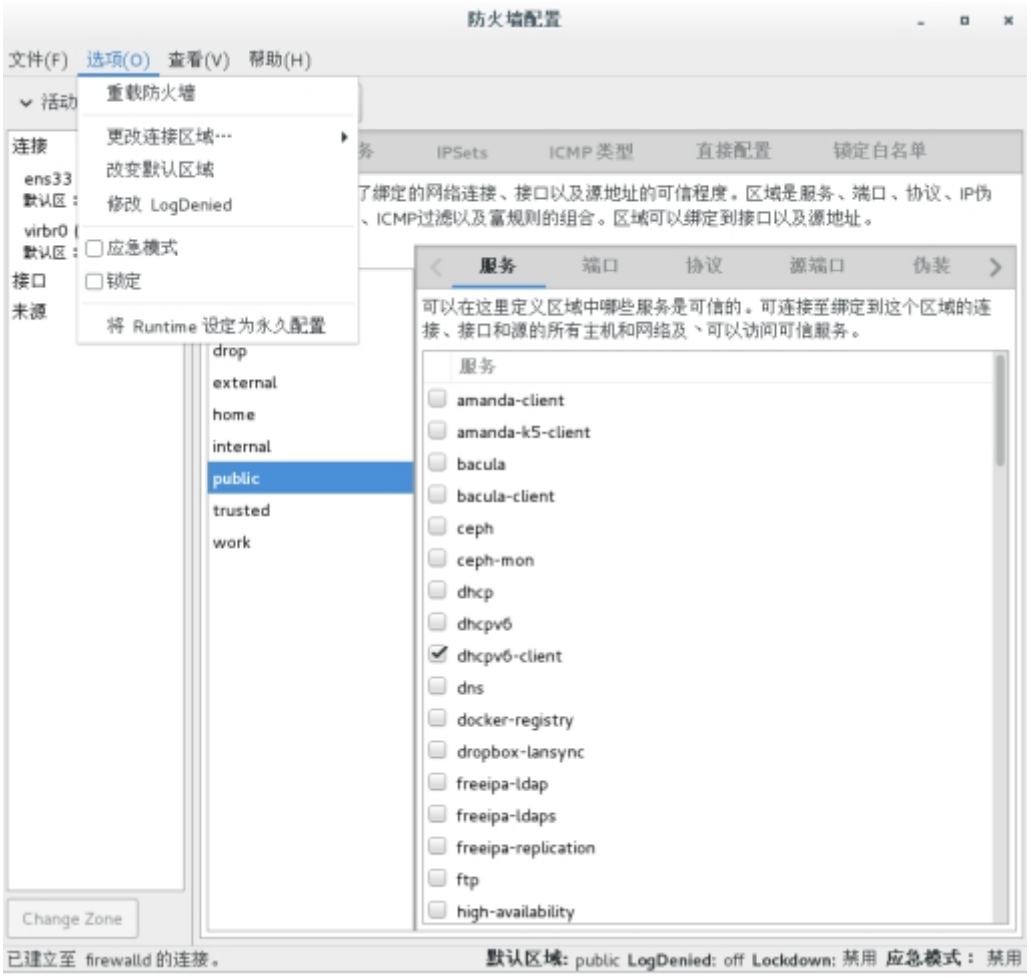
在左下方角落可以看到“已建立至firewalld的连接”，这标志着 firewall-config 工具已经连接到用户区后台程序firewalld。注意，ICMP类型、直接配置和锁定白名单标签只在从“查看”下拉菜单中选择之后才能看见。

要立刻改变现在的防火墙设置，须确定当前视图设定在”运行时”。或者，从下拉菜单中选择“永久”，编辑下次启动系统或者防火墙重新加载时执行的设定。您可以选择左边列里的分区。您将注意到这些分区包含一些可用的服务，您可能需要调整或者滚动窗口才能看见整个列表。您可以通过选择和取消选择一个服务来自定义设定。

要增加或者重新分配一个连接到区域的接口，从菜单栏选择“选项”，由下拉菜单里选择“更改连接区域”，“连接”列表就出现了。选择被分配的连接，出现“为连接选择区域”窗口。从下拉菜单中选择新的防火墙区域并点击确定。



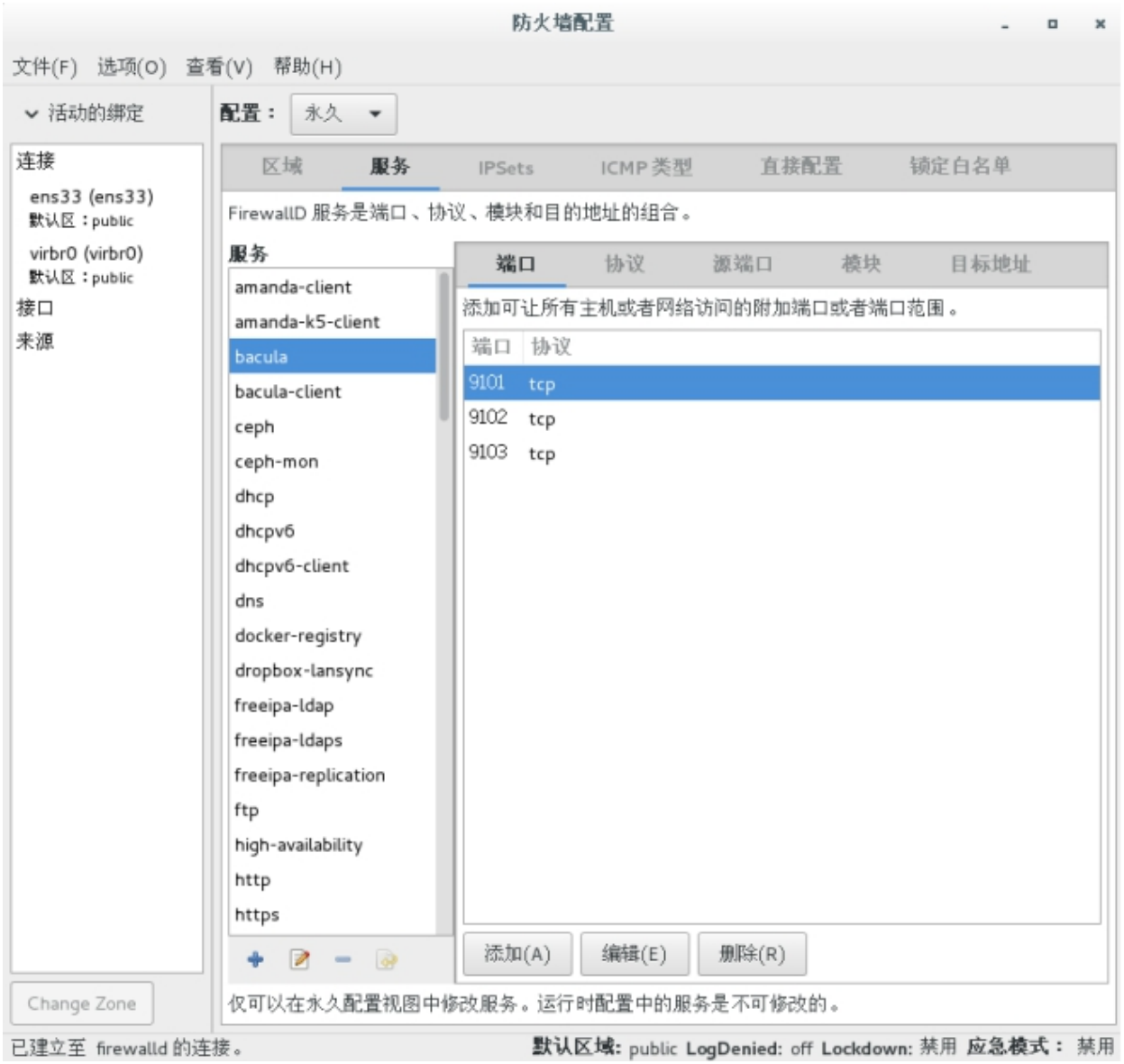
要设定一个将要被分配新接口的区域作为默认值，从菜单栏选择“选项”，由下拉菜单中选择“改变默认区域”，出现“默认区域”窗口。从给出的列表中选择您需要用的“区域”作为默认区域，点击确定。



