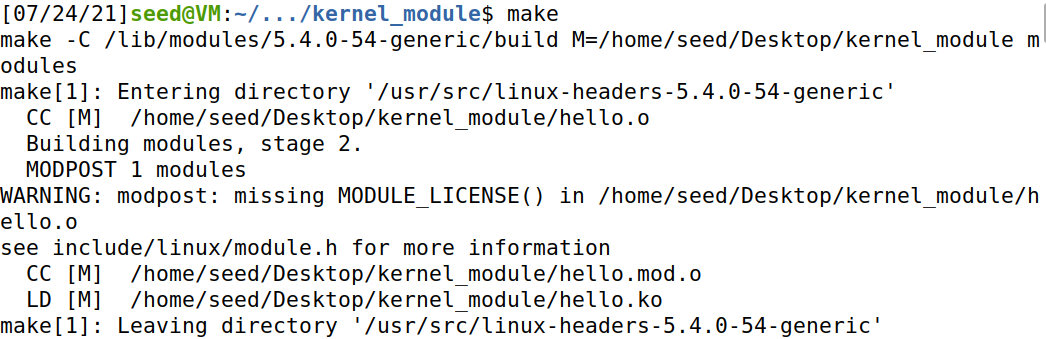
**Lab[6]-report**

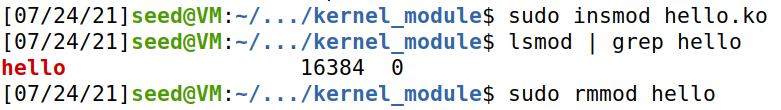
**57118120 黄逸飞**

**Task1.A**

**Make过程**

****

**Insmod，显示和rmmod**

****

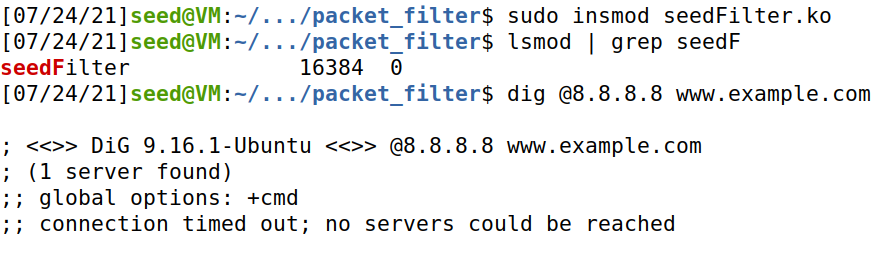
**在syslog中可以看到**

****

**Task1.B**

**1、**

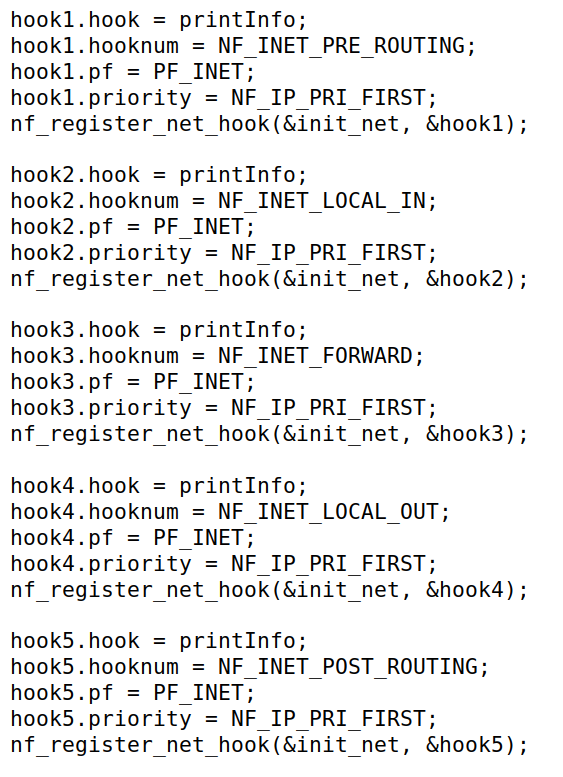
