链滴

Firewall-cmd 防火墙配置

作者：lewsuy

原文链接：https://ld246.com/article/1679461555708

来源网站：链滴

<h2 id="补充说明">补充说明</h2>
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 比起来至少有一大好处：
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，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etfilter，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 ...]
<h2 id="选项">选项</h2>
<pre><code class="highlight-chroma">通用选项
-h, --help   # 显帮助信息;
-V, --version # 显版本信息. （这个选项不能与其他选项组合）;
-q, --quiet   # 不印状态消息;
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-per anent # 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' 其中的个;
</code></pre>
<h2 id="实例">实例</h2>
<h2 id="安装firewalld-yum---install-firewalld-firewall-config--">安装 firewalld yum** install fi ewalld firewall-config**</h2>
<pre><code class="language-bash highlight-chroma"><span c ass="highlight-cl">systemctl start firewalld # 启动

systemctl status fi ewalld # 或者 firewall-cmd --state 查看状态

systemctl disable f rewalld # 停止

systemctl stop fir walld # 禁用
</code></pre>
<h2 id="关闭服务的方法">关闭服务的方法</h2>

<h2 id="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-">你也可以关闭前还不熟悉的 FirewallD 防火墙，而使用 iptables，命令如下：</h2>
<hr>
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
<hr>
<h2 id="查看设置-">查看设置：</h2>
**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
<h2 id="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-第二个需要断开接-类似重启服务">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，第个需要断开连接，类似重启服务</h2>
<hr>
<h2 id="将接口添加到区域-默认接口都在public">将接口添加到区域，默认接口都在 public</h2>
firewall-cmd --zone=public --add-interface=eth0
<h2 id="永久生效再加上---permanent-然后reload防火墙">永久生效再加上 --permanent 然后 reload 防火墙</h2>
<hr>
<h2 id="设置默认接口区域-立即生效无需重启">设置默认接口区域，立即生效无需重启</h2>
firewall-cmd --set-default-zone=public
<hr>
<h2 id="查看所有打开的端口-">查看所有打开的端口：</h2>
firewall-cmd --zone=dmz --list-ports
<hr>
<h2 id="加入一个端口到区域-">加入一个端口到区域：</h2>
firewall-cmd --zone=dmz --add-port=8080/tcp
<h2 id="若要永久生效方法同上">若要永久生效方法同上</h2>
<hr>
<h2 id="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">打开一个服务，类似于将端口可视化，服务需要在配置件中添加，/etc/firewalld 目录下有 services 文件夹，这个不详细说了，详情参考文档</h2>
firewall-cmd --zone=work --add-service=smtp
<hr>
<h2 id="移除服务">移除服务</h2>
firewall-cmd --zone=work --remove-service=smtp
<hr>
<h2 id="显示支持的区域列表">显示支持的区域列表</h2>
firewall-cmd --get-zones
<hr>
<h2 id="设置为家庭区域">设置为家庭区域</h2>
firewall-cmd --set-default-zone=home
<hr>