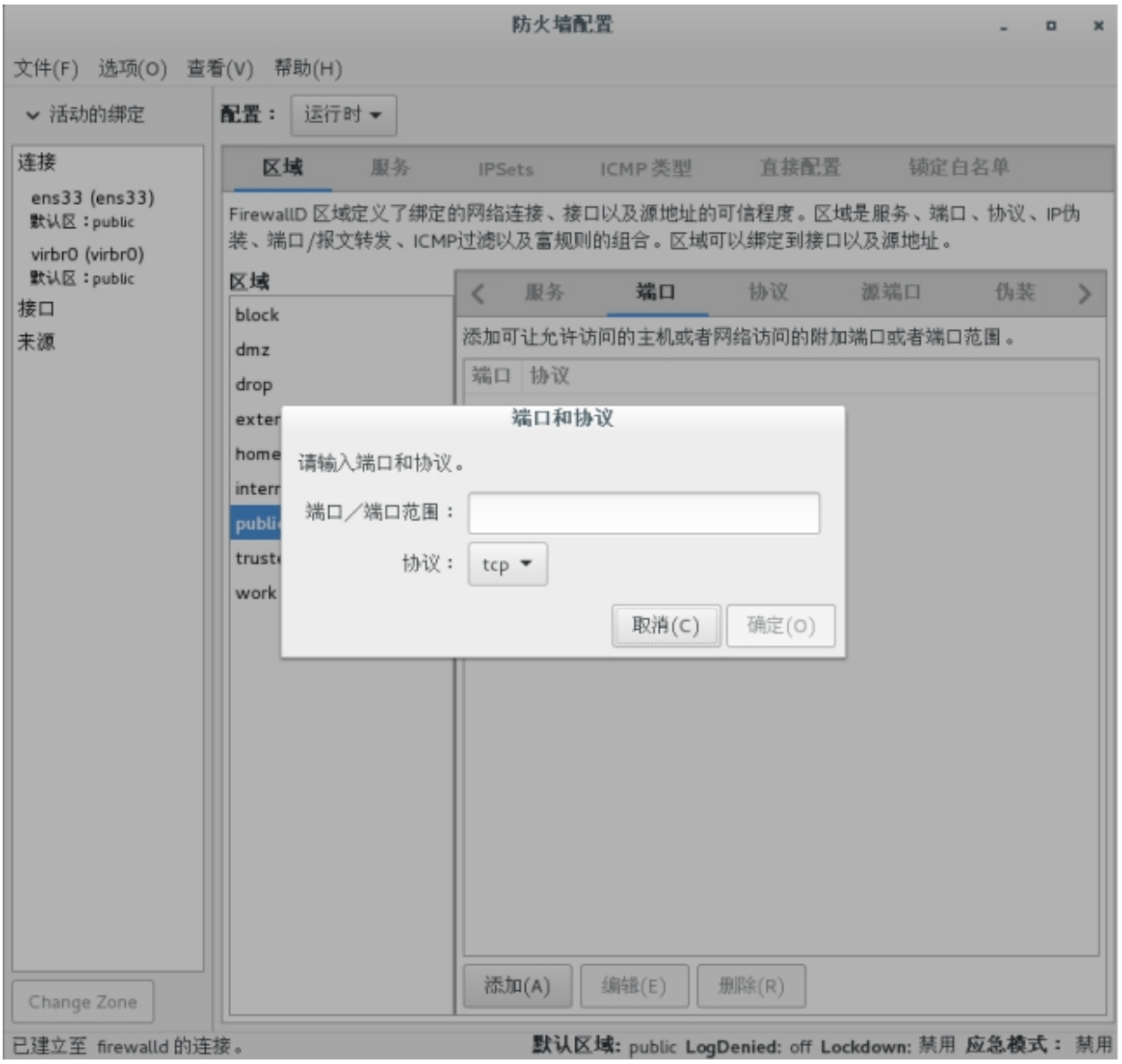
要使用或者禁用一个预先定义的或用户服务，则启动 firewall-config工具并选择将要配置服务的网络区域。选中“服务”标签并选择每个您需要信任的服务类型的复选框。清除复选框则限制服务。

要编辑一项服务，开始 firewall-config 工具，然后从标记为“配置”的下拉选项菜单选择“永久”模式。其余的图标和菜单案件会出现在“服务”窗口的底部。选择您想要配置的服务。

