iptables详解（7）：iptables扩展之udp扩展与icmp扩展

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

前文总结了iptables的tcp扩展模块，此处，我们来总结一下另外两个跟协议有关的常用的扩展模块，udp扩展与icmp扩展。

udp扩展

我们先来说说udp扩展模块，这个扩展模块中能用的匹配条件比较少，只有两个，就是–sport与–dport，即匹配报文的源端口与目标端口。

没错，tcp模块中也有这两个选项，名称都一模一样。

只不过udp扩展模块的–sport与–dport是用于匹配UDP协议报文的源端口与目标端口，比如，放行samba服务的137与138这两个UDP端口，示例如下

```
#iptables -t filter -I INPUT -p udp -m udp --dport 137 -j ACCEPT
#iptables -t filter -I INPUT -p udp -m udp --dport 138 -j ACCEPT
```

前文说明过，当使用扩展匹配条件时，如果未指定扩展模块，iptables会默认调用与"-p"对应的协议名称相同的模块，所以，当使用"-p udp"时，可以省略"-m udp"，示例如下。

```
#iptables -t filter -I INPUT -p udp --dport 137 -j ACCEPT
#iptables -t filter -I INPUT -p udp --dport 138 -j ACCEPT
```

udp扩展中的–sport与–dport同样支持指定一个连续的端口范围，示例如下

```
#iptables -t filter -I INPUT -p udp --dport 137:157 -j ACCEPT
#
```

上图中的配置表示137到157之间的所有udp端口全部对外开放，其实与tcp扩展中的使用方法相同。

但是udp中的–sport与–dport也只能指定连续的端口范围，并不能一次性指定多个离散的端口，没错，聪明如你一定想到，使用之前总结过的multiport扩展模块，即可指定多个离散的UDP端口，如果你忘了multiport模块怎样使用，请回顾前文。

总之有了前文的基础，再理解上述示例就容易多了，此处不再对udp模块的–sport与–dport进行赘述。

icmp扩展

最常用的tcp扩展、udp扩展已经总结完毕，现在聊聊icmp扩展，没错，看到icmp，你肯定就想到了ping命令，因为ping命令使用的就是icmp协议。

ICMP协议的全称为Internet Control Message Protocol，翻译为互联网控制报文协议，它主要用于探测网络上的主机是否可用，目标是否可达，网络是否通畅，路由是否可用等。

我们平常使用ping命令ping某主机时，如果主机可达，对应主机会对我们的ping请求做出回应（此处不考虑禁ping等情况），也就是说，我们发出ping请求，对方回应ping请求，虽然ping请求报文与ping回应报文都属于ICMP类型的报文，但是如果在概念上细分的话，它们所属的类型还是不同的，我们发出的ping请求属于类型8的icmp报文，而对方主机的ping回应报文则属于类型0的icmp报文，根据应用场景的不同，icmp报文被细分为如下各种类型。

type	code	Description	Query	Error
0	0	echo reply (Ping reply, Chapter 7)	•	
3		destination unreachable:		
	0	network unreachable (Section 9.3)		•
	1	host unreachable (Section 9.3)		•
	2	protocol unreachable		•
	3	port unreachable (Section 6.5)		•
	4	fragmentation needed but don't-fragment bit set (Section 11.6)		•
	5	source route failed (Section 8.5)		•
	6	destination network unknown		•
	7	destination host unknown		•
	8	source host isolated (obsolete)		•
	9	destination network administratively prohibited		•
	10	destination host administratively prohibited		•
	11	network unreachable for TOS (Section 9.3)		•
	12	host unreachable for TOS (Section 9.3)		•
	13	communication administratively prohibited by filtering		•
	14	host precedence violation		•
	15	precedence cutoff in effect		•
4	0	source quench (elementary flow control, Section 11.11)		•
5		redirect (Section 9.5):		
	0	redirect for network		•
	1	redirect for host		•
	2	redirect for type-of-service and network		•
	3	redirect for type-of-service and host		•
8	0	echo request (Ping request, Chapter 7)	•	
9	0	router advertisement (Section 9.6)	•	
10	0	router solicitation (Section 9.6)	•	
11		time exceeded:		
	0	time-to-live equals 0 during transit (Traceroute, Chapter 8)		•
	1	time-to-live equals 0 during reassembly (Section 11.5)		•
12		parameter problem:		
	0	IP header bad (catchall error)		•
	1	required option missing		•
13	0	timestamp request (Section 6.4)	•	
14	0	timestamp reply (Section 6.4)	•	
15	0	information request (obsolete)	•	
16	0	information reply (obsolete)	•	
17	0	address mask request (Section 6.3)	•	
18	0	address mask reply (Section 6.3)	•	

