链滴

firewalld 基础知识整理

作者：iwang-peng
原文链接：https://ld246.com/article/1579246437736
来源网站：链滴

什么是动态防火墙?
我们首先需要弄明白的第一个问题是到底什么是动态防火墙。为了解答这个问题，我们先来回忆下 iptables service 管理防火墙规则的模式：用户将新的防火墙规则添加进 /etc/sysconfig/iptables 配置文件当中，再执行命令 service iptables reload 使变更的规则生效。在这整个过程的背后，iptables service 首先对旧的防火墙规则进行了清空，然后重新完整地加载所有新的防火墙规则，而如果配了需要 reload 内核模块的话，过程背后还会包含卸载和重新加载内核模块的动作，而不幸的是，这动作很可能对运行中的系统产生额外的不良影响，特别是在网络非常繁忙的系统中。
如果我们把这种哪怕只修改一条规则也要进行所有规则的重新载入的模式称为静态防火墙的话，么 firewalld 所提供的模式就可以叫做动态防火墙，它的出现就是为了解决这一问题，任何规则的变都不需要对整个防火墙规则列表进行重新加载，只需要将变更部分保存并更新到运行中的 iptables 即。
这里有必要说明一下 firewalld 和 iptables 之间的关系， firewalld 提供了一个 daemon 和 servie，还有命令行和图形界面配置工具，它仅仅是替代了 iptables service 部分，其底层还是使用 iptabls 作为防火墙规则管理入口。firewalld 使用 python 语言开发，在新版本中已经计划使用 c++ 重写 demon 部分。
什么是区域(zone)?
firewalld 将网卡对应到不同的区域（zone），zone 默认共有 9 个，block dmz public external hoe internal public trusted work.
不同的区域之间的差异是其对待数据包的默认行为不同，根据区域名字我们可以很直观的知道该区域特征，在 CentOS7 系统中，默认区域被设置为 public.
在最新版本的 fedora（fedora21）当中随着 server 版和 workstation 版的分化则添加了两个不同自定义 zone FedoraServer 和 FedoraWorkstation 分别对应两个版本。
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。列表并简要说明如：
drop（丢弃）
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。
block（限制）
任何接收的网络连接都被 IPv4 的 icmp-host-prohibited 信息和 IPv6 的 icmp6-adm-prohibited 息所拒绝。
public（公共）
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接。
external（外部）
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计算机造成危害，只能接收经过选择的连接。
dmz（非军事区）
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，仅仅接收经过选的连接。
work（工作）
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。
home（家庭）
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
internal（内部）
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。
trusted（信任）
可接受所有的网络连接。
指定其中一个区域为默认区域是可行的。当接口连接加入了 NetworkManager，它们就被分配为默区域。安装时，firewalld 里的默认区域被设定为公共区域。
使用下面的命令分别列出所有支持的 zone 和查看当前的默认 zone：
<pre><code class="highlight-chroma">firewall-cmd --get-zones
block dmz public

xternal home internal public trusted work

firewall-cmd --get default-zone
public

</code></pre>
<p>基本指令参数：</p>
<p>Target：目标
icmp-block-inversion：ICMP 协议类型黑白名单开关（yes/no）
Interfaces：关联的网卡接口
sources：来源，可以是 IP 地址，也可以是 mac 地址
services：允许的服务
ports：允许的目标端口，即本地开放的端口
protocols：允许通过的协议
masquerade：是否允许伪装（yes/no），可改写来源 IP 地址及 mac 地址
forward-ports：允许转发的端口
source-ports：允许的来源端口
icmp-blocks：可添加 ICMP 类型，当 icmp-block-inversion 为 no 时，这些 ICMP 类型被拒绝；当 cmp-block-inversion 为 yes 时，这些 ICMP 类型被允许。
rich rules：富规则，即更细致、更详细的防火墙规则策略，它的优先级在所有的防火墙策略中也是最的。</p>
<p>什么是服务?
在 /usr/lib/firewalld/services/ 目录中，还保存了另外一类配置文件，每个文件对应一项具体的网络务，如 ssh 服务等.
与之对应的配置文件中记录了各项服务所使用的 tcp/udp 端口，在最新版本的 firewalld 中默认已经义了 70+ 种服务供我们使用.
当默认提供的服务不够用或者需要自定义某项服务的端口时，我们需要将 service 配置文件放置在 /et /firewalld/services/ 目录中.
service 配置的好处显而易见:
第一，通过服务名字来管理规则更加人性化，
第二，通过服务来组织端口分组的模式更加高效，如果一个服务使用了若干个网络端口，则服务的配文件就相当于提供了到这些端口的规则管理的批量操作快捷方式。</p>
<p>每加载一项 service 配置就意味着开放了对应的端口访问，使用下面的命令分别列出所有支持的 s rvice 和查看当前 zone 种加载的 service：</p>
<pre><code class="highlight-chroma">firewall-cmd --get-services
RH-Satellite-6 am nda-client bacula bacula-client dhcp dhcpv6 dhcpv6-client dns ftp high-availability http https maps ipp ipp-client ipsec kerberos kpasswd ldap ldaps libvirt libvirt-tls mdns mountd ms-wbt mysql nfs ntp openvpn pmcd pmproxy pmwebapi pmwebapis pop3s postgresql proxy-dhcp r dius rpc-bind samba samba-client smtp ssh telnet tftp tftp-client transmission-client vnc-serv r wbem-https

firewall-cmd --list services
dhcpv6-client ssh

</code></pre>
<pre><code class="highlight-chroma">动态添加一条防火墙规则如下:

<span class="highlight-line"><span class="highlight-cl">假设自定义的 ssh

口号为 12222，使用下面的命令来添加新端口的防火墙规则：
firewall-cmd --ad
-port=12222/tcp --permanent
firewall-cmd --re
ove-port=80/tcp --permanent