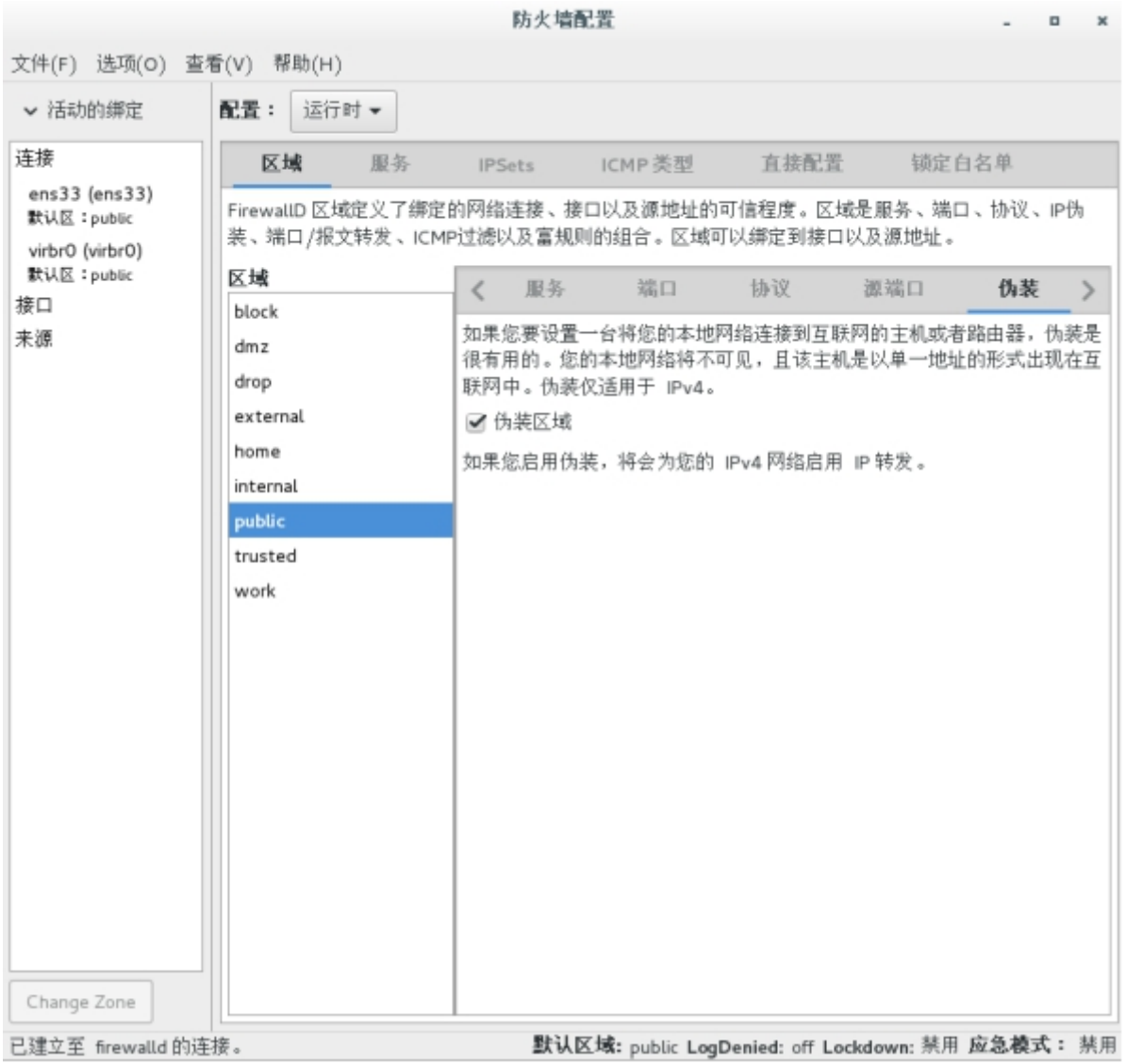
“端口”和“协议”标签可以为选择的服务执行增加、更改、移除端口和协议。“模块”标签用于配置 Netfilter 辅助模块。“目标地址”模块使得流量进入一个受限的特定的目的地址和互联网协议 (IPv4 or IPv6)。



要允许流量通过防火墙到达某个端口，则启动 firewall-config 并选择您想更改设定的网络区域。选择“端口”图标并点击右边的“添加”按钮，“端口和协议”就打开了。输入端口数量或者端口号范围，获得许可。从下拉菜单中选择 tcp 或者 udp。



要将 IPv4 地址转换为一个单一的外部地址，则启动 firewall-config 工具并选择需要转换地址的网络区域。选择“伪装”标签和复选框以便把 IPv4 地址转换成一个单一的地址。

