firewall-cmd

Linux上新用的防火墙软件，跟iptables差不多的工具

补充说明

firewall-cmd 是 firewalld的字符界面管理工具，firewalld是centos7的一大特性，最大的好处有两个：支持动态更新，不用重启服务；第二个就是加入了防火墙的“zone”概念。

firewalld跟iptables比起来至少有两大好处：

1. firewalld可以动态修改单条规则，而不需要像iptables那样，在修改了规则后必须得全部刷新才可以生效。
2. firewalld在使用上要比iptables人性化很多，即使不明白“ 五张表五条链 ”而且对TCP/IP协议也不理解也可以实现大部分功能。

firewalld自身并不具备防火墙的功能，而是和iptables一样需要通过内核的netfilter来实现，也就是说firewalld和 iptables一样，他们的作用都是用于维护规则，而真正使用规则干活的是内核的netfilter，只不过firewalld和iptables的结 构以及使用方法不一样罢了。

命令格式

```
firewall-cmd [选项 … ]
```

选项

通用选项

```
-h, --help    # 显示帮助信息；
-V, --version # 显示版本信息.（这个选项不能与其他选项组合）；
-q, --quiet   # 不打印状态消息；
```

状态选项

```
--state                # 显示firewalld的状态；
--reload               # 不中断服务的重新加载；
--complete-reload      # 中断所有连接的重新加载；
--runtime-to-permanent # 将当前防火墙的规则永久保存；
--check-config         # 检查配置正确性；
```

日志选项

```
--get-log-denied         # 获取记录被拒绝的日志；
--set-log-denied=<value> # 设置记录被拒绝的日志，只能为
'all','unicast','broadcast','multicast','off' 其中的一个；
```

实例

```
#安装firewalld
yum install firewalld firewall-config
```

```
systemctl start  firewalld # 启动
systemctl stop firewalld  # 停止
systemctl enable firewalld # 启用自动启动
systemctl disable firewalld # 禁用自动启动
systemctl status firewalld # 或者 firewall-cmd --state 查看状态

# 关闭服务的方法
# 你也可以关闭目前还不熟悉的FirewallD防火墙，而使用iptables命令如下：

systemctl stop firewalld
systemctl disable firewalld
yum install iptables-services
systemctl start iptables
systemctl enable iptables
```

配置firewalld

```
firewall-cmd --version  # 查看版本
firewall-cmd --help     # 查看帮助

# 查看设置：
firewall-cmd --state  # 显示状态
firewall-cmd --get-active-zones  # 查看区域信息
firewall-cmd --get-zone-of-interface=eth0  # 查看指定接口所属区域
firewall-cmd --panic-on  # 拒绝所有包
firewall-cmd --panic-off  # 取消拒绝状态
firewall-cmd --query-panic  # 查看是否拒绝

firewall-cmd --reload # 更新防火墙规则
firewall-cmd --complete-reload
#两者的区别就是第一个无需断开连接，就是firewalld特性之一动态添加规则，第二个需要断开连接，类似重启服务


# 将接口添加到区域，默认接口都在public
firewall-cmd --zone=public --add-interface=eth0
# 永久生效再加上 --permanent 然后reload防火墙

# 设置默认接口区域，立即生效无需重启
firewall-cmd --set-default-zone=public

# 查看所有打开的端口：
firewall-cmd --zone=dmz --list-ports

# 加入一个端口到区域：
firewall-cmd --zone=dmz --add-port=8080/tcp
# 若要永久生效方法同上

# 打开一个服务，类似于将端口可视化，服务需要在配置文件中添加/etc/firewalld 目录下有services文件夹，这个不详细说了，详情参考文档
firewall-cmd --zone=work --add-service=smtp
```

```
# 移除服务
firewall-cmd --zone=work --remove-service=smtp

# 显示支持的区域列表
firewall-cmd --get-zones

# 设置为家庭区域
firewall-cmd --set-default-zone=home

# 查看当前区域
firewall-cmd --get-active-zones

# 设置当前区域的接口
firewall-cmd --get-zone-of-interface=enp03s

# 显示所有公共区域（public）
firewall-cmd --zone=public --list-all

# 临时修改网络接口（enp0s3）为内部区域（internal）
firewall-cmd --zone=internal --change-interface=enp03s

# 永久修改网络接口enp03s为内部区域（internal）
firewall-cmd --permanent --zone=internal --change-interface=enp03s
```

服务管理

```
# 显示服务列表
Amanda, FTP, Samba和TFTP等最重要的服务已经被FirewallD提供相应的服务，可以使用如下命令查看：

firewall-cmd --get-services

# 允许SSH服务通过
firewall-cmd --new-service=ssh

# 禁止SSH服务通过
firewall-cmd --delete-service=ssh

# 打开TCP的8080端口
firewall-cmd --enable ports=8080/tcp

# 临时允许Samba服务通过600秒
firewall-cmd --enable service=samba --timeout=600

# 显示当前服务
firewall-cmd --list-services

# 添加HTTP服务到内部区域（internal）
firewall-cmd --permanent --zone=internal --add-service=http
firewall-cmd --reload     # 在不改变状态的条件下重新加载防火墙
```

端口管理

```
# 打开443/TCP端口
firewall-cmd --add-port=443/tcp

# 永久打开3690/TCP端口
firewall-cmd --permanent --add-port=3690/tcp

# 永久打开端口好像需要reload一下，临时打开好像不用，如果用了reload临时打开的端口就失效了
# 其它服务也可能是这样的，这个没有测试
firewall-cmd --reload

# 查看防火墙，添加的端口也可以看到
firewall-cmd --list-all
```

直接模式

```
# FirewallD包括一种直接模式，使用它可以完成一些工作，例如打开TCP协议的9999端口

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

自定义服务管理

选项

