A+

iptables详解（3）：iptables规则管理

在本博客中，从理论到实践，系统的介绍了iptables，如果你想要从头开始了解iptables，可以查看iptables文章列表，直达链接如下

iptables零基础快速入门系列

上一篇文章中，我们已经学会了怎样使用iptables命令查看规则，那么这篇文章我们就来总结一下，怎样管理规则。

之前，我们把查看iptables规则的操作比作"增删改查"当中的"查"，那么在这篇文章中，我们就聊聊怎样对iptables进行"增、删、改"操作。

注意：在参照本文进行iptables实验时，请务必在个人的测试机上进行，因为如果iptables规则设置不当，有可能使你无法连接到远程主机中。

首先，我们来回顾一下什么是iptables的规则。

之前打过一个比方，每条"链"都是一个"关卡"，每个通过这个"关卡"的报文都要匹配这个关卡上的规则，如果匹配，则对报文进行对应的处理，比如说，你我二人此个"报文"，你我二人此刻都要入关，可是城主有命，只有器宇轩昂之人才能入关，不符合此条件的人不能入关，于是守关将士按照城主制定的"规则"，开始打量你我你顺利入关了，而我已被拒之门外，因为你符合"器宇轩昂"的标准，所以把你"放行"了，而我不符合标准，所以没有被放行，其实，"器宇轩昂"就是一种"匹配条件"一种"动作"，"匹配条件"与"动作"组成了规则。

只不过，在iptables的世界中，最常用的匹配条件并不是"器宇轩昂"，而是报文的"源地址"、"目标地址"、"源端口"、"目标端口"等，在iptables的世界中，最常用的PT（接受）、DROP（丢弃）、REJECT（拒绝），其中ACCEPT就与我们举例中的"放行"类似，但是，我们刚才提到的这些并不是全部的匹配条件与动作，只是最常了，具体的匹配条件与动作不是我们今天讨论的重点，我们会在以后的文章中再做总结。

好了，我们已经回顾了规则的概念，并且已经明白了，规则大致由两个逻辑单元组成，匹配条件与动作，那么多说无益，我们来动手定义一条规则，此处仍然以filter链为例，因为filter表负责"过滤"功能，而所有发往本机的报文如果需要被过滤，首先会经过INPUT链（PREROUTING链没有过滤功能），这与我们所比喻的"入关"相似，所以，使用filter表的INPUT链为例，有助于我们进行理解。

首先，查看一下filter表中的INPUT链中的规则，查看规则的相关命令在前文已经总结了，此处不再赘述，如果你忘了，请回顾前文。

使用如下命令查看filter表INPUT链的规则，下图中的规则为centos6默认添加的规则。

```
[www.zsythink.net]#iptables -nL INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited
[www.zsythink.net]#
```

zsythink.net 朱双印博客

注意：在参照本文进行iptables实验时，请务必在个人的测试机上进行。

为了准备一个从零开始的环境，我们将centos6默认提供的规则清空，以便我们进行实验，使用iptables -F INPUT命令清空filter表INPUT链中的规则，后面我们会则的相关命令进行总结，此处不用纠结此命令。

清空INPUT链以后，filter表中的INPUT链已经不存在任何的规则，但是可以看出，INPUT链的默认策略是ACCEPT，也就是说，INPUT链默认"放行"所有发往本机的任何规则时，会接受所有报文，当报文没有被任何规则匹配到时，也会默认放行报文。

那么此刻，我们就在另外一台机器上，使用ping命令，向当前机器发送报文，如下图所示，ping命令可以得到回应，证明ping命令发送的报文已经正常的发送到了主机，ping命令所在机器IP地址为146，当前测试防火墙主机的IP地址为156，我们就用这样的环境，对iptables进行操作演示。

增加规则

那么此处，我们就在156上配置一条规则，拒绝192.168.1.146上的所有报文访问当前机器，之前一直在说，规则由匹配条件与动作组成，那么"拒绝192.168.1.146访问当前机器"这条规则中，报文的"源地址为192.168.1.146"则属于匹配条件，如果报文来自"192.168.1.146"，则表示满足匹配条件，而"拒绝"这个报文，就属于好了，那么怎样用命令去定义这条规则呢？使用如下命令即可