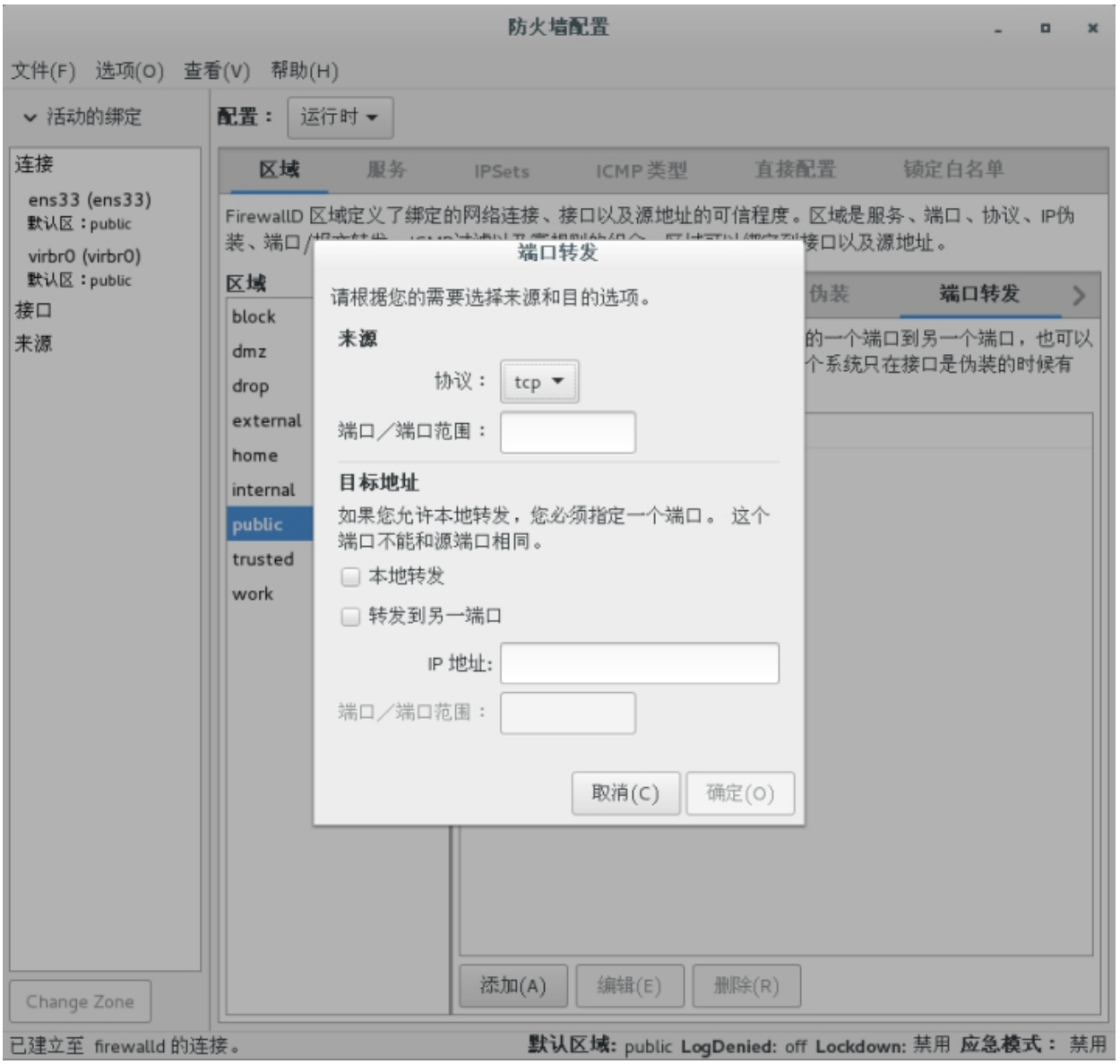


为一个特定端口转发入站网络流量或报文到一个内部地址或者替代端口，首先激活伪装IP地址，然后选择 Port Forwarding 标签。

在窗口靠上部分选择入站流量协议和端口或者端口范围。靠下部分是用于设置目的端口细节的。

要转发流量到一个本地端口即同一系统上的端口，需选择“本地转发”复选框，输入要转发的流量的本地端口或者端口值范围。

要转发流量到其他的IPv4地址，则选择“转发到另一端口”复选框，输入目的地IP地址和端口或者端口范围。如果端口位置空缺则默认发送到同一个端口。点击确定执行更改。



命令行工具firewall-cmd支持全部防火墙特性，基本应用如下：

一般应用：

1、 获取firewalld状态

[root@sunday-test ~]# firewall-cmd --state

```
[root@server1 ~]# firewall-cmd --state
running
```

2、在不改变状态的条件下重新加载防火墙：

[root@sunday-test ~]# firewall-cmd --reload

```
[root@server1 ~]# firewall-cmd --reload
success
```

如果你使用--complete-reload，状态信息将会丢失。

3、获取支持的区域列表

[root@sunday-test ~]# firewall-cmd --get-zones

```
[root@server1 ~]# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

这条命令输出用空格分隔的列表

4、获取所有支持的服务

[root@sunday-test ~]# firewall-cmd --get-services

```
[root@server1 ~]# firewall-cmd --get-services
amanda-client bacula bacula-client dhcp dhcpv6 dhcpv6-client dns ftp high-availability http https im
aps ipp ipp-client ipsec kerberos kpasswd ldap ldaps libvirt libvirt-tls mdns mountd ms-wbt mysql nf
s ntp openvpn pmcd pmproxy pmwebapi pmwebapis pop3s postgresql proxy-dhcp radius rpc-bind samba samb
a-client smtp ssh telnet tftp tftp-client transmission-client vnc-server wbem-https
```

这条命令输出用空格分隔的列表。

服务是firewalld所使用的有关端口和选项的规则集合。被启动的服务会在firewalld服务开启或者运行时自动加载。默认情况下，很多服务是有效的。使用下面命令可列出有效的服务。

想要列出默认有效的服务，也可以进入下面的目录也能够取得。

cd /usr/lib/firewalld/services/

```
[root@localhost ~]# cd /usr/lib/firewalld/services/
[root@localhost services]# ls
amanda-client.xml      http.xml          libvirt.xml  pmwebapis.xml     ssh.xml
bacula-client.xml      imaps.xml         mdns.xml     pmwebapi.xml      telnet.xml
bacula.xml             ipp-client.xml    mountd.xml   pop3s.xml         tftp-client.xml
dhcpv6-client.xml      ipp.xml           ms-wbt.xml   postgresql.xml    tftp.xml
dhcpv6.xml             ipsec.xml         mysql.xml    proxy-dhcp.xml    transmission-client.xml
dhcp.xml               kerberos.xml      nfs.xml      radius.xml        vnc-server.xml
dns.xml                kpasswd.xml       ntp.xml      rpc-bind.xml      wbem-https.xml
ftp.xml                ldaps.xml         openvpn.xml  samba-client.xml
high-availability.xml  ldap.xml          pmcd.xml     samba.xml
https.xml              libvirt-tls.xml   pmproxy.xml  smtp.xml
[root@localhost services]# cat ssh.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>Secure Shell (SSH) is a protocol for logging into and executing commands on remote ma
chines. It provides secure encrypted communications. If you plan on accessing your machine remotely
via SSH over a firewalled interface, enable this option. You need the openssh-server package install
ed for this option to be useful.</description>
  <port protocol="tcp" port="22"/>
</service>
```

想要创建自己的服务，需要在下面的目录下定义它。比如，现在我想添加一个rhmp服务，端口号1935。首先，任选一个服务复制过来。

```
[root@localhost ~]# cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/
[root@localhost ~]# cd /etc/firewalld/services/
[root@localhost services]# ls -l
total 4
-rw-r-----. 1 root root 463 Apr 23 23:04 ssh.xml
```

接下来把复制过来的文件重命名为“rtmp.xml”，

接下来打开并编辑文件的头部、描述、协议和端口号，以供RTMP服务使用，如下图所示。

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>rtmp</short>
  <description>to allow rtmp service</description>
  <port protocol="tcp" port="1935"/>
</service>
```

重启firewalld服务或者重新加载设置，以激活这些设置。

firewall-cmd --reload

为确认服务是否已经启动，运行下面的命令获取有效的服务列表。

firewall-cmd --get-services

```
[root@localhost services]# firewall-cmd --get-services
amanda-client bacula bacula-client dhcp dhcpv6 dhcpv6-client dns ftp high-availability http https im
aps ipp ipp-client ipsec kerberos kpasswd ldap ldaps libvirt libvirt-tls mdns mountd ms-wbt mysql nf
s ntp openvpn pmcd pmproxy pmwebapi pmwebapis pop3s postgresql proxy-dhcp radius rpc-bind rtmp samba
 samba-client smtp ssh telnet tftp tftp-client transmission-client vnc-server wbem-https
```

5、获取所有支持的ICMP类型

[root@sunday-test services]# firewall-cmd --get-icmptypes

```
[root@server1 ~]# firewall-cmd --get-icmptypes
destination-unreachable echo-reply echo-request parameter-problem redirect router-advertisement rout
er-solicitation source-quench time-exceeded
```

这条命令输出用空格分隔的列表。

6、列出全部启用的区域的特性（即查询当前防火墙策略）

[root@sunday-test services]# firewall-cmd --list-all-zones

解释：特性可以是定义的防火墙策略，如：服务、端口和协议的组合、端口/数据报转发、伪装、ICMP 拦截或自定义规则等

```
[root@server1 ~]# firewall-cmd --list-all-zones
block
  interfaces:
  sources:
  services:
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

dmz
  interfaces:
  sources:
  services: ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

上面的命令将会列出每种区域如block、dmz、drop、external、home、internal、public、trusted以及work。如果区域还有其它详细规则（rich-rules）、启用的服务或者端口，这些区域信息也会分别被罗列出来

7、输出区域全部启用的特性。如果省略区域，将显示默认区域的信息。

firewall-cmd [--zone=] --list-all

```
[root@server1 ~]# firewall-cmd --list-all
trusted (default, active)
  interfaces: eno33554992
  sources:
  services:
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

输出指定区域启动的特性

[root@sunday-test services]# firewall-cmd --list-all --zone=public

```
[root@server1 ~]# firewall-cmd --zone=public --list-all
public (active)
  interfaces: eno16777736
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

8、查看默认区域

[root@sunday-test services]# firewall-cmd --get-default-zone