<h2 id="查看当前区域">查看当前区域</h2>
firewall-cmd --get-active-zones
<hr>
<h2 id="显示当前区域的接口">显示当前区域的接口</h2>
firewall-cmd --get-zone-of-interface=enp03s
<hr>
<h2 id="显示所有公共区域-public-">显示所有公共区域（public）</h2>
firewall-cmd --zone=public --list-all
<hr>
<h2 id="临时修改网络接口-enp0s3-为内部区域-internal-">临时修改网络接口（enp0s3）为内部域（internal）</h2>
firewall-cmd --zone=internal --change-interface=enp03s
<hr>
<h2 id="永久修改网络接口enp03s为内部区域-internal-">永久修改网络接口 enp03s 为内部区域（nternal）</h2>
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，可以使用如命令查看：
<hr>
firewall-cmd --get-services
<hr>
<h2 id="允许ssh服务通过-注-在-0-9-4-版本的firewalld上-不存在--enable-或---disable-参数来开服务--">允许 ssh 服务通过（注：在 0.9.4 版本的 firewalld 上，不存在--enable 或 --disable 参数开关服务。）</h2>
firewall-cmd --enable service=ssh
<hr>
<h2 id="禁止SSH服务通过">禁止 SSH 服务通过</h2>
firewall-cmd --disable service=ssh
<hr>
<h2 id="打开TCP的8080端口">打开 TCP 的 8080 端口</h2>
firewall-cmd --enable ports=8080/tcp
<hr>
<h2 id="临时允许Samba服务通过600秒">临时允许 Samba 服务通过 600 秒</h2>
firewall-cmd --enable service=samba --timeout=600
<hr>
<h2 id="显示当前服务">显示当前服务</h2>
firewall-cmd --list-services
<hr>
<h2 id="添加HTTP服务到内部区域-internal-">添加 HTTP 服务到内部区域（internal）</h2>
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
<h2 id="打开443-TCP端口">打开 443/TCP 端口</h2>
firewall-cmd --add-port=443/tcp
<hr>
<h2 id="永久打开3690-TCP端口">永久打开 3690/TCP 端口</h2>
firewall-cmd --permanent --add-port=3690/tcp
<hr>
<h2 id="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了">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 临时打开的端口就效了</h2>
<h2 id="其它服务也可能是这样的-这个没有测试">其它服务也可能是这样的，这个没有测试</h2>

firewall-cmd --reload
<hr>
<h2 id="查看防火墙-添加的端口也可以看到">查看防火墙，添加的端口也可以看到</h2>
firewall-cmd --list-all
直接模式
<h2 id="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">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</h2>
**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 [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 [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 >]/ ** 给该服务添一个新的端口(端口段) [P only]**
--service=< 服务名 > --remove-port=< 端口号 >[-< 端口号 >]/ ** 从该服上移除一个端口(端口段) [P only]**
--service=< 服务名 > --query-port=< 端口号 >[-< 端口号 >]/ ** 查询该服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= ** ** ** 为该服务添加一个协议 [P only]**
--service=< 服务名 > --remove-protocol= ** ** 从该服务上移除一个协议 [P only]
--service=< 服务名 > --query-protocol= ** ** 查询该服务是否添加了某个协议 [P only]/p>
--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 >]/ 添加新源端口(端口段)到该服务 [P only]
--service=< 服务名 > --remove-source-port=< 端口号 >[-< 端口号 >]/ 从服务中删除源端口(端口段) [P only]
--service=< 服务名 > --query-source-port=< 端口号 >[-< 端口号 >]/ ** 查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= ** ** ** 为该服务添加一个模块 [P only]**
--service=< 服务名 > --remove-module= ** ** ** 为该服务移除一个模块 [P only]**</p

--service=< 服务名 > --query-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]</p

--service=< 服务名 > --set-destination=: [/] ** **Set destination for ipv to addre
s in service [P only]
--service=< 服务名 > --remove-destination= ** ** Disable destination for ipv i servic
[P only]
--service=< 服务名 > --query-destination=: [/] ** **Return whether destination
pv 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
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
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
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
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
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
发之后不能用，可以检查下面两个问题：

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
了
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
至 8080
firewall-cmd --add-forward-port=port=80:proto=tcp:toaddr=192.168.0.1 # 将 80 端口的
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

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
的端口，之后配置防火墙的端口转发，将流量转发过去。
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
端口的流量转发至不同机器。

<hr>
1.firewalld 的基本使用
启动： systemctl start firewalld
停止： systemctl stop firewalld
查状态：systemctl status firewalld
禁用： systemctl disable firewalld
<hr>
在开机时启用一个服务：systemctl enable firewalld.service

