Centos7系列（三）防火墙与网络区域详解

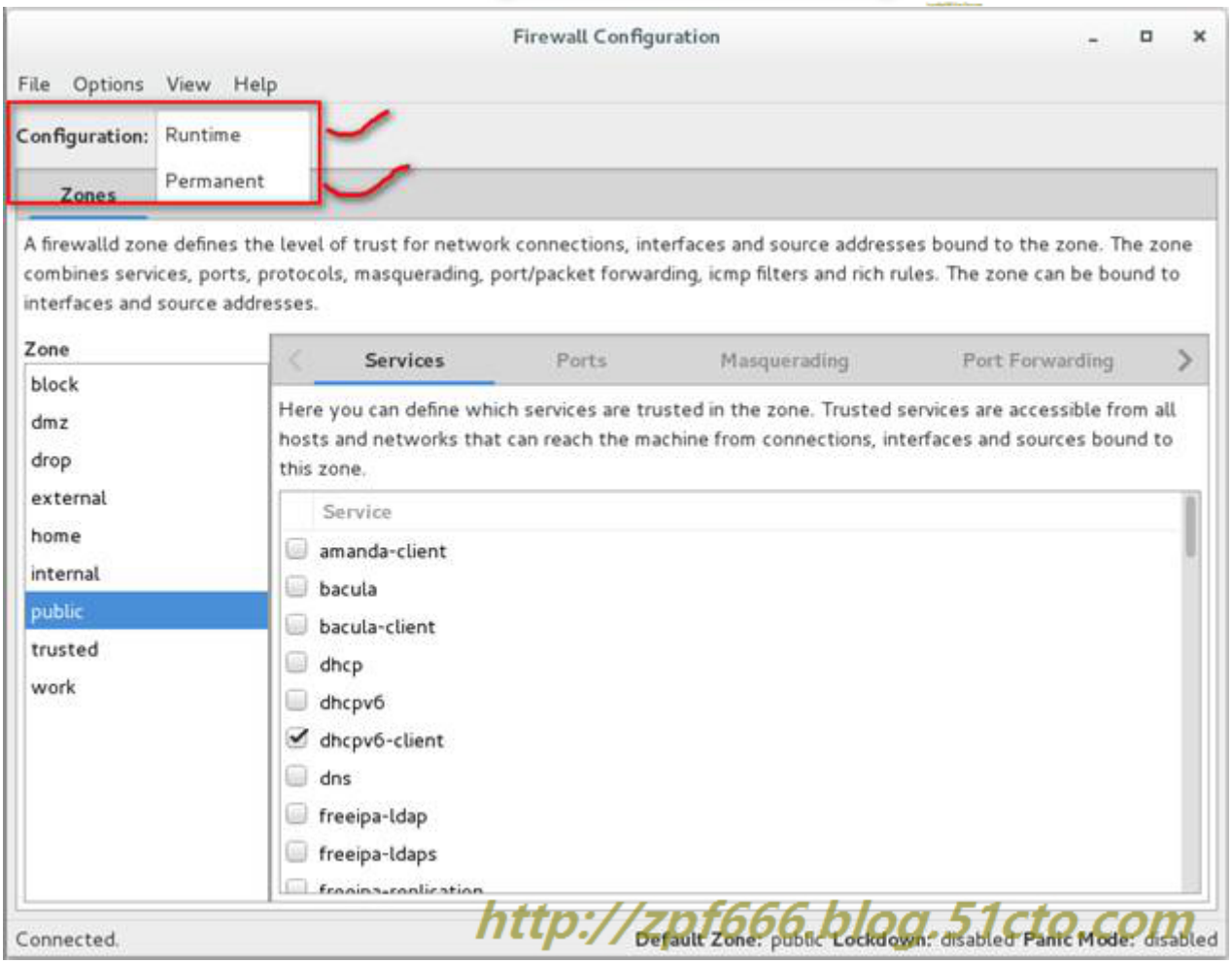
原创 Mr大表哥 2017-05-04 17:48:15 评论(4) 1090人阅读