```
[root@server1 ~]# firewall-cmd --get-default-zone
trusted
```

public区域是默认区域。

在文件/etc/firewalld/firewalld.conf中定义成DefaultZone=public。

9、设置默认区域

firewall-cmd --set-default-zone=区域名

```
[root@server1 ~]# firewall-cmd --set-default-zone=drop
success
[root@server1 ~]# firewall-cmd --get-default-zone
drop
```

流入默认区域中配置的接口的新访问请求将被置入新的默认区域。当前活动的连接将不受影响。

10、获取活动的区域

[root@sunday-test ~]# firewall-cmd --get-active-zones

```
[root@server1 ~]# firewall-cmd --get-active-zones
drop
  interfaces: eno33554992
public
  interfaces: eno16777736
```

这条命令将用以下格式输出每个区域所含接口：

区域名

interfaces： 接口名

11、根据接口获取区域即需要查看哪个区域和这个接口绑定即查看某个接口是属于哪个zone的：

firewall-cmd --get-zone-of-interface=接口名

```
[root@server1 ~]# firewall-cmd --get-zone-of-interface=eno16777736
public
```

这条命令将输出接口所属的区域名称。

12、将接口（网卡）增加到区域

firewall-cmd [--zone=] --add-interface=接口名

```
[root@server1 ~]# firewall-cmd --add-interface=eno16777736
success
```

如果接口不属于区域，接口将被增加到区域。如果区域被省略了，将使用默认区域。接口在重新加载后将重新应用。

13、修改接口所属区域

firewall-cmd [--zone=] --change-interface=接口名

```
[root@server1 ~]# firewall-cmd --zone=trusted --change-interface=eno16777736
success
```

这个选项与 --add-interface 选项相似，但是当接口已经存在于另一个区域的时候，该接口将被添加到新的区域。

14、从区域中删除一个接口

firewall-cmd [--zone=] --remove-interface=接口名

```
[root@server1 ~]# firewall-cmd --get-active-zones
drop
  interfaces: eno33554992
trusted
  interfaces: eno16777736
[root@server1 ~]# firewall-cmd --zone=drop --remove-interface=eno33554992
success
```

注：如果某个接口不属于任何Zone，那么这个接口的所有数据包使用默认的Zone的规则

15、查询接口是否属于一个区域

firewall-cmd [--zone=] --query-interface=接口名

```
[root@localhost ~]# firewall-cmd --query-interface=eno16777736
yes
```

如果区域被省略了，将使用默认区域

16、列举区域中启用的服务

firewall-cmd [--zone=] --list-services

```
[root@localhost ~]# firewall-cmd --list-services
dhcpv6-client ssh
```

如果区域被省略了，将使用默认区域

查看home区域中启用服务

[root@sunday-test ~]# firewall-cmd --list-services --zone=home

```
[root@localhost ~]# firewall-cmd --list-services --zone=home
dhcpv6-client ipp-client mdns samba-client ssh
```

17、启用应急模式阻断所有网络连接，以防出现紧急状况

[root@sunday-test ~]# firewall-cmd --panic-on

```
[root@localhost ~]# firewall-cmd --panic-on
success
```

18、禁用应急模式

firewall-cmd --panic-off

19、查询应急模式

firewall-cmd --query-panic

其他相关的配置项可以查看firewall-cmd的手册页：#man firewall-cmd

处理运行时区域：

运行时模式下对区域进行的修改不是永久有效的。重新加载或者重启后修改将失效。

1、启用区域中的一种服务即给某个区域开启某个服务

firewall-cmd [--zone=区域] --add-service=服务 [--timeout=秒数]

此操作启用区域中的一种服务。如果未指定区域，将使用默认区域。如果设定了超时时间，服务将只启用特定秒数。

```
[root@localhost ~]# firewall-cmd --zone=trusted --add-service=ipp-client --timeout=60
success
[root@localhost ~]# firewall-cmd --zone=trusted --list-all
trusted (active)
  interfaces: eno33554960
  sources:
  services: ipp-client
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

使区域中的 ipp-client 服务生效60秒:

启用默认区域中的http服务:firewall-cmd --add-service=http

2、禁用区域中的某种服务即关闭某个服务

firewall-cmd [--zone=区域] --remove-service=服务

此举禁用区域中的某种服务。如果未指定区域，将使用默认区域。

例:禁止默认区域中的 http 服务:

```
[root@localhost ~]# firewall-cmd  --remove-service=http
success
[root@localhost ~]# firewall-cmd  --list-all
public (default, active)
  interfaces: eno16777736
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

3、查询区域中是否启用了特定服务

firewall-cmd [--zone=区域] --query-service=服务

```
[root@localhost ~]# firewall-cmd --zone=trusted --query-service=http
yes
```

Yes表示服务启用，no表示服务关掉了。

4、启用区域端口和协议组合

firewall-cmd [--zone=区域] --add-port=portid[-portid]/protocol [--timeout=seconds]

此操作将启用端口和协议的组合。端口可以是一个单独的端口或者是一个端口范围 - 。协议可以是tcp或udp。

```
[root@server1 ~]# firewall-cmd --zone=public --add-port=8080/tcp
success
[root@server1 ~]# firewall-cmd --zone=public --add-port=25-80/tcp
success
[root@server1 ~]# firewall-cmd --zone=public --list-all
public (default, active)
  interfaces: eno16777736 eno33554992
  sources:
  services: dhcpv6-client http ssh
  ports: 25-80/tcp 8080/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

5、禁用端口和协议组合

firewall-cmd [--zone=区域] --remove-port=portid[-portid]/protocol

```
[root@server1 ~]# firewall-cmd --remove-port=8080/tcp
success
```

6、查询区域中是否启用了端口和协议组合

firewall-cmd [--zone=区域] --query-port=portid[-portid]/protocol

```
[root@server1 ~]# firewall-cmd --query-port=25-80/tcp
yes
```

7、启用区域中的 IP 伪装功能

firewall-cmd [--zone=区域] --add-masquerade

此操作启用区域的伪装功能。私有网络的地址将被隐藏并映射到一个公有IP。这是地址转换的一种形式，常用于路由。由于内核的限制，伪装功能仅可用于IPv4。

8、禁用区域中的 IP 伪装

firewall-cmd [--zone=区域] --remove-masquerade

9、查询区域的伪装状态

firewall-cmd [--zone=区域] --query-masquerade

注意：启用伪装功能的主机同时也需要开启转发服务：

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

或

#vi /etc/sysctl.conf 添加如下内容

net.ipv4.ip_forward = 1

保存退出并执行#sysctl -p使修改生效

10、启用区域的 ICMP 阻塞功能

firewall-cmd [--zone=区域] --add-icmp-block=icmp类型

```
[root@server1 ~]# firewall-cmd --add-icmp-block=echo-request
success
```