上图中，使用 -t选项指定了要操作的表，此处指定了操作filter表，与之前的查看命令一样，不使用-t选项指定表时，默认为操作filter表。

使用-I选项，指明将"规则"插入至哪个链中，-I表示insert，即插入的意思，所以-I INPUT表示将规则插入于INPUT链中，即添加规则之意。

使用-s选项，指明"匹配条件"中的"源地址"，即如果报文的源地址属于-s对应的地址，那么报文则满足匹配条件，-s为source之意，表示源地址。

使用-j选项，指明当"匹配条件"被满足时，所对应的动作，上例中指定的动作为DROP，在上例中，当报文的源地址为192.168.1.146时，报文则被DROP（丢弃）。

再次查看filter表中的INPUT链，发现规则已经被添加了，在iptables中，动作被称之为"target"，所以，上图中taget字段对应的动作为DROP。

那么此时，我们再通过192.168.1.146去ping主机156，看看能否ping通。

如上图所示，ping 156主机时，PING命令一直没有得到回应，看来我们的iptables规则已经生效了，ping发送的报文压根没有被156主机接受，而是被丢弃了，所么回应了，好了，我们已经成功的配置了一条iptables规则，看来，我们已经入门了。

还记得我们在前文中说过的"计数器"吗？此时，我们再次查看iptables中的规则，可以看到，已经有24个包被对应的规则匹配到，总计大小2016bytes。

此刻，我们来做一个实验。

现在INPUT链中已经存在了一条规则，它拒绝了所有来自192.168.1.146主机中的报文，如果此时，我们在这条规则之后再配置一条规则，后面这条规则规定，接受168.1.146主机中的报文，那么，iptables是否会接受来自146主机的报文呢？我们动手试试。

使用如下命令在filter表的INPUT链中追加一条规则，这条规则表示接受所有来自192.168.1.146的发往本机的报文。

上图中的命令并没有使用-t选项指定filter表，我们一直在说，不使用-t选项指定表时表示默认操作filter表。

上图中，使用-A选项，表示在对应的链中"追加规则"，-A为append之意，所以，-A INPUT则表示在INPUT链中追加规则，而之前示例中使用的-I选项则表示在链中聪明如你一定明白了，它们的本意都是添加一条规则，只是-A表示在链的尾部追加规则，-I表示在链的首部插入规则而已。

使用-j选项，指定当前规则对应的动作为ACCEPT。

执行完添加规则的命令后，再次查看INPUT链，发现规则已经成功"追加"至INPUT链的末尾，那么现在，第一条规则指明了丢弃所有来自192.168.1.146的报文，第了接受所有来自192.168.1.146的报文，那么结果到底是怎样的呢？实践出真知，在146主机上再次使用ping命令向156主机发送报文，发现仍然是ping不通的，看并没有生效。

而且从上图中第二条规则的计数器可以看到，根本没有任何报文被第二条规则匹配到。

聪明如你一定在猜想，发生上述情况，会不会与规则的先后顺序有关呢？测试一下不就知道了，我们再添加一条规则，新规则仍然规定接受所有来自192.168.1.146文，只是这一次，我们将新规则添加至INPUT链的最前面试试。

在添加这条规则之前，我们先把146上的ping命令强制停止了，然后使用如下命令，在filter表的INPUT链的前端添加新规则。

好了，现在第一条规则就是接受所有来自192.168.1.146的报文，而且此时计数是0，此刻，我们再从146上向156发起ping请求。

146上已经可以正常的收到响应报文了，那么回到156查看INPUT链的规则，第一条规则的计数器已经显示出了匹配到的报文数量。

看来，规则的顺序很重要。

如果报文已经被前面的规则匹配到，iptables则会对报文执行对应的动作，即使后面的规则也能匹配到当前报文，很有可能也没有机会再对报文执行相应的动作了，例，报文先被第一条规则匹配到了，于是当前报文被"放行"了，因为报文已经被放行了，所以，即使上图中的第二条规则即使能够匹配到刚才"放行"的报文，也没有的报文进行丢弃操作了。这就是iptables的工作机制。