**Insmod，可以看到dig被阻拦**

****

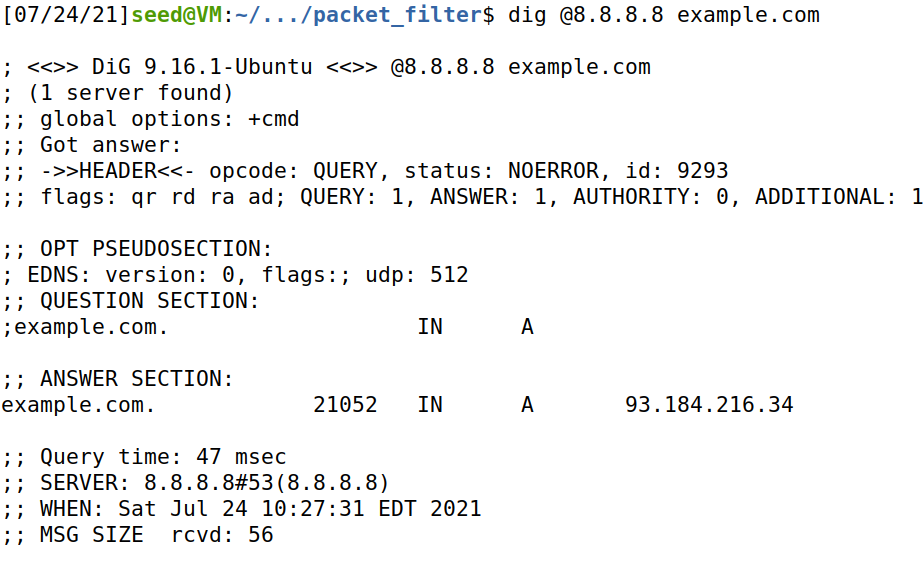
**2、**

**代码中加入的部分**

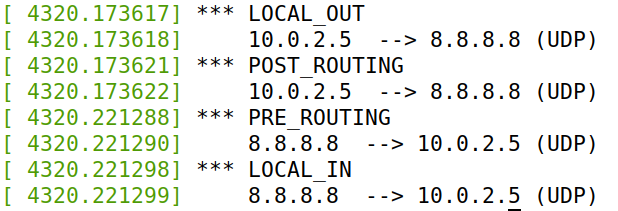
****

****

**实验如下：dig @8.8.8.8 example.com**

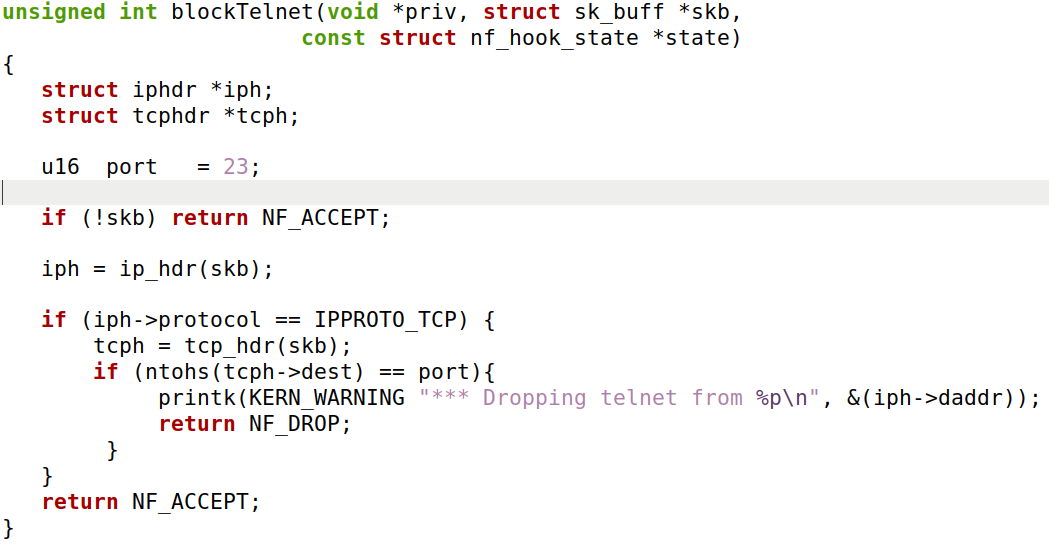
****

**在syslog中看到了如下内容，可以看到local\_out和post\_routing在发出时被顺序触发，pre\_routing和local\_in在接收时被顺序触发，而forward是转发时被触发的，在该实验中不触发**