此操作将启用选中的 Internet 控制报文协议（ICMP）报文进行阻塞。 ICMP 报文可以是请求信息或者创建的应答报文，以及错误应答。

11、禁止区域的 ICMP 阻塞功能

firewall-cmd [--zone=区域] --remove-icmp-block=icmp类型

12、查询区域的 ICMP 阻塞功能

firewall-cmd [--zone=区域] --query-icmp-block=icmp类型

13、在区域中启用端口转发或映射

firewall-cmd [--zone=区域] --add-forward-port=port=portid[-portid]:proto=protocol[:toport=portid[-portid]][:toaddr=address [/mask]]

端口可以映射到另一台主机的同一端口，也可以是同一主机或另一主机的不同端口。端口号可以是一个单独的端口或者是端口范围 - 。协议可以为tcp或udp。目标端口可以是端口号或者是端口范围 - 。目标地址可以是 IPv4 地址。受内核限制，端口转发功能仅可用于 IPv4。

意思是凡是来从external进来的22端口的数据包全部转发到211.106.65.50

firewall-cmd --zone=external --add-forward-port=port=22:proto=tcp:toaddr=211.106.65.50

14、禁止区域的端口转发或者端口映射

firewall-cmd [--zone=] --remove-forward-port=port=portid[-portid]:proto=protocol[:toport=portid[-portid]][:toaddr=address [/mask]]

15、查询区域的端口转发或者端口映射

firewall-cmd [--zone=]--query-forward-port=port=portid[-portid]:proto=protocol[:toport=portid[-portid]][:toaddr=address [/mask]]

处理永久区域：

永久选项不直接影响运行时的状态。这些选项仅在重载或者重启服务时可用。为了使用运行时和永久设置，需要分别设置两者。选项--permanent需要是永久设置的第一个参数。

1、获取永久选项所支持的服务

firewall-cmd --permanent --get-services

2、获取永久选项所支持的ICMP类型列表

firewall-cmd --permanent --get-icmptypes

3、获取支持的永久区域

firewall-cmd --permanent --get-zones

4、配置防火墙在public区域打开http协议，并保存，以致重启有效

firewall-cmd --permanent --zone=public --add-service=http

查看永久模式下public区域是否打开http服务。

firewall-cmd --permanent --zone=public --query-service=http

5、防火墙开放8080端口在public区域

firewall-cmd --permanent --zone=public --add-port=8080/tcp

Firewall中理解直接规则：

firewalld 有一个被称为“direct interface”（直接接口），它允许管理员将手动编码的iptables、ip6tables和ebtables 规则插入到firewalld管理的区域中。它适用于应用程序，而不是用户。如果您不太熟悉 iptables，那么使用直接接口是很危险的，因为您可能无意中导致防火墙被入侵。firewalld 保持对所增加项目的追踪，所以它还能质询 firewalld 和发现由使用直接端口模式的程序造成的更改。直接端口由增加 --direct 选项到 firewall-cmd 命令来使用。除非将直接规则显式插入到由Firewalld管理的区域，否则将首先解析直接规则，然后才会解析任何Firewalld规则。添加一些直接规则以将某个IP范围列入黑名单的简短示范：

[root@web ~]# firewall-cmd --direct --permanent --add-chain ipv4 raw blacklist

[root@web ~]# firewall-cmd --direct --permanent --add-rule ipv4 raw PREROUTING 0 -s 192.168.0.0/24 -j blacklist

[root@web ~]# firewall-cmd --direct --permanent --add-rule ipv4 raw blacklist 0 -m limit --limit 1/min -j LOG --log-prefix "blacklisted"

[root@web ~]# firewall-cmd --direct --permanent --add-rule ipv4 raw blacklist 1 -j DROP

可以来个简单的规则：打开9000端口

firewall-cmd --direct --add-rule ipv4 filter INPUT 0 -p tcp --dport 9000 -j ACCEPT

使用富语言（rich language）

富语言特性提供了一种不需要了解iptables语法的通过高级语言配置复杂IPv4和IPv6防火墙规则的机制，为管理员提供了一种表达性语言，通过这种语言可以表达firewalld的基本语法中未涵盖的自定义防火墙规则；例如：仅允许从单个IP地址（而非通过某个区域路由的所有IP地址）连接到服务。

富规则可用于表达基本的允许/拒绝规则，也可以用于配置记录（面向syslog和auditd）以及端口转发、伪装和速率限制。下面是表达富规则的基本语法：

rule [family="<rule family>"]

[source address="<address>" [invert="True"]]

[destination address="<address>" [invert="True"]]

[<element>]

[log [prefix="<prefix text>"] [level="<log level>"] [limit value="rate/duration"]]

[audit]

[accept|reject|drop]

规则的几乎每个单一元素都能够以option=value的形式来采用附加参数。

规则排序：

一旦向某个区域（一般是指防火墙）中添加了多个规则，规则的排序会在很大程度上影响防火墙的行为。对于所有的区域，区域内的规则的基本排序是相同的。

1、为该区域设置的任何记录规则(log)

2、为该区域设置的任何拒绝规则(deny)

3、为该区域设置的任何允许规则(allow)

如果区域中的任何规则与包均不匹配，那么通常会拒绝该包，但是区域可能具有不同的默认值，例如：可信区域（trusted）将接受任何不匹配的包。此外，在匹配某个记录规则后，将继续正常处理包。

直接规则是个例外。对于大部分直接规则，将首先进行解析，然后再由firewalld进行任何其他处理，但是直接规则语法允许管理员在任何区域中的任何位置插入任何规则。

测试和调试

为了便于测试和调试，几乎所有规则都可以与超时一起添加到运行时配置。当包含超时的规则添加到防护墙，计时器便针对该规则开始倒计时，一旦规则的计时器达到0秒，便从运行时配置中删除该规则。

在使用远程防火墙时，使用超时会是一种极其有用的工具，特别是在测试更复杂的规则集时。如果规则有效，则管理员可以再次添加该规则，如果规则没有按照预期运行，甚至可能将管理员锁定而使其无法进入系统，那么规则将被自动删除，以允许管理员可以继续其工作。

通过在启用规则的firewall-cmd的结尾添加选项--timeout=<TIMEINSECONDS>,即可向运行时规则中添加超时。

理解富规则命令

firewall-cmd有四个选项可以用于处理富规则，所有这些选项都可以同常规的--permanent或--zone=<ZONE>选项组合使用。

选项	说明
--add-rich-rule=’ RULE’	向指定区域中添加RULE，如果没有指定区域，则为默认区域。
--remove-rich-rule=’ RULE’	从指定区域中删除RULE，如果没有指定区域，则为默认区域。
--query-rich-rule=’ RULE’	查询RULE是否已添加到指定区域，如果未指定区域，则为默认区域。规则存在，则返回0，否则返回1。
--list-rich-rules	输出指定区域的所有富规则，如果未指定区域，则为默认区域。