博主QQ：819594300
博客地址：http://zpf666.blog.51cto.com/
有什么疑问的朋友可以联系博主，博主会帮你们解答，谢谢支持！ Firewalld防火墙：

1、使用图像化的firewall-config 工具：

firewall-config 工具里有一个标记为 Configuration 的下拉菜单，可以在运 行时间（即临时模式 runtime）和永久(permanent) 两种模式之间进行选择。要注意，如果您选择了 Permanent，在左上角会出现一排附加的图标。因为不能在运行模式下改变一个服务参数，所以这些图标仅在永久配置模式中出现。

```
[root@localhost ~]# firewall-config
```



由 firewalld 提供的是动态的防火墙服务，而非静态的。因为配置的改变可以随时随地立刻执行，不再需 要保存或者执行这些改变。现行网络连接的意外中断不会发生，正如防火墙的所有部分都不需要重新下载。

提供命令行客户端，firewall-cmd:

firewalld 和 iptables service 之间最本质的不同是：
iptables service在 /etc/sysconfig/iptables中储存配置，而 firewalld将配置储存在 /usr/lib/firewalld/（系统配置，尽量不要修改） 和/etc/firewalld/（用户配置地址）中的各种 XML 文件里。
使用 iptables service，每一个单独更改意味着清除所有旧有的规则和从/etc/sysconfig/iptables里读取所有新的规则，然而使用 firewalld 却不会再创建任何新的规则；仅仅运行规则中的不同之处。因此，firewalld 可以在运行时间内，改变设置而不丢失现行连接。
2、对网络区（zone）的理解：
基于用户对网络中设备和交通所给与的信任程度，防火墙可以用来将网络分割成不同的区域。 NetworkManager（网络管理器） 通知 firewalld 一个接口归属某个区域。接口所分配的区域可以由 NetworkManager 改变，也可以通过能为您打开相关NetworkManager 窗口的 firewall-config 工具 进行。
在/etc/firewalld/的区域设定是一系列可以被快速执行到网络接口的预设定。
数据包妖姬纳入到内核必须要通过9个zone区域中的一个，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）。
一个网卡最多绑定到一个zone（可以不绑定任何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。（echo-reply 应答）、（echo-request 请求）
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。
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）
任何接收的网络数据包都被丢弃，没有任何回复。仅能有发送出去的网络连接。
这个类似与我们之前使用iptables -j drop。
阻塞/限制区域（Block Zone）
阻塞区域会拒绝进入的网络连接，返回icmp-host-prohibited（ICMP-主机-禁止），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。
任何接收的网络连接都被 IPv4 的 icmp-host-prohibited 信息和 IPv6 的 icmp6-admprohibited 信息所拒绝。
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（即默认）zone。
在公共区域内使用，不能相信网络内的其他计算机不会对您的计算机造成危害，只能接收经过选取的连接。
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。

特别是为路由器启用了伪装功能的外部网。您不能信任来自网络的其他计算，不能相信它们不会对您的计算机造成危害，只能接收经过选择的连接

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。

用于您的非军事区内的电脑，此区域内可公开访问，可以有限地进入您的内部网络，仅仅接收经过 选择的连接。

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。

用于工作区。您可以基本相信网络内的其他电脑不会危害您的电脑。仅仅接收经过选择的连接。

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。

用于家庭网络。您可以基本信任网络内的其他计算机不会危害您的计算机。仅仅接收经过选择的连接。

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。

用于内部网络。您可以基本上信任网络内的其他计算机不会威胁您的计算机。仅仅接受经过选择的连接。

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

3、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。但是一个zone里可以有多个source。

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（接口），则么该 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。但是一个zone里可以有多个接口。

3）如果上述两个原则都不满足，那么缺省的 zone 将被应用。

我们可以使用任何一种 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。命令行工具firewall-cmd支持全部防火墙特性。