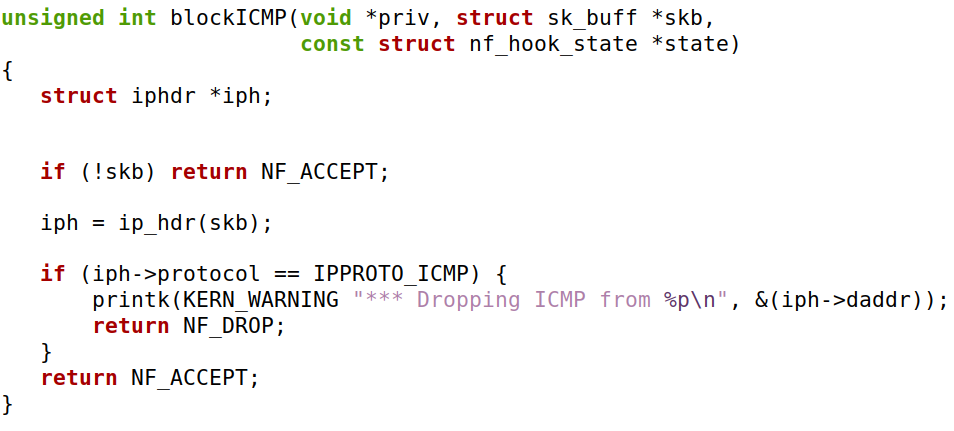
****

**3、加入的代码如下**

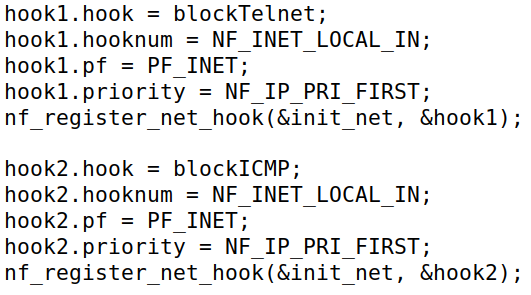
**阻拦telnet的函数**

****

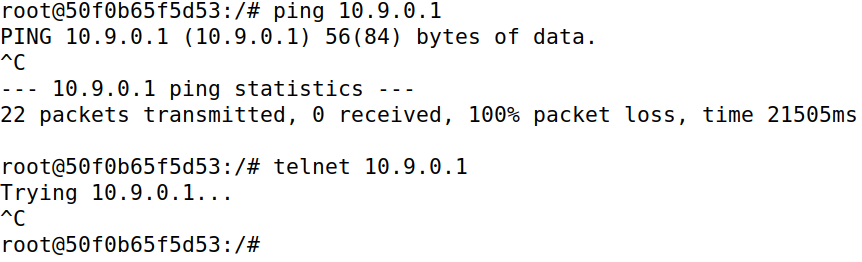
**阻拦ping的函数**

****

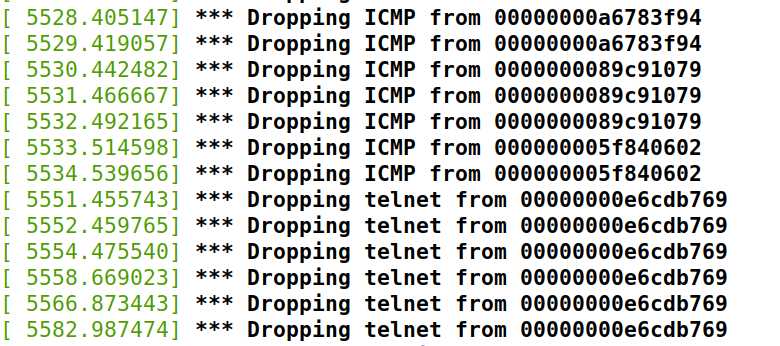
**两个hook**

****

**测试如下，无法telnet也无法ping**

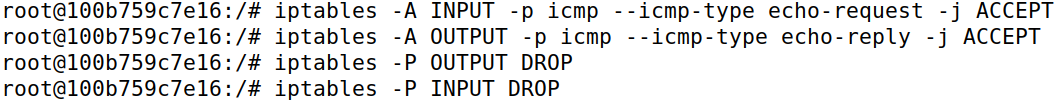
****

**在syslog中可以看到输出**

****

**Task2.A**

**Pdf的代码错误，如下为正确代码**

****

**四条规则的作用分别是：**

**iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT**

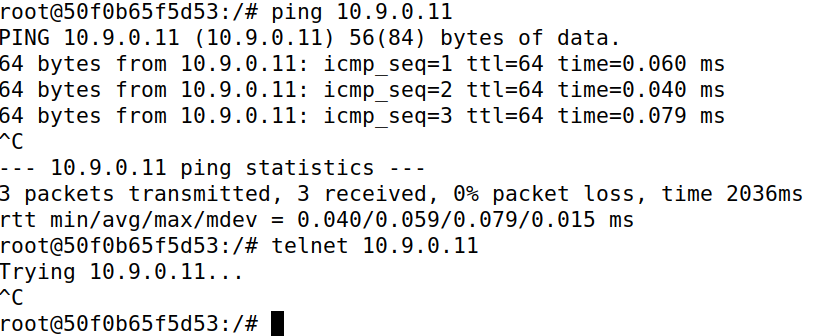
**允许接收icmp请求报文iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT**

