iptables详解（10）：iptables自定义链

朱双印
2年前更新

> 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

前文中，我们一直在定义规则，准确的说，我们一直在iptables的**默认链**中定义规则，那么此处，我们就来了解一下**自定义链**。

你可能会问，iptables的默认链就已经能够满足我们了，为什么还需要自定义链呢？

原因如下：

当默认链中的规则非常多时，不方便我们管理。

想象一下，如果INPUT链中存放了200条规则，这200条规则有针对httpd服务的，有针对sshd服务的，有针对私网IP的，有针对公网IP的，假如，我们突然想要修改针对httpd服务的相关规则，难道我们还要从头看一遍这200条规则，找出哪些规则是针对httpd的吗？这显然不合理。

所以，iptables中，可以自定义链，通过自定义链即可解决上述问题。

假设，我们自定义一条链，链名叫IN_WEB，我们可以将所有针对80端口的入站规则都写入到这条自定义链中，当以后想要修改针对web服务的入站规则时，就直接修改IN_WEB链中的规则就好了，即使默认链中有再多的规则，我们也不会害怕了，因为我们知道，所有针对80端口的入站规则都存放在IN_WEB链中，同理，我们可以将针对sshd的出站规则放入到OUT_SSH自定义链中，将针对Nginx的入站规则放入到IN_NGINX自定义链中，这样，我们就能想改哪里改哪里，再也不同担心找不到规则在哪里了。

但是需要注意的是，自定义链并不能直接使用，而是需要被默认链引用才能够使用，空口白话说不明白，等到示例时我们自然会明白。

说了这么多，我们来动手创建一条自定义链，使用-N选项可以创建自定义链，示例如下

```
[www.zsythink.net]#iptables -F
[www.zsythink.net]#iptables -t filter -N IN_WEB
[www.zsythink.net]#iptables -nvL
Chain INPUT (policy ACCEPT 25 packets, 2455 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 10 packets, 1104 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain IN_WEB (0 references)
 pkts bytes target     prot opt in     out     source               destination
[www.zsythink.net]#
```


如上图所示，"-t filter"表示操作的表为filter表，与之前的示例相同，省略-t选项时，缺省操作的就是filter表。

"-N IN_WEB"表示创建一个自定义链，自定义链的名称为"IN_WEB"

自定义链创建完成后，查看filter表中的链，如上图所示，自定义链已经被创建，而且可以看到，这条自定义链的引用计数为0 (0 references)，也就是说，这条自定义链还没有被任何默认链所引用，所以，即使IN_WEB中配置了规则，也不会生效，我们现在不用在意它，继续聊聊我们的自定义链。

好了，自定义链已经创建完毕，现在我们就可以直接在自定义链中配置规则了，如下图所示，我们配置一些规则用于举例。

```
[www.zsythink.net]# iptables -t filter -I IN_WEB -s 192.168.1.139 -j REJECT
[www.zsythink.net]# iptables -I IN_WEB -s 192.168.1.188 -j REJECT
[www.zsythink.net]#
[www.zsythink.net]# iptables -t filter --line -nvL IN_WEB
Chain IN_WEB (0 references)
num   pkts bytes target     prot opt in     out     source               desti
1        0     0 REJECT     all  --  *      *       192.168.1.188        0.0.0
2        0     0 REJECT     all  --  *      *       192.168.1.139        0.0.0
[www.zsythink.net]#
```


如上图所示，对自定义链的操作与对默认链的操作并没有什么不同，一切按照操作默认链的方法操作自定义链即可。

现在，自定义链中已经有了一些规则，但是目前，这些规则无法匹配到任何报文，因为我们并没有在任何默认链中引用它。

既然IN_WEB链是为了针对web服务的入站规则而创建的，那么这些规则应该去匹配入站的报文，所以，我们应该用INPUT链去引用它。

当然，自定义链在哪里创建，应该被哪条默认链引用，取决于实际的工作场景，因为此处示例的规则是匹配入站报文，所以在INPUT链中引用自定义链。

示例如下。

```
[www.zsythink.net]# iptables -I INPUT -p tcp --dport 80 -j IN_WEB
[www.zsythink.net]#
[www.zsythink.net]# iptables -nvL
Chain INPUT (policy ACCEPT 34 packets, 2540 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 IN_WEB     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 19 packets, 2028 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain IN_WEB (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 REJECT     all  --  *      *       192.168.1.188        0.0.0.0/0           reject-with
    0     0 REJECT     all  --  *      *       192.168.1.139        0.0.0.0/0
```