Firewalls一般应用：

1）获取firewalld状态

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

2）在不改变状态的条件下重新加载防火墙（如果使用--complete-reload，状态信息将会丢失。）

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

3）获取支持的区域列表

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

9个区域全在这

4）获取所有支持的服务（这条命令输出用空格分隔的列表）

```
[root@localhost ~]# firewall-cmd --get-services
RH-Satellite-6 amanda-client bacula bacula-client dhcp dhcpv6 dhcpv6-client dns freeipa-ldap freeipa-ldaps fr
eeipa-replication ftp high-availability http https imaps ipp ipp-client ipsec iscsi-target kerberos kpasswd l
dap ldaps libvirt libvirt-tls mdns mountd ms-wbt mysql nfs ntp openvpn pmcd pmproxy pmwebapi pmwebapis pop3s
postgresql proxy-dhcp radius rpc-bind rsyncd samba samba-client smtp ssh telnet tftp tftp-client transmission
-client vdsm vnc-server wbem-https
[root@localhost ~]#
```

说明：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。

列出默认有效的服务，可以进入到下面的目录中也能够取得。

```
[root@localhost ~]# cd /usr/lib/firewalld/services/
[root@localhost services]# ls
amanda-client.xml         high-availability.xml  ldap.xml         pmproxy.xml          samba.xml
bacula-client.xml         https.xml              libvirt-tls.xml  pmwebapis.xml        smtp.xml
bacula.xml                http.xml               libvirt.xml      pmwebapi.xml         ssh.xml
dhcpv6-client.xml         imaps.xml              mdns.xml         pop3s.xml            telnet.xml
dhcpv6.xml                ipp-client.xml         mountd.xml       postgresql.xml       tftp-client.xml
dhcp.xml                  ipp.xml                ms-wbt.xml       proxy-dhcp.xml       tftp.xml
dns.xml                   ipsec.xml              mysql.xml        radius.xml           transmission-client.xml
freeipa-ldaps.xml         iscsi-target.xml       nfs.xml          RH-Satellite-6.xml   vdsm.xml
freeipa-ldap.xml          kerberos.xml           ntp.xml          rpc-bind.xml         vnc-server.xml
freeipa-replication.xml   kpasswd.xml            openvpn.xml      rsyncd.xml           wbem-https.xml
ftp.xml                   ldaps.xml              pmcd.xml         samba-client.xml
[root@localhost services]#
```

http://zpf666.blog.51cto.com

创建自定义服务的方法：

举例：比如，现在我想添加一个rhmp服务，端口号1935。

①首先任选一个服务复制过来。

```
[root@localhost services]# pwd
/usr/lib/firewalld/services
[root@localhost services]# cp ssh.xml /etc/firewalld/services/
[root@localhost services]# cd /etc/firewalld/services/
[root@localhost services]# ls
ssh.xml
[root@localhost services]#
```

http://zpf666.blog.51cto.com

②接下来把复制过来的文件重命名为“rtmp.xml”。

```
[root@localhost services]# pwd
/etc/firewalld/services
[root@localhost services]# mv ssh.xml rtmp.xml
[root@localhost services]# ls
rtmp.xml
[root@localhost services]#
```


③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。

```
[root@localhost services]# vim rtmp.xml
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <service>
3   <short>RTMP</short>
4   <description>aaaaaaaaaaaaaaaaaaaaa.</description>
5   <port protocol="tcp" port="1935"/>
6 </service>
```


④重启firewalld服务或者重新加载设置，以激活这些设置（此命令完后服务还没开启）。

```
[root@localhost services]# firewall-cmd --reload
success
[root@localhost services]#
```