**允许发出icmp响应报文**

**iptables -P OUTPUT DROP 丢弃所有发送报文**

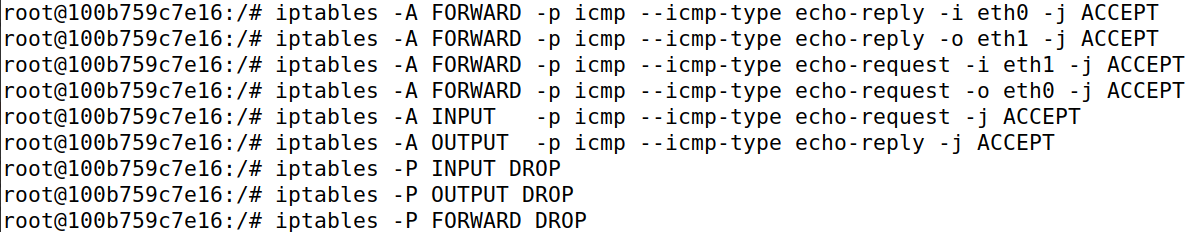
**iptables -P INPUT DROP 丢弃所有接收报文**

**防火墙效果如图所示，可以ping通，但无法telnet**

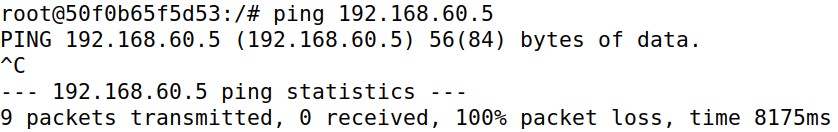
****

**Task2.B**

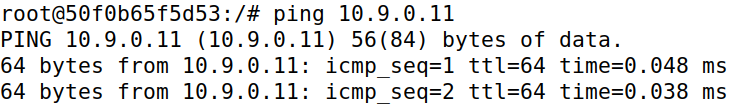
**规则如下**

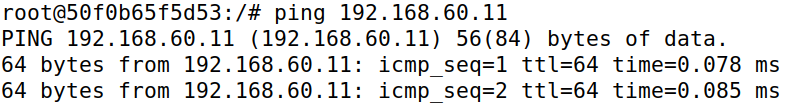
****

**10.9.0.0/24无法ping通 192.168.60.0/24**

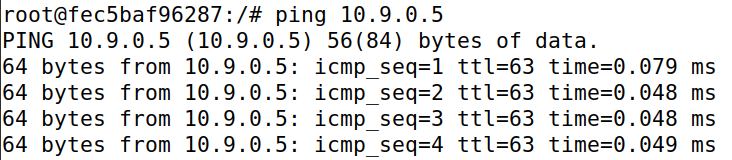
****

**10.9.0.0/24可以ping通两个网段的路由接口**

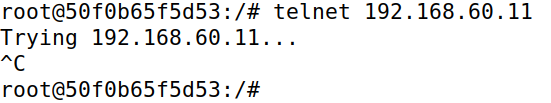
****

****

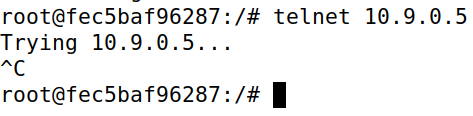
**192.168.60.0/24可以反过来ping通10.9.0.0/24**