任何已配置的富规则还将显示在firewall-cmd --list-all 和firewall-cmd --list-all-zones的输出中。具体语法解释：

source

规定限制源IP地址，源地址可以是一个IPv4或者IPv6地址或者一个网络地址段。

destination

规定限制目标。目标地址使用跟源地址相同的语法。

element

这个要素只可以是以下要素类型之一： service， port， protocol， masquerade， icmp-block 和 forward-port 。

service

服务名称是 firewalld 提供的其中一种服务。要获得被支持的服务的列表，输入以下命令： firewall-cmd --get-services 。如果一个服务提供了一个目标地址，它将和规则中的目标地址冲突，并且导致一个错误。命令为以下形式：

service name=service_name

port

端口既可以是一个独立端口数字，又或者是端口范围，例如，5060-5062。协议可以指定为 tcp 或 udp 。命令为以下形式：

port port=number_or_range protocol=protocol

protocol

协议值可以是一个协议ID号，或者一个协议名。预知可用协议，请查阅/etc/protocols。命令为以下形式：

protocol value=protocol_name_or_ID

icmp-block

用这个命令阻断一个或多个 ICMP 类型。 ICMP 类型是 firewalld 支持的 ICMP 类型之一。要获得被支持的 ICMP 类型列表，输入以下命令：

~]$ firewall-cmd --get-icmptypes

在此，指定一个动作是不被允许的。icmp-block 在内部使用 reject 动作。命令为以下形式：

icmp-block name=icmptype_name

masquerade

打开规则里的IP伪装。用源地址而不是目的地址来把伪装限制在一个范围内。在此，指定一个动作是不被允许的。

forward-port

从一个带有指定为 tcp 或 udp 协议的本地端口转发数据包到另一个本地端口，或另一台机器，或另一台机器上的另一个端口。 port 和 to-port 可以是一个单独的端口数字，或一个端口范围。而目的地址是一个简单的 IP 地址。在此，指定一个动作是不被允许的。forward-port 命令使用内部动作 accept 。这个命令为以下形式：

forward-port port=number_or_range protocol=protocol /

to-port=number_or_range to-addr=address

log

注册含有内核日志的新的连接请求到规则中，比如系统日志。可以定义一个前缀文本把日志信息作为前缀加入。日志等级可以是 emerg 、 alert 、 crit 、 error 、warning 、 notice 、 info 或者 debug 中的一个。可以选择日志的用法，按以下方式限制日志：

log [prefix=prefix text] [level=log level] limit value=rate/duration

持续时间的单位为 s 、 m 、 h 、 d 。 s 表示秒， m 表示分钟， h 表示小时， d 表示天。最大限定值是 1/d ，意为每天最多有一条日志进入。

audit

审核为发送到 auditd 服务的审核记录来注册提供了另一种方法。审核类型可以是 ACCEPT 、 REJECT 或 DROP 中的一种，但不能在 audit 命令后指定，因为审核类型将会从规则动作中自动收集。审核不包含自身参数，但可以选择性地增加限制。审核的使用是可选择的。

accept|reject|drop

可以是accept、reject 或 drop 中的一个行为。

accept | reject [type=reject type] | drop

选择 accept，所有新的连接请求都会被允许。选择 reject ，连接将被拒绝，连接来源将接到一个拒绝信息。选择 drop，所有数据包会被丢弃，并且不会向来源地发送任何信息。

富规则配置举例：

查看富规则：# firewall-cmd --list-rich-rules

为认证报头协议AH使用新的IPv4 和 IPv6 连接：

[root@gateway-server ~]# firewall-cmd --add-rich-rule='rule protocol value=ah accept'

同意新的IPv4和IPv6连接 FTP，并使用审核每分钟记录一次：

[root@gateway-server ~]# firewall-cmd --add-rich-rule='rule service name=ftp log limit value=1/m audit accept'

为TFTP协议同意来自192.168.0.0/24地址的新的IPv4连接，并且使用系统日志每分钟记录一次：

[root@gateway-server ~]# firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.168.0.0/24" service name="tftp" log prefix="tftp" level="info" limit value="1/m" accept'

为 RADIUS协议拒绝所有来自1:2:3:4:6::的新IPv6连接，并每分钟在级别3登录。接受来自其他来源的新的IPv6连接

[root@gateway-server ~]# firewall-cmd --add-rich-rule='rule family="ipv6" source address="1:2:3:4:6::" service name="radius" log prefix="dns" level="info" limit value="3/m" reject'

[root@gateway-server ~]# firewall-cmd --add-rich-rule='rule family="ipv6" service name="radius" accept'

转发带有TCP协议的端口4011上的来自1:2:3:4:6::的IPv6包，到端口4012上的1::2:3:4:7

[root@gateway-server ~]# firewall-cmd --add-rich-rule='rule family="ipv6" source address="1:2:3:4:6::" forward-port to-addr="1::2:3:4:7" to-port="4012" protocol="tcp" port="4011"'

把一个源地址加入白名单，以便允许来自这个源地址的所有连接

[root@gateway-server ~]# firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.168.2.2" accept'

拒绝来自public区域中IP地址192.168.0.11的所有流量

[root@gateway-server ~]# firewall-cmd --zone=public --add-rich-rule='rule family=ipv4 source address=192.168.0.11/32 reject'

丢弃来自默认区域中任何位置的所有传入的ipsec esp协议包

[root@gateway-server ~]# firewall-cmd --add-rich-rule='rule protocol value="esp" drop'

在192.168.1.0/24子网的dmz区域中，接收端口7900--7905的所有TCP包

[root@gateway-server ~]# firewall-cmd --zone=dmz --add-rich-rule='rule family=ipv4 source address=192.168.1.0/24 port port=7900-7905 protocol=tcp accept'

接收从work区域到SSH的新连接，以notice级别且每分钟最多三条消息的方式将新连接记录到syslog

[root@gateway-server ~]# firewall-cmd --zone=work --add-rich-rule='rule service name=ssh log prefix="ssh" level="notice" limit value="3/m" accept'

在接下来的5分钟内，将拒绝从默认区域中的子网192.168.2.0/24到DNS的新连接，并且拒绝的连接将记录到audit系统，且每小时最多一条消息。

[root@gateway-server ~]# firewall-cmd --add-rich-rule='rule family=ipv4 source address=192.168.2.0/24 service name=dns audit limit value="1/h" reject' --timeout=300

更多的富规则配置例子：

firewall-cmd --add-rich-rule 'rule family=ipv4 source address=10.35.89.0/24 service name=ftp log prefix="ftp" level=info accept' --permanent

firewall-cmd --add-rich-rule 'rule family=ipv4 source address=10.35.89.0/24 port port=80 protocol=tcp log prefix="80" level=info accept' --permanent

firewall-cmd --add-rich-rule rule family="ipv4" source address="192.168.10.30" forward-port port="808" protocol="tcp" to-port="80" to-addr="10.10.10.2"