如果需要使规则保
到 zone 配置文件，则需要加参数 –permanent

</code></pre>
以下举几个在工作中常用的例子：

获取默认区域的网络设置

<pre><code class="highlight-chroma"><span class="highlight
cl">firewall-cmd --get-default-zone
public

</code></pre>

设置默认区域

<pre><code class="highlight-chroma"><span class="highlight
cl">firewall-cmd --set-default-zone=work
success

</code></pre>
注意: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
。

获取活动的区域

<code>firewall-cmd --get-active-zones</code>

根据接口获取区域

<code>firewall-cmd --get-zone-of-interface=eth0</code>

将接口增加到区域

<code>firewall-cmd [--zone=<zone>] --add-interface=<interface></code></
>
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
载后将重新应用。

修改接口所属区域

<code>firewall-cmd [--zone=<zone>] --change-interface=<interface></code

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
加到新的区域。

从区域中删除一个接口

<code>firewall-cmd [--zone=<zone>] --remove-interface=<interface></code>

查询区域中是否包含某接口

<code>firewall-cmd [--zone=<zone>] --query-interface=<interface></code>

在 public 区域开放 https 服务

<code>firewall-cmd --zone=public --add-service=https</code>

取消开放 https 服务，即禁止 https 服务

<code>firewall-cmd --zone=public --remove-service=https</code>

开放 8080 和 8081 端口

<code>firewall-cmd --zone=public --add-port=8080-8081/tcp</code>

查询 public 区域开放了哪些端口

<code>firewall-cmd --zone=public --list-ports</code>

允许 icmp 协议流量，即允许 ping

<code>firewall-cmd --zone=public --add-protocol=icmp</code>

取消允许 icmp 协议的流量，即禁 ping

<code>firewall-cmd --zone=public --remove-protocol=icmp</code>

查询 public 区域开放了哪些协议

<code>firewall-cmd --zone=public --list-protocols</code>

将原本访问本机 888 端口的流量转发到本机 22 端口

<code>firewall-cmd --zone=public --add-forward-port=port=888:proto=tcp:toport=22</code>

在区域中启用端口转发或映射

<code>firewall-cmd [--zone=<zone>] --add-forward-port=port=<port>[-<ort>]:proto=<protocol> { :toport=<port>[-<port>] | :toaddr=<address> | :toport=<port>[-<port>]:toaddr=<address> }</code>
将原本访问本机 888 端口的流量转发到 ip 为 192.168.2.208 的主机的 22 端口，需要开启 masuerade

永久禁止区域的端口转发或者端口映射

<code>firewall-cmd --permanent [--zone=<zone>] --remove-forward-port=port=lt;port>[-<port>]:proto=<protocol> { :toport=<port>[-<port>] | :toadr=<address> | :toport=<port>[-<port>]:toaddr=<address> }</code></

>

查询区域的端口转发或者端口映射状态

<pre><code class="language-firewall-cmd highlight-chroma"><span class="highlight-line"

firewall-cmd --zo
e=public --add-masquerade
firewall-cmd --zo
e=public --add-forward port=port=888:proto=tcp:toport=22:toaddr=192.168.2.208

</code></pre>
接下来我们来看富规则的设置，即 rich rules

允许 192.168.2.208 主机的所有流量

<code>firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" source address=
192.168.2.208" accept"</code>

允许 192.168.2.208 主机的 icmp 协议，即允许 192.168.2.208 主机 ping

<code>firewall-cmd --add-rich-rule="rule family="ipv4" source address="192.168.2.208"
protocol value="icmp" accept"</code>

取消允许 192.168.2.208 主机的所有流量

<code>firewall-cmd --zone=public --remove-rich-rule="rule family="ipv4" source addre
s="192.168.2.208" accept"</code>

允许 192.168.2.208 主机访问 ssh 服务

<code>firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" source address=
192.168.2.208" service name="ssh" accept"</code>

禁止 192.168.2.208 访问 https 服务，并返回错误信息

<code>firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" source address=
192.168.2.208" service name="https" reject"</code>
注：如果是 public 的话是直接丢弃，会返回 timeout（连接超时）

允许 192.168.2.0/24 网段的主机访问 22 端口

<code>firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" source address=
192.168.2.0/24" port protocol="tcp" port="22" accept"</code>

允许新的 ipv4 和 ipv6 连接 ftp，并使用日志和审核，每分钟允许访问一次

<code>firewall-cmd --add-rich-rule="rule service name=ftp log limit value="1/m" audit
ccept"</code>

拒绝来自 192.168.2.0/24 网段的连接，10 秒后自动取消

<code>firewall-cmd --add-rich-rule="rule family=ipv4 source address=192.168.2.0/24 re

ect" --timeout=10</code>

将来自 192.168.2.0/24 网段访问本机 80 端口的流量转发到本机的 22 端口

<code>firewall-cmd --zone=public --add-rich-rule="rule family=ipv4 source address=12.168.2.0/24 forward-port port=80 protocol=tcp to-port=22"</code>

将来自 192.168.2.0/24 网段访问本地 80 端口的流量转发到 192.168.2.208 主机的 22 端口

<code>firewall-cmd --zone=public --add-rich-rule="rule family=ipv4 source address=12.168.2.0/24 forward-port port=80 protocol=tcp to-port=22 to-addr=192.168.2.208"</code>

伪装，将来自局域网 192.168.2.0/24 网段访问外网的流量映射为网络出口公网 IP，即修改源 IP 址

<pre><code class="highlight-chroma">firewall-cmd --zone=public --add-masquerade

firewall-cmd --zone=public --add-rich-rule="rule family=ipv4 source address=192.168.2.0/24 masquerade"

</code></pre>