****

**10.9.0.0/24无法telnet通 192.168.60.0/24**

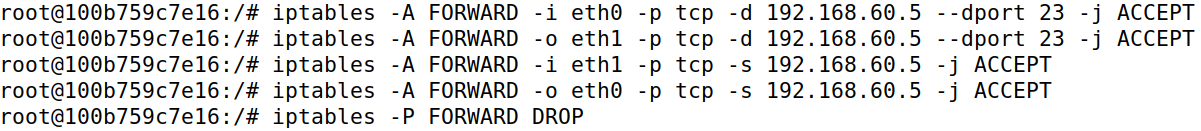
****

**反之也不行**

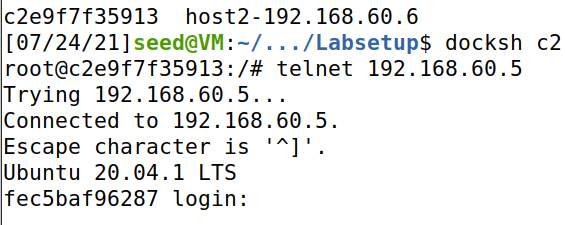
****

**Task2.C**

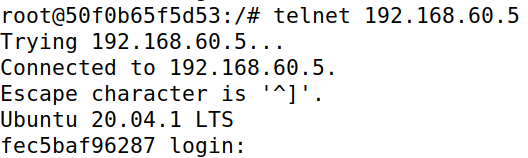
**规则如下**

****

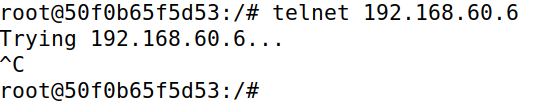
**内部主机可以相互telnet**

****

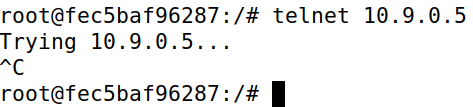
**外部可以telnet到192.168.60.5**

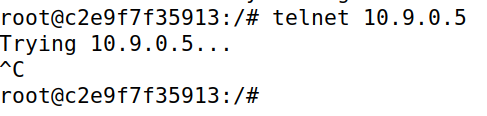
****

**外部无法telnet到除192.168.60.5的主机**

****

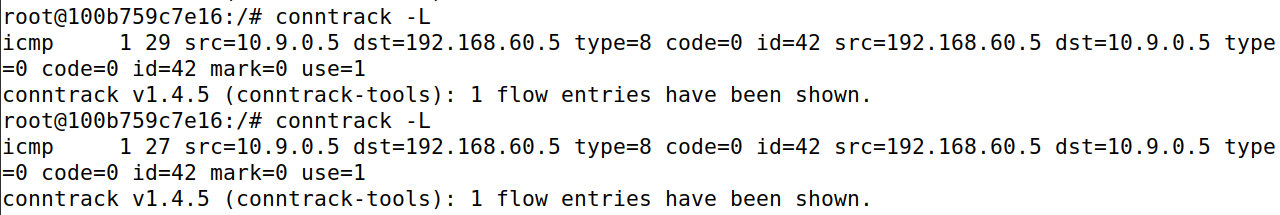
**内部的都telnet不到外面的**

****

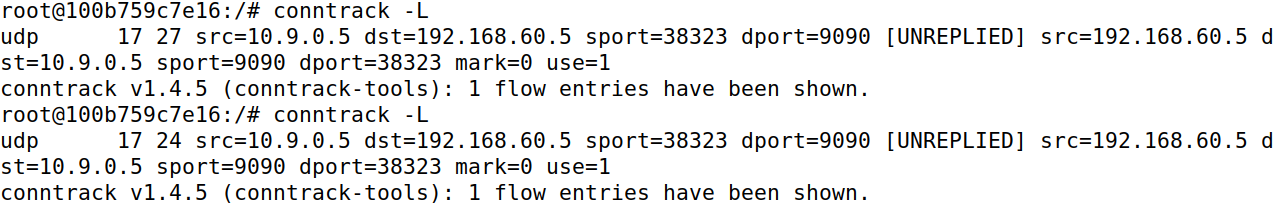
****

**Task3.A**

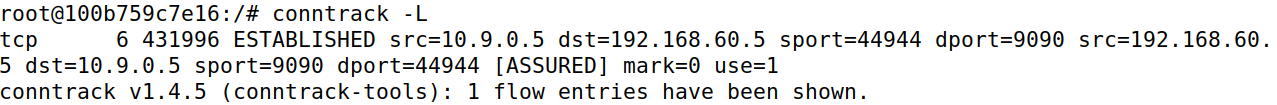