在开机时禁用一个服务：systemctl disable firewalld.service
查看服务是否开机启动：systemctl is-enabled firewalld.service
查看已启动的服务列表：systemctl list-unit-files|grep enabled
查看启动失败的服务列表：systemctl --failed
<hr>
Firewall-cmd 防火墙配置 Firewall-cmd 防火墙配置
2.配置 firewalld-cmd
查看版本： ** ** ** ** firewall-cmd --version
查看帮助： ** ** ** ** firewall-cmd --help
显示状态： ** ** ** ** firewall-cmd --state
查看所有打开的端口： ** **firewall-cmd --zone=public --list-ports
更新防火墙规则： ** ** ** **firewall-cmd --reload
查看区域信息: ** ** ** **firewall-cmd --get-active-zones
查看指定接口所属区域： ** **firewall-cmd --get-zone-of-interface=eth0
拒绝所有包： ** ** ** **firewall-cmd --panic-on
取消拒绝状态： ** ** ** **firewall-cmd --panic-off
查看是否拒绝： ** ** ** **firewall-cmd --query-panic
显示所有公共区域（public）：firewall-cmd --zone=public --list-all
<hr>
3.那怎么开启一个端口呢
添加
firewall-cmd --zone=public(作用域) --add-port=80/tcp(端口和访问类型) --permanent(永生效)
firewall-cmd --zone=public --add-service=http --permanent
firewall-cmd --reload # 重新载入，更新防火墙规则
firewall-cmd --zone= public --query-port=80/tcp #查看
firewall-cmd --zone= public --remove-port=80/tcp --permanent # 删除
<hr>
firewall-cmd --list-services
firewall-cmd --get-services
firewall-cmd --add-service=
firewall-cmd --delete-service=
在每次修改端口和服务后/etc/firewalld/zones/public.xml 文件就会被修改,所以也可以在文件之间修改,然后重新加载
使用命令实际也是在修改文件，需要重新加载才能生效。
<hr>
4.详细使用
firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address="123.56.161.40" drop' #禁止 IP(123.56.161.140)访问机器
firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address="123.56.0.0/6" drop' #禁止一个 IP 段， 比如禁止 123.56.*.*
firewall-cmd --permanent --zone=public --add-rich-rule='rule family="ipv4" source address="192.168.0.4/24" service name="http" accept' //设置某个 ip 访问某个服务
firewall-cmd --permanent --zone=public --remove-rich-rule='rule family="ipv4" source ddress="192.168.0.4/24" service name="http" accept' //删除配置
firewall-cmd --permanent --add-rich-rule 'rule family=ipv4 source address=192.168.0.1/2 port port=80 protocol=tcp accept' //设置某个 ip 访问某个端口
firewall-cmd --permanent --remove-rich-rule 'rule family=ipv4 source address=192.168.01/2 port port=80 protocol=tcp accept' //删除配置
<hr>
**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
<hr>

firewall-cmd --add-forward-port=port=80:proto=tcp:toport=8080** **# 将 80 端口的流转发至 8080
firewall-cmd --add-forward-port=proto=80:proto=tcp:toaddr=192.168**.1.0.1 **# 将 80 端口的流量转发至 192.168.0.1
firewall-cmd --add-forward-port=proto=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
<hr>
<h2 id="重新加载配置">重新加载配置</h2>
**firewall-cmd --reload **
<h2 id="查询所有开放的端口-本次运行">查询所有开放的端口。本次运行</h2>
firewall-cmd --list-ports
80/tcp 8080/tcp
<hr>
<h2 id="查询某个端口是否开放-持久">查询某个端口是否开放。持久</h2>
firewall-cmd --query-port=8080/tcp --permanent
no
<hr>
<h2 id="开放某个端口-立即生效-本次运行">开放某个端口，立即生效。本次运行</h2>
firewall-cmd --add-port=80/tcp
success
<hr>
<h2 id="开放某个端口-重新加载配置后生效-持久">开放某个端口，重新加载配置后生效。持久</h>
firewall-cmd --add-port=3306/tcp --permanent
success
<hr>
<h2 id="开放多个不连续端口-立即生效-本次运行">开放多个不连续端口，立即生效。本次运行</h>
firewall-cmd --add-port=80/tcp --add-port=8080/tcp
success
<hr>
<h2 id="开放多个连续端口-立即生效-本次运行">开放多个连续端口，立即生效。本次运行</h2>
firewall-cmd --add-port=8080-8090/tcp
success
<hr>
<h2 id="开放多个连续端口-重新加载配置后生效-持久">开放多个连续端口，重新加载配置后生效。久</h2>
firewall-cmd --add-port=8080-8090/tcp --permanent
success
<hr>
<h2 id="关闭某个端口-立即生效-本次运行">关闭某个端口，立即生效。本次运行</h2>
firewall-cmd --remove-port=80/tcp
success
<hr>
双网卡只允许内网访问 ssh22 端口
firewall-cmd --set-default-zone=internal
firewall-cmd --zone=internal --add-interface=**ens160 --permanent **#内网网卡
firewall-cmd --zone=drop --change-interface=**ens192 --permanent **#公网网卡
firewall-cmd --zone=public --remove-service=ssh --permanent
<hr>
只允许固定 IP 访问 22 端口
rm -rf /usr/lib/firewalld/services/ssh.xml
firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="10.2.222./24" port protocol="udp" port="161" accept'