之前在总结查看命令时提到过，使用--line-number选项可以列出规则的序号，如下图所示

我们也可以在添加规则时，指定新增规则的编号，这样我们就能在任意位置插入规则了，我们只要把刚才的命令稍作修改即可，如下。

仍然使用-I选项进行插入规则操作，-I INPUT 2表示在INPUT链中新增规则，新增的规则的编号为2，好了，自己动手试试吧。

删除规则

注意：在参照本文进行iptables实验时，请务必在个人的测试机上进行。

此刻，如果我们想要删除filter表中INPUT中的一条规则，该怎么做呢？

有两种办法

方法一：根据规则的编号去删除规则

方法二：根据具体的匹配条件与动作删除规则

那么我们先看看方法一，先查看一下filter表中INPUT链中的规则

假如我们想要删除上图中的第3条规则，则可以使用如下命令。

上例中，使用了-t选项指定了要操作的表（没错，省略-t默认表示操作filter表），使用-D选项表示删除指定链中的某条规则，-D INPUT 3表示删除INPUT链中的第3

当然，我们也可以根据具体的匹配条件与动作去删除规则，比如，删除下图中源地址为192.168.1.146，动作为ACCEPT的规则，于是，删除规则的命令如下。

上图中，删除对应规则时，仍然使用-D选项，-D INPUT表示删除INPUT链中的规则，剩下的选项与我们添加规则时一毛一样，-s表示以对应的源地址作为匹配条件示对应的动作为接受，所以，上述命令表示删除INPUT链中源地址为192.168.1.146，动作为ACCEPT的规则。

而删除指定表中某条链中的所有规则的命令，我们在一开始就使用到了，就是"iptables -t 表名 -F 链名"

-F选项为flush之意，即冲刷指定的链，即删除指定链中的所有规则，但是注意，此操作相当于删除操作，在没有保存iptables规则的情况下，请慎用。

其实，-F选项不仅仅能清空指定链上的规则，其实它还能清空整个表中所有链上的规则，不指定链名，只指定表名即可删除表中的所有规则，命令如下

iptables -t 表名 -F

不过再次强调，在没有保存iptables规则时，请勿随便清空链或者表中的规则，除非你明白你在干什么。

修改规则

注意：在参照本文进行iptables实验时，请务必在个人的测试机上进行。

那么，我们怎样修改某条规则中的动作呢？比如，我想把如下规则中的动作从DROP改为REJECT，改怎么办呢？

我们可以使用-R选项修改指定的链中的规则，在修改规则时指定规则对应的编号即可(有坑，慎行)，示例命令如下

上例中，-R选项表示修改指定的链，使用-R INPUT 1表示修改INPUT链的第1条规则，使用-j REJECT表示将INPUT链中的第一条规则的动作修改为REJECT，注意：项以及对应的源地址不可省略，即使我们已经指定了规则对应的编号，但是在使用-R选项修改某个规则时，必须指定规则对应的原本的匹配条件（如果有多个匹配条定）。

如果上例中的命令没有使用-s指定对应规则中原本的源地址，那么在修改完成后，你修改的规则中的源地址会自动变为0.0.0.0/0（此IP表示匹配所有网段的IP地址）对应的动作又为REJECT，所以在执行上述命令时如果没有指明规则原本的源地址，那么所有IP的请求都被拒绝了（因为没有指定原本的源地址，当前规则的源地址自0/0），如果你正在使用ssh远程到服务器上进行iptables设置，那么你的ssh请求也将会被阻断。

既然使用-R选项修改规则时，必须指明规则原本的匹配条件，那么我们则可以理解为，只能通过-R选项修改规则对应的动作了，所以我觉得，如果你想要修改某条规将这条规则删除，然后在同样位置再插入一条新规则，这样更好，当然，如果你只是为了修改某条规则的动作，那么使用-R选项时，不要忘了指明规则原本对应的匹

好了，上例中，我们已经将规则中的动作从DROP改为了REJECT，那么DROP与REJECT有什么不同呢？从字面上理解，DROP表示丢弃，REJECT表示拒绝，REJECT像更坚决一点，我们再次从146主机上向156主机上发起ping请求，看看与之前动作为DROP时有什么不同。