**ICMP 状态持续30秒**

****

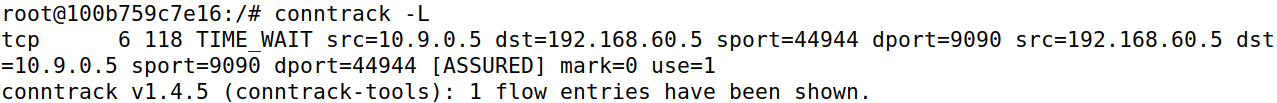
**UDP每次发送消息都有30秒的状态持续**

****

**TCP连接后有43200秒的持续时间**

****

**断后有120秒的持续时间**

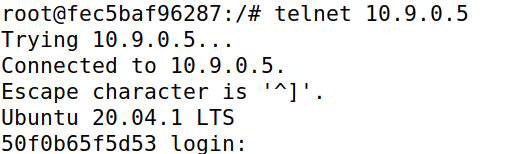
****

**Task3.B**

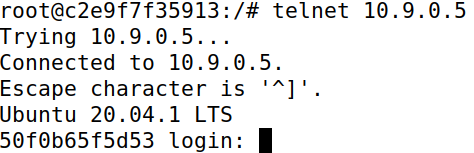
**附加的规则如下**

****

**192.168.60.5可以telnet到10.9.0.5**

****

**192.168.60.6也可以telnet到10.9.0.5**

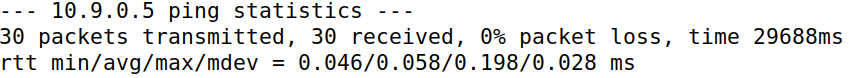
****

**Task4**

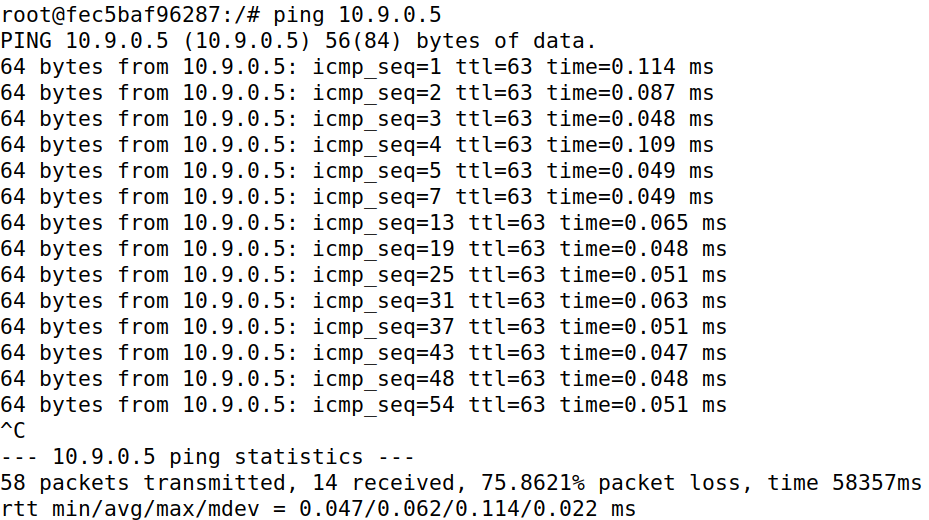
**只有一条规则时**

****

**100%received，不具有拦截作用**

****

**两条规则后ping的结果，可以看到第5条后开始出现拦截**

****

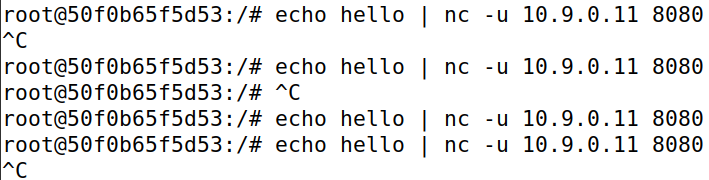
**当没有第二条规则时，第一条规则过滤的包也默认ACCEPT，只有加入第二条后才被DROP。**