```
（末尾带有 [P only] 的话表示该选项除了与--permanent之外，不能与其他选项一同使用！）
--new-service=<服务名> 新建一个自定义服务 [P only]
--new-service-from-file=<文件名> [--name=<服务名>]
        从文件中读取配置用以新建一个自定义服务 [P only]
--delete-service=<服务名>
        删除一个已存在的服务 [P only]
--load-service-defaults=<服务名>
                        Load icmptype default settings [P only]
--info-service=<服务名>
        显示该服务的相关信息
--path-service=<服务名>
        显示该服务的文件的相关路径 [P only]
--service=<服务名> --set-description=<描述>
        给该服务设置描述信息 [P only]
--service=<服务名> --get-description
        显示该服务的描述信息 [P only]
--service=<服务名> --set-short=<描述>
        给该服务设置一个简短的描述 [P only]
--service=<服务名> --get-short
        显示该服务的简短描述 [P only]

--service=<服务名> --add-port=<端口号>[-<端口号>]/<protocol>
        给该服务添加一个新的端口(端口段) [P only]

--service=<服务名> --remove-port=<端口号>[-<端口号>]/<protocol>
        从该服务上移除一个端口(端口段) [P only]
```

```
--service=<服务名> --query-port=<端口号>[-<端口号>]/<protocol>
        查询该服务是否添加了某个端口(端口段) [P only]

--service=<服务名> --get-ports
        显示该服务添加的所有端口 [P only]

--service=<服务名> --add-protocol=<protocol>
        为该服务添加一个协议 [P only]

--service=<服务名> --remove-protocol=<protocol>
        从该服务上移除一个协议 [P only]

--service=<服务名> --query-protocol=<protocol>
        查询该服务是否添加了某个协议 [P only]

--service=<服务名> --get-protocols
        显示该服务添加的所有协议 [P only]

--service=<服务名> --add-source-port=<端口号>[-<端口号>]/<protocol>
        添加新的源端口(端口段)到该服务 [P only]

--service=<服务名> --remove-source-port=<端口号>[-<端口号>]/<protocol>
        从该服务中删除源端口(端口段) [P only]

--service=<服务名> --query-source-port=<端口号>[-<端口号>]/<protocol>
        查询该服务是否添加了某个源端口(端口段) [P only]

--service=<服务名> --get-source-ports
        显示该服务所有源端口 [P only]

--service=<服务名> --add-module=<module>
        为该服务添加一个模块 [P only]
--service=<服务名> --remove-module=<module>
        为该服务移除一个模块 [P only]
--service=<服务名> --query-module=<module>
        查询该服务是否添加了某个模块 [P only]
--service=<服务名> --get-modules
        显示该服务添加的所有模块 [P only]
--service=<服务名> --set-destination=<ipv>:<address>[/<mask>]
                      Set destination for ipv to address in service [P only]
--service=<服务名> --remove-destination=<ipv>
                      Disable destination for ipv i service [P only]
--service=<服务名> --query-destination=<ipv>:<address>[/<mask>]
                      Return whether destination ipv is set for service [P
only]
--service=<服务名> --get-destinations
                      List destinations in service [P only]
```

控制端口 / 服务

可以通过两种方式控制端口的开放，一种是指定端口号另一种是指定服务名。虽然开放 http 服务就是开放了80端口，但是还是不能通过端口号来关闭，也就是说通过指定服务名开放的就要通过指定服务名关闭；通过指定端口号开放的就要通过指定端口号关闭。还有一个要注意的就是指定端口的时候一定要指定是什么协议tcp 还是 udp知道这个之后以后就不用每次先关防火墙了，可以让防火墙真正的生效。

```
firewall-cmd --add-service=mysql          # 开放mysql端口
firewall-cmd --remove-service=http        # 阻止http端口
firewall-cmd --list-services              # 查看开放的服务
firewall-cmd --add-port=3306/tcp          # 开放通过tcp访问3306
firewall-cmd --remove-port=80tcp          # 阻止通过tcp访问3306
firewall-cmd --add-port=233/udp           # 开放通过udp访问233
firewall-cmd --list-ports                 # 查看开放的端口
```

伪装 IP

```
firewall-cmd --query-masquerade # 检查是否允许伪装IP
firewall-cmd --add-masquerade   # 允许防火墙伪装IP
firewall-cmd --remove-masquerade# 禁止防火墙伪装IP
```

端口转发

端口转发可以将指定地址访问指定的端口时，将流量转发至指定地址的指定端口。转发的目的如果不指定 ip 的话就默认为本机，如果指定了 ip 却没指定端口，则默认使用来源端口。 如果配置好端口转发之后不能用，可以检查下面两个问题：

1. 比如我将 80 端口转发至 8080 端口，首先检查本地的 80 端口和目标的 8080 端口是否开放监听了
2. 其次检查是否允许伪装 IP没允许的话要开启伪装 IP

```
firewall-cmd --add-forward-port=port=80:proto=tcp:toport=8080   # 将80端口的流量转发至8080
firewall-cmd --add-forward-port=port=80:proto=tcp:toaddr=192.168.0.1 # 将80端口的流量转发至192.168.0.1
firewall-cmd --add-forward-port=port=80:proto=tcp:toaddr=192.168.0.1:toport=8080 # 将80端口的流量转发至192.168.0.1的8080端口
```

1. 当我们想把某个端口隐藏起来的时候，就可以在防火墙上阻止那个端口访问，然后再开一个不规则的端口，之后配置防火墙的端口转发，将流量转发过去。
2. 端口转发还可以做流量分发，一个防火墙拖着好多台运行着不同服务的机器，然后用防火墙将不同端口的流量转发至不同机器。