firewall-cmd --zone=internal --permanent --add-rich-rule='rule family="ipv4" source address="10.1.128.211" port protocol="tcp" port="22" accept'
firewall-cmd --zone=internal --permanent --add-rich-rule='rule family="ipv4" source address="10.1.1.6" port protocol="tcp" port="3306" accept'
firewall-cmd --zone=internal --permanent --add-rich-rule='rule family="ipv4" source address="10.0.0.0/8" port protocol="tcp" port="5601" accept'
firewall-cmd --zone=internal --permanent --add-rich-rule='rule family="ipv4" source address="10.2.222.56" port protocol="udp" port="161" accept'
<hr>
备注:
source address 也可以设置为单个 IP 地址,例如 192.168.1.1
port 可以为单个端口或端口范围，例如 1-10000
<hr>
删除：
<pre><code class="highlight-chroma">firewall-cmd --zone=public --permanent --remove-rich-rule=''

firewall-cmd --zone=public --permanent --remove-rich-rule='rule family="ipv4" source address="10.1.1.2" port port="22" protocol="tcp" accept'

firewall-cmd --zone=public --permanent --remove-rich-rule='rule family="ipv4" source address="10.1.2.3" port port="9100" protocol="tcp" accept'
</code></pre>
<hr>
firewalld 的 9 个 zone
1、zone 是 firewalld 的单位。默认使用 public zone
2、查看所有的 zone : firewall-cmd --get-zones
3、查看默认的 zone : firewall-cmd --get-default-zone
<hr>
4、9 个 zone 说明：
<pre><code class="highlight-chroma">drop(丢弃)任何接收到的网络数据都被丢弃，没有任何回复，公有发送出去的网络连接

block（限制）任接收的网络连接都被IPV4 的icmp-host-prohibited信息和IPV6的icmp6-adm-prohibited信息所拒绝

public(公共)在公区域内使用，不能相信网络内的其它计算机不会对你的计算机造成危害，只接收经过选取的连接

external（外部）别是为路由器启用了伪装功能的外部网。你不能信任来自网络的其它计算，不能信任它们不会对你的算机造成危害，只能接收经过选择的连接。

dmz(非军事区)用你的非军事区的电脑 ，此区域内可公开访问，可以有限的进入你的内部网络，仅接收经过选择的连接

work(工作)可以基信任网络内的其它计算机不会危害你的计算机，仅接收经过选择的连接。

</code></pre>

home(家庭)用于庭网络，可以基本信任网络内的其它计算机不会危害你的计算机，仅接收经过选择的连接。

internal（内部）于内部网络，可以基本信任网络内的其它计算机不会危害你的计算机，仅接收经过选择的连接

trusted(信任) 可收所有的网络连接
</code></pre>