从上图可以看出，所有表示"目标不可达"的icmp报文的type码为3，而"目标不可达"又可以细分为多种情况，是网络不可达呢？还是主机不可达呢？再或者是端口不可达呢？所以，为了更加细化的区分它们，icmp对每种type又细分了对应的code，用不同的code对应具体的场景， 所以，我们可以使用type/code去匹配具体类型的ICMP报文，比如可以使用"3/1"表示主机不可达的icmp报文。

上图中的第一行就表示ping回应报文，它的type为0，code也为0，从上图可以看出，ping回应报文属于查询类（query）的ICMP报文，从大类上分，ICMP报文还能分为查询类与错误类两大类，目标不可达类的icmp报文则属于错误类报文。

而我们发出的ping请求报文对应的type为8，code为0。

了解完上述概念，就好办了，我们来看一些应用场景。

假设，我们现在想要禁止所有icmp类型的报文进入本机，那么我们可以进行如下设置。

```
#iptables -t filter -I INPUT -p icmp -j REJECT
#
```

上例中，我们并没有使用任何扩展匹配条件，我们只是使用"-p icmp"匹配了所有icmp协议类型的报文。

如果进行了上述设置，别的主机向我们发送的ping请求报文无法进入防火墙，我们向别人发送的ping请求对应的回应报文也无法进入防火墙。所以，我们既无法ping通别人，别人也无法ping通我们。

假设，此刻需求有变，我们只想要ping通别人，但是不想让别人ping通我们，刚才的配置就不能满足我们了，我们则可以进行如下设置（此处不考虑禁ping的情况）

```
[www.zsythink.net]#iptables -F
[www.zsythink.net]#iptables -t filter -I INPUT -p icmp -m icmp --icmp-type 8/0 -j REJECT
[www.zsythink.net]#
[www.zsythink.net]#ping 192.168.1.146
PING 192.168.1.146 (192.168.1.146) 56(84) bytes of data.
64 bytes from 192.168.1.146: icmp_seq=1 ttl=64 time=0.279 ms
64 bytes from 192.168.1.146: icmp_seq=2 ttl=64 time=0.324 ms
```


上图中，使用"-m icmp"表示使用icmp扩展，因为上例中使用了"-p icmp"，所以"-m icmp"可以省略，使用"–icmp-type"选项表示根据具体的type与code去匹配对应的icmp报文，而上图中的"–icmp-type 8/0"表示icmp报文的type为8，code为0才会被匹配到，也就是只有ping请求类型的报文才能被匹配到，所以，别人对我们发起的ping请求将会被拒绝通过防火墙，而我们之所以能够ping通别人，是因为别人回应我们的报文的icmp type为0，code也为0，所以无法被上述规则匹配到，所以我们可以看到别人回应我们的信息。

因为type为8的类型下只有一个code为0的类型，所以我们可以省略对应的code，示例如下

```
[www.zsythink.net]#iptables -F
[www.zsythink.net]#iptables -t filter -I INPUT -p icmp --icmp-type 8 -j REJECT
[www.zsythink.net]#
[www.zsythink.net]#ping 192.168.1.146
PING 192.168.1.146 (192.168.1.146) 56(84) bytes of data.
64 bytes from 192.168.1.146: icmp_seq=1 ttl=64 time=0.414 ms
64 bytes from 192.168.1.146: icmp_seq=2 ttl=64 time=0.320 m
```


除了能够使用对应type/code匹配到具体类型的icmp报文以外，我们还能用icmp报文的描述名称去匹配对应类型的报文，示例如下

```
#iptables -F
#iptables -t filter -I INPUT -p icmp --icmp-type "echo-request" -j REJECT
```

没错，上例中使用的 –icmp-type "echo-request"与 –icmp-type 8/0的效果完全相同，参考本文最上方的表格即可获取对应的icmp类型的描述名称。

8	0	echo request (Ping request, Chapter 7)

注意：名称中的"空格"需要替换为"-"。

小结

udp扩展

常用的扩展匹配条件

–sport：匹配udp报文的源地址

–dport：匹配udp报文的目标地址

```
#示例
iptables -t filter -I INPUT -p udp -m udp --dport 137 -j ACCEPT
iptables -t filter -I INPUT -p udp -m udp --dport 137:157 -j ACCEPT
#可以结合multiport模块指定多个离散的端口
```

icmp扩展

常用的扩展匹配条件

–icmp-type：匹配icmp报文的具体类型

```
#示例
iptables -t filter -I INPUT -p icmp -m icmp --icmp-type 8/0 -j REJECT
iptables -t filter -I INPUT -p icmp --icmp-type 8 -j REJECT
iptables -t filter -I OUTPUT -p icmp -m icmp --icmp-type 0/0 -j REJECT
iptables -t filter -I OUTPUT -p icmp --icmp-type 0 -j REJECT
iptables -t filter -I INPUT -p icmp --icmp-type "echo-request" -j REJECT
```