**Task5**

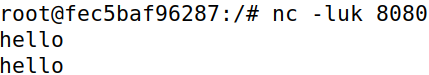
**开始代码如下**

****

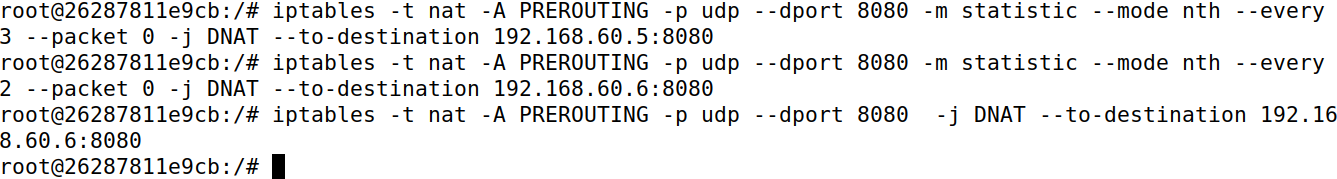
**10.9.0.5发包4个，可以看到只有第一个和第四个发出**

****

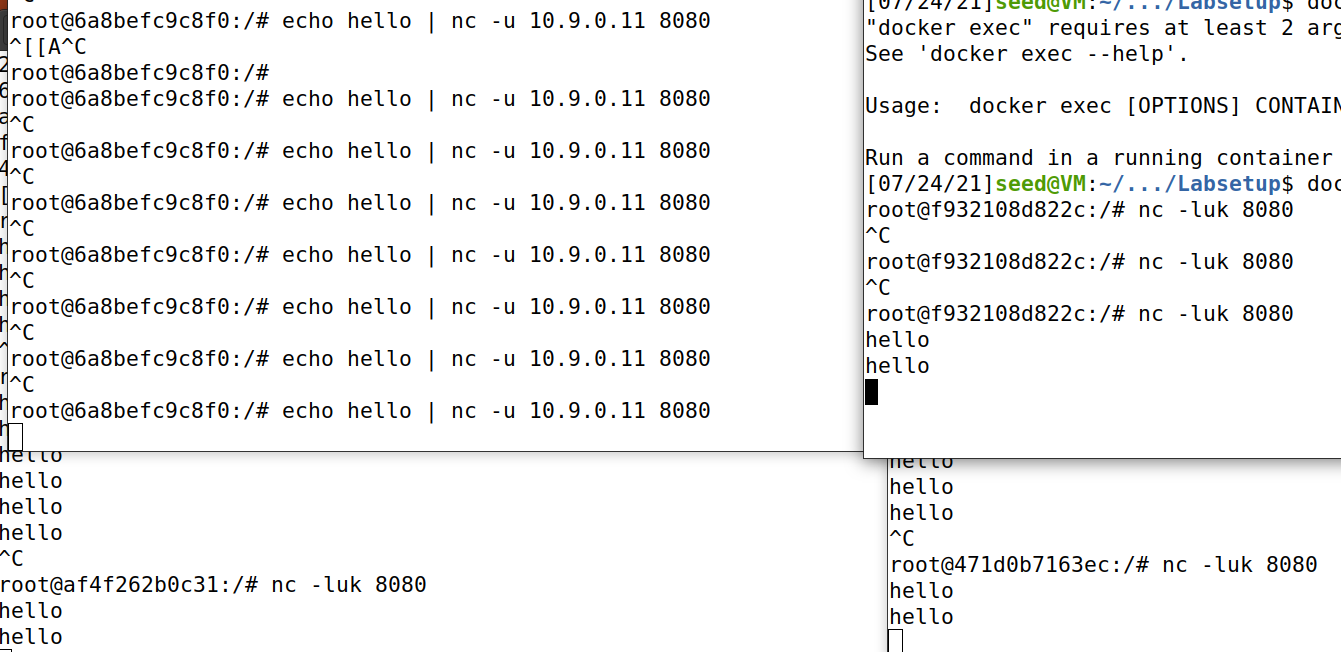
**结果符合**

****

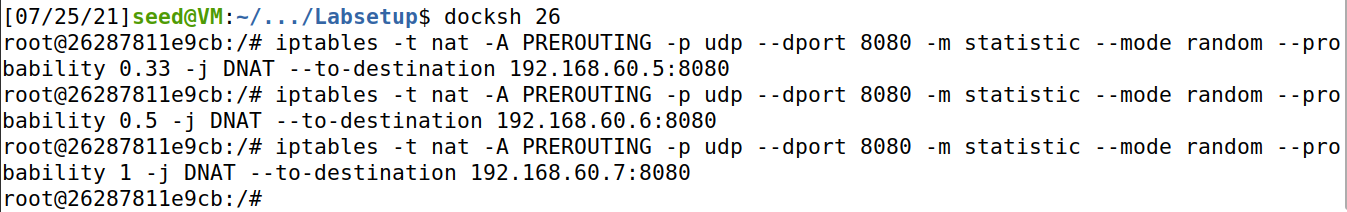
**为了负载均衡的代码**

****