上图中，我们在INPUT链中添加了一条规则，访问本机80端口的tcp报文将会被这条规则匹配到

而上述规则中的"-j IN_WEB"表示：访问80端口的tcp报文将由自定义链"IN_WEB"中的规则进行处理。没错，在之前的示例中，我们使用"-j"选项指定动作，而此处，我们将"动作"替换为了"自定义链"，当"-j"对应的值为一个自定义链时，就表示被当前规则匹配到的报文将交由对应的自定义链处理，具体怎样处理，取决于自定义链中的规则。当IN_WEB自定义链被INPUT链引用以后，可以发现，IN_WEB链的引用计数已经变为1，表示这条自定义链已经被引用了1次，自定义链还可以引用其他的自定义链，感兴趣的话，动手试试吧。

在之前的文章中，我们说过，"动作"在iptables中被称为"target"，这样描述并不准确，因为target为目标之意，报文被规则匹配到以后，target能是一个"动作"，target也能是一个"自定义链"，当target为一个动作时，表示报文按照指定的动作处理，当target为自定义链时，表示报文由自定义链中的规则处理，现在回过头再理解之前的术语，似乎更加明了了。

那么此刻，我们在192.168.1.139上尝试访问本机的80端口，已经被拒绝访问，证明刚才自定义链中的规则已经生效了。

过了一段时间，我们发现IN_WEB这个名字不太合适，我们想要将这条自定义链重命名，把名字改成WEB，可以吗？必须能啊，示例如下

```
[www.zsythink.net]# iptables -E IN_WEB WEB
[www.zsythink.net]# iptables -nvL
Chain INPUT (policy ACCEPT 27 packets, 2036
 pkts bytes target     prot opt in     out
    2   120 WEB        tcp  --  *      *

Chain FORWARD (policy ACCEPT 0 packets, 0 by
 pkts bytes target     prot opt in     out

Chain OUTPUT (policy ACCEPT 15 packets, 1420
 pkts bytes target     prot opt in     out

Chain WEB (1 references)
 pkts bytes target     prot opt in     out
    0     0 REJECT     all  --  *      *
    2   120 REJECT     all  --  *      *
[www.zsythink.net]#
```


如上图所示，使用"-E"选项可以修改自定义链名，如上图所示，引用自定义链处的名称会自动发生改变。

好了，我们已经能够创建自定义了，那么怎样删除自定义链呢？

使用"-X"选项可以删除自定义链，但是删除自定义链时，需要满足两个条件：

1、自定义链没有被任何默认链引用，即自定义链的引用计数为0。

2、自定义链中没有任何规则，即自定义链为空。

那么，我们来删除自定义链WEB试试。

```
[www.zsythink.net]# iptables -X WEB
iptables: Too many links.
```

如上图所示，使用"-X"选项删除对应的自定义链，但是上例中，并没有成功删除自定义链WEB，提示：Too many links，是因为WEB链已经被默认链所引用，不满足上述条件1，所以，我们需要删除对应的引用规则，示例如下。

```
[www.zsythink.net]# iptables -nvL INPUT
Chain INPUT (policy ACCEPT 468 packets, 5665
 pkts bytes target     prot opt in     out
    2   120 WEB        tcp  --  *      *
[www.zsythink.net]# iptables -D INPUT 1
[www.zsythink.net]# iptables -X WEB
iptables: Directory not
```


如上图所示，删除引用自定义链的规则后，再次尝试删除自定义链，提示：Directory not empty，是因为WEB链中存在规则，不满足上述条件2，所以，我们需要清空对应的自定义链，示例如下

```
[www.zsythink.net]# iptables -t filter -F WEB
[www.zsythink.net]# iptables -t filter -X WEB
[www.zsythink.net]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[www.zsythink.net]#
```


如上图所示，使用"-X"选项可以删除一个引用计数为0的、空的自定义链。

小结

为了方便以后回顾，我们将上述命令进行总结，

创建自定义链

```
#示例：在filter表中创建IN_WEB自定义链
iptables -t filter -N IN_WEB
```

引用自定义链

```
#示例：在INPUT链中引用刚才创建的自定义链
iptables -t filter -I INPUT -p tcp --dport 80 -j IN_WEB
```

重命名自定义链

```
#示例：将IN_WEB自定义链重命名为WEB
iptables -E IN_WEB WEB
```

删除自定义链

删除自定义链需要满足两个条件

1、自定义链没有被引用

2、自定义链中没有任何规则

```
#示例：删除引用计数为0并且不包含任何规则的WEB链
iptables -X WEB
```

好了，自定义链就总结到这里，希望这篇文章能够对你有所帮助~~~各位客官，再见啦，么么哒~~~~