⑤确认服务是否已经被Firewall防火墙的默认区域（public）所支持（支持不等于开启）

```
[root@localhost services]# firewall-cmd --get-services
RH-Satellite-6 amanda-client bacula bacula-client dhcp dhcpv6 dhcpv6-client dns freeipa-ldap freeipa-ldaps fr
eeipa-replication ftp high-availability http https imaps ipp ipp-client ipsec iscsi-target kerberos kpasswd l
dap ldaps libvirt libvirt-tls mdns mountd ms-wbt mysql nfs ntp openvpn pmcd pmproxy pmwebapi pmwebapis pop3s
postgresql proxy-dhcp radius rpc-bind rsyncd rtmp samba samba-client smtp ssh telnet tftp tftp-client transmi
ssion-client vdsm vnc-server wbem-https
[root@localhost services]#
```

⑥启用新服务让其在默认区域（public）生效

```
[root@localhost services]# firewall-cmd --list-services
dhcpv6-client ssh
[root@localhost services]# firewall-cmd --add-service=rtmp
success
[root@localhost services]# firewall-cmd --list-services
dhcpv6-client rtmp ssh
[root@localhost services]#
```


5）获取所有支持的ICMP类型（这条命令输出用空格分隔的列表）

```
[root@localhost ~]# firewall-cmd --get-icmptypes
destination-unreachable echo-reply echo-request parameter-problem redirect router-advertisement router-solici
tation source-quench time-exceeded
[root@localhost ~]#
```

6）列出全部启用的区域的特性（即查询当前防火墙策略）

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@localhost ~]# 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:

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



```
external
  interfaces:
  sources:
  services: ssh
  ports:
  masquerade: yes
  forward-ports:
  icmp-blocks:
  rich rules:

home
  interfaces:
  sources:
  services: dhcpv6-client ipp-client mdns samba-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

```
internal
  interfaces:
  sources:
  services: dhcpv6-client ipp-client mdns samba-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

public (default)
  interfaces:
  sources:
  services: dhcpv6-client rtmp ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

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



```
work
  interfaces:
  sources:
  services: dhcpv6-client ipp-client 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 说明:rich-rules是自定义规则）、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。

①不指定区域，则显示默认区域

```
[root@localhost ~]# firewall-cmd --list-all
public (default)
  interfaces:
  sources:
  services: dhcpv6-client rtmp ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

[root@localhost ~]#
```


②指定区域

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

[root@localhost ~]#
```


8）查看默认区域（在文件/etc/firewalld/firewalld.conf中定义成DefaultZone=public）

```
[root@localhost ~]# firewall-cmd --get-default-zone
public
[root@localhost ~]#
```

9）设置默认区域（格式：firewall-cmd --set-default-zone=区域名）

说明：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。

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

10）获取活动的区域

说明：如果有活跃区域。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

```
[root@localhost ~]# firewall-cmd --get-active-zones
[root@localhost ~]#
```

因我没有给任何区域配置源，也没有和任何网卡接口绑定，所以就没有活跃区域，默认区域不一定就是活跃区域（即默认区域和活跃不活跃没关系。）

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）（格式：firewall-cmd--get-zone-of-interface=接口名）

```
[root@localhost ~]# firewall-cmd --get-zone-of-interface=eno16777736
no zone
[root@localhost ~]#
```

因为我没有给eno16777736网卡绑定到任何区域，所以是“no zone”

12）将接口（网卡）增加到区域

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。

```
[root@localhost ~]# firewall-cmd --add-interface=eno16777736
success
[root@localhost ~]# firewall-cmd --get-zone-of-interface=eno16777736
drop
[root@localhost ~]# firewall-cmd --get-default-zone
drop
[root@localhost ~]#
```

因为默认区域是drop，我们可以在把接口绑定在默认区域drop后，查看一下是不是绑定在默认区域drop下。

13）修改接口所属区域（格式：firewall-cmd [--zone=]--change-interface=接口名）

说明：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，--add-interface选项不能被添加到新的区域。

```
[root@localhost ~]# firewall-cmd --zone=public --change-interface=eno16777736
success
[root@localhost ~]# firewall-cmd --get-zone-of-interface=eno16777736
public
[root@localhost ~]#
[root@localhost ~]# firewall-cmd --zone=drop --add-interface=eno16777736
Error: ZONE_CONFLICT
[root@localhost ~]#
```