如上图所示，当156主机中的iptables规则对应的动作为REJECT时，从146上进行ping操作时，直接就提示"目标不可达"，并没有像之前那样卡在那里，看来，REJ加"干脆"。

其实，我们还可以修改指定链的"默认策略"，没错，就是下图中标注的默认策略。

每张表的每条链中，都有自己的默认策略，我们也可以理解为默认"动作"。

当报文没有被链中的任何规则匹配到时，或者，当链中没有任何规则时，防火墙会按照默认动作处理报文，我们可以修改指定链的默认策略，使用如下命令即可。

使用-t指定要操作的表，使用-P选项指定要修改的链，上例中，-P FORWARD DROP表示将表中FORWRD链的默认策略改为DROP。

保存规则

在默认的情况下，我们对"防火墙"所做出的修改都是"临时的"，换句话说就是，当重启iptables服务或者重启服务器以后，我们平常添加的规则或者对规则所做出的失，为了防止这种情况的发生，我们需要将规则"保存"。

centos7与centos6中的情况稍微有些不同，我们先说centos6中怎样保存iptables规则。

centos6中，使用"service iptables save"命令即可保存规则，规则默认保存在/etc/sysconfig/iptables文件中，如果你刚刚安装完centos6，在刚开始使用iptabl
lter表中会有一些默认的规则，这些默认提供的规则其实就保存在/etc/sysconfig/iptables中， 保存规则的示例如下。

如上图所示，文件中保存了filter表中每条链的默认策略，以及每条链中的规则，由于其他表中并没有设置规则，也没有使用过其他表，所以文件中只保存了filter表

当我们对规则进行了修改以后，如果想要修改永久生效，必须使用service iptables save保存规则，当然，如果你误操作了规则，但是并没有保存，那么使用servic
tart命令重启iptables以后，规则会再次回到上次保存/etc/sysconfig/iptables文件时的模样。

从现在开始，最好养成及时保存规则的好习惯。

centos7中，已经不再使用init风格的脚本启动服务，而是使用unit文件，所以，在centos7中已经不能再使用类似service iptables start这样的命令了，所以servic
ve也无法执行，同时，在centos7中，使用firewall替代了原来的iptables service，不过不用担心，我们只要通过yum源安装iptables与iptables-services即可（ip
被默认安装，但是iptables-services在centos7中一般不会被默认安装），在centos7中安装完iptables-services后，即可像centos6中一样，通过service iptable
存规则了，规则同样保存在/etc/sysconfig/iptables文件中。

此处给出centos7中配置iptables-service的步骤

```
#配置好yum源以后安装iptables-service
# yum install -y iptables-services
#停止firewalld
# systemctl stop firewalld
#禁止firewalld自动启动
# systemctl disable firewalld
#启动iptables
# systemctl start iptables
#将iptables设置为开机自动启动，以后即可通过iptables-service控制iptables服务
# systemctl enable iptables
```

上述配置过程只需一次，以后即可在centos7中愉快的使用service iptables save命令保存iptables规则了。

其他通用方法

还可以使用另一种方法保存iptables规则，就是使用iptables-save命令

使用iptables-save并不能保存当前的iptables规则，但是可以将当前的iptables规则以"保存后的格式"输出到屏幕上。

所以，我们可以使用iptables-save命令，再配合重定向，将规则重定向到/etc/sysconfig/iptables文件中即可。

iptables-save > /etc/sysconfig/iptables

我们也可以将/etc/sysconfig/iptables中的规则重新载入为当前的iptables规则，但是注意，未保存入/etc/sysconfig/iptables文件中的修改将会丢失或者被覆盖。

使用iptables-restore命令可以从指定文件中重载规则，示例如下

iptables-restore < /etc/sysconfig/iptables

再次提醒：重载规则时，现有规则将会被覆盖。

命令小结

上文已经详细的举例并描述了怎样进行iptables规则管理，为了以后能够快速的回顾，我们把上述命令总结一下。

添加规则

注意点：添加规则时，规则的顺序非常重要

在指定表的指定链的尾部添加一条规则，-A选项表示在对应链的末尾添加规则，省略-t选项时，表示默认操作filter表中的规则

```
命令语法：iptables -t 表名 -A 链名 匹配条件 -j 动作
```