富规则中使用伪装功能可以更精确详细的限制：

firewall-cmd --add-rich-rule 'rule family=ipv4 source address=10.10.10.2/24 masquerade'

仅允许部分IP访问本机服务配置

firewall-cmd --permanent --zone=public --add-rich-rule="rule family="ipv4"

source address="192.168.0.0/24" service name="http" accept"

禁止远程IP访问ssh

firewall-cmd --permanent --zone=public --add-rich-rule=’ rule family=ipv4 source address=192.168.0.0/24 service name=ssh reject’

删除rich规则

firewall-cmd --permanent --zone=public --remove-rich-rule=’ rule family=ipv4 source address=192.168.0.0/24 service name=ssh reject’

仅允许部分IP访问本机端口配置

firewall-cmd --permanent --zone=public --add-rich-rule="rule family="ipv4" source address="192.168.0.0/24"port protocol="tcp" port="8080" accept"

创建rich规则，可以指定日志的前缀和输出级别

firewall-cmd --permanent --zone=public --add-rich-rule="rule family="ipv4" source address="192.168.0.4/24"port port=8080 protocol="tcp" log prefix=proxy level=warning accept"

可以通过查看/var/log/messages日志文件

端口转发。实验环境下，desktop访问server的5423端口，将访问server的80端口。

Server上的操作：（172.25.0.10是desktop的IP地址）

```
[root@srv1 ~]# firewall-cmd --permanent --add-rich-rule 'rule family=ipv4 source address=172.25.0.10
/32 forward-port port=5432 protocol=tcp to-port=80'
success
```

172.25.1.0/24网段内的客户端不能访问主机的SSH

```
[root@srv1 ~]# firewall-cmd --permanent --add-rich-rule 'rule family="ipv4" source address=192.168.1
0.0/24 service name=ssh drop'
success
[root@srv1 ~]# firewall-cmd --reload
success
```

通过编辑 XML 文件为一个区域增加服务

以root身份输入以下命令，查看默认区域文件：

~]# ls /usr/lib/firewalld/zones/

block.xml dmz.xml drop.xml external.xml home.xml internal.xml public.xml trusted.xml work.xml

这些文件不能编辑。如果/etc/firewalld/zones/目录里没有等效文件存在，它们被默认为可使用。也就是说如果两个目录同时存在相同名称的区域文件，则/etc/firewalld/zones/目录中的区域文件优先级高。以root身份输入以下命令，查看从默认区域被更改的区域文件：

~]# ls /etc/firewalld/zones/

public.xml public.xml.old

在上述示例中，work区域文件不存在。以 root身份输入以下命令，加入work文件：

~]# cp /usr/lib/firewalld/zones/work.xml /etc/firewalld/zones/

现在您可以在/etc/firewalld/zones/目录中编辑该文件。如果您删除该文件，firewalld 将切换到使用 /usr/lib/firewalld/zones/ 里的默认文件。要将一个服务加入work区域，比如允许SMTP 进入work区域，则以root权限编辑 /etc/firewalld/zones/work.xml 文件，使之包括如下行：

<service name="smtp"/>

然后执行以下命令重新加载firewalld以使其生效

firewall-cmd --reload

以下是一个修改了的public区域的文件例子：

#cat /etc/firewalld/zones/public.xml

<?xml version="1.0" encoding="utf-8"?>

<zone>

<short>Public</short>

<description>For use in public areas. You do not trust the other computers on networks

to not harm your computer. Only selected incoming connections are accepted.

</description>

<service name="dhcpv6-client"/>

<service name="ssh"/>

<rule family="ipv4">

<source address="192.168.0.4/24"/>

<service name="http"/>

<accept/>

</rule>

</zone>

总结

netfilter 防火墙总是容易受到规则顺序的影响，因为一条规则在链中没有固定的位置。在一条规则之前添加或者删除规则都会改变此规则的位置。在静态防火墙模型中，改变防火墙就是重建一个干净和完善的防火墙设置，默认链通常也没有安全的方式添加或删除规则而不影响其他规则。

动态防火墙有附加的防火墙功能链。这些特殊的链按照已定义的顺序进行调用，因而向链中添加规则将不会干扰先前调用的拒绝和丢弃规则。从而利于创建更为合理完善的防火墙配置。

下面是一些由守护进程创建的规则，过滤列表中启用了在公共区域对 ssh，mdns 和 ipp-client 的支持：

```
Chain IN_public (4 references)
target     prot opt source               destination
IN_public_log  all  --  anywhere             anywhere
IN_public_deny  all  --  anywhere             anywhere
IN_public_allow  all  --  anywhere             anywhere

Chain IN_public_allow (1 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate NEW mark match 0x64
ACCEPT     tcp  --  172.16.16.21         anywhere             tcp dpt:webcache ctstate NEW
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http ctstate NEW
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh ctstate NEW

Chain IN_public_deny (1 references)
target     prot opt source               destination
REJECT     icmp --  anywhere             anywhere             icmp echo-request reject-with icmp-hos
t-prohibited

Chain IN_public_log (1 references)
target     prot opt source               destination
LOG        tcp  --  172.16.16.21         anywhere             tcp dpt:webcache ctstate NEW LOG level
 warning prefix "proxy"
```

总结：

图形化配置工具

firewall daemon 主要的配置工具是**firewall-config**。它支持防火墙的所有特性。管理员也可以用它来改变系统或用户策略。

命令行客户端

firewall-cmd是命令行下提供大部分图形工具配置特性的工具。

附录：要想了解更多firewall防火墙更多知识可以查看firewall的相关手册页，下图所显示的就是firewall防火墙的相关手册页：

```
[root@srv1 ~]# man -k firewalld
firewall-cmd (1)      - firewalld command line client
firewall-config (1)   - firewalld GUI configuration tool
firewall-offline-cmd (1) - firewalld offline command line client
firewalld (1)         - Dynamic Firewall Manager
firewalld.conf (5)    - firewalld configuration file
firewalld.dbus (5)    - firewalld D-Bus interface description
firewalld.direct (5) - firewalld direct configuration file
firewalld.icmptype (5) - firewalld icmptype configuration files
firewalld.lockdown-whitelist (5) - firewalld lockdown whitelist configuration file
firewalld.richlanguage (5) - Rich Language Documentation
firewalld.service (5) - firewalld service configuration files
firewalld.zone (5)    - firewalld zone configuration files
firewalld.zones (5)   - firewalld zones
[root@srv1 ~]#
```

若要查看rich-rule手册页

```
[root@srv1 ~]# man firewalld.richlanguage
```

```
General rule structure

    rule
      [source]
      [destination]
      service¦port¦protocol¦icmp-block¦masquerade¦forward-port
      [log]
      [audit]
      [accept¦reject¦drop]
```

例如：允许icmp协议的数据包通信

```
[root@server1 ~]# firewall-cmd --add-rich-rule="rule protocol value="icmp" accept"
```