当接口绑定有一个区域的时候。再用--add就不行可，只能用--change。

14）从区域中删除一个接口（格式：firewall-cmd [--zone=]--remove-interface=接口名）

说明：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

```
[root@localhost ~]# firewall-cmd --get-active-zones
public
  interfaces: eno16777736
[root@localhost ~]# firewall-cmd --zone=public --remove-interface=eno16777736
success
[root@localhost ~]# firewall-cmd --get-active-zones
[root@localhost ~]#
```

15）查询区域中是否包含某接口（格式：firewall-cmd [--zone=]--query-interface=接口名）（如果区域被省略了，将使用默认区域）

```
[root@localhost ~]# firewall-cmd --query-interface=eno16777736
no
[root@localhost ~]#
```

16）列举区域中启用的服务（格式：firewall-cmd [--zone=]--list-services）（如果区域被省略了，将使用默认区域）

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

因为默认区域我没有开启任何服务，所以没有。

（查看home区域中启用服务）

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

17）启用应急模式阻断所有网络连接，以防出现紧急状况

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

18）禁用应急模式

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

19）查询应急模式

```
[root@localhost ~]# firewall-cmd --query-panic
no
[root@localhost ~]#
```

应急模式处于关闭状态

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

Firewall防火墙处理运行时区域：

说明：**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）启用某区域中的某一种服务（即给某个区域开启某个服务）

说明：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
  interfaces:
  sources:
  services: ipp-client
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

[root@localhost ~]#
```


（上图中使区域中的 ipp-client 服务生效60秒）

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

```
[root@localhost ~]# firewall-cmd --add-service=http
success
[root@localhost ~]# firewall-cmd --list-all
drop (default)
  interfaces:
  sources:
  services: http
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

[root@localhost ~]#
```


2）禁用区域中的某种服务即关闭某个区域的某个服务

（格式：firewall-cmd [--zone=区域] --remove-service=服务）

说明：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。

例子：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
drop (default)
  interfaces:
  sources:
  services:
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

[root@localhost ~]#
```


3）查询区域中是否启用了特定服务（格式：firewall-cmd [--zone=区域] --query-service=服务）

```
[root@localhost ~]# firewall-cmd --zone=trusted --query-service=http
no
[root@localhost ~]#
```

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（格式：firewall-cmd [--zone=区域] --add-port=端口/协议或者一段端口组/协议）

说明：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@localhost ~]# firewall-cmd --zone=public --add-port=8080/tcp
success
[root@localhost ~]# firewall-cmd --zone=public --add-port=20-21/tcp
success
[root@localhost ~]# firewall-cmd --zone=public --list-all
public
  interfaces:
  sources:
  services: dhcpv6-client rtmp ssh
  ports: 20-21/tcp 8080/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

[root@localhost ~]#
```


5）禁用/移除端口和协议组合（格式：firewall-cmd[--zone=区域] --remove-port=端口/协议或者一段端口组/协议）

```
[root@localhost ~]# firewall-cmd --zone=public --remove-port=8080/tcp
success
[root@localhost ~]# firewall-cmd --zone=public --list-all
public
  interfaces:
  sources:
  services: dhcpv6-client rtmp ssh
  ports: 20-21/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

[root@localhost ~]#
```


6）查询区域中是否启用了端口和协议组合（格式：firewall-cmd [--zone=区域] --remove-port=端口/协议或者一段端口组/协议）

```
[root@localhost ~]# firewall-cmd --zone=public --query-port=20-21/tcp
yes
[root@localhost ~]# firewall-cmd --zone=public --list-all
public
  interfaces:
  sources:
  services: dhcpv6-client rtmp ssh
  ports: 20-21/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

[root@localhost ~]#
```

格外小知识：Centos7挂光盘“mount /dev/sr0 /media”，跟6系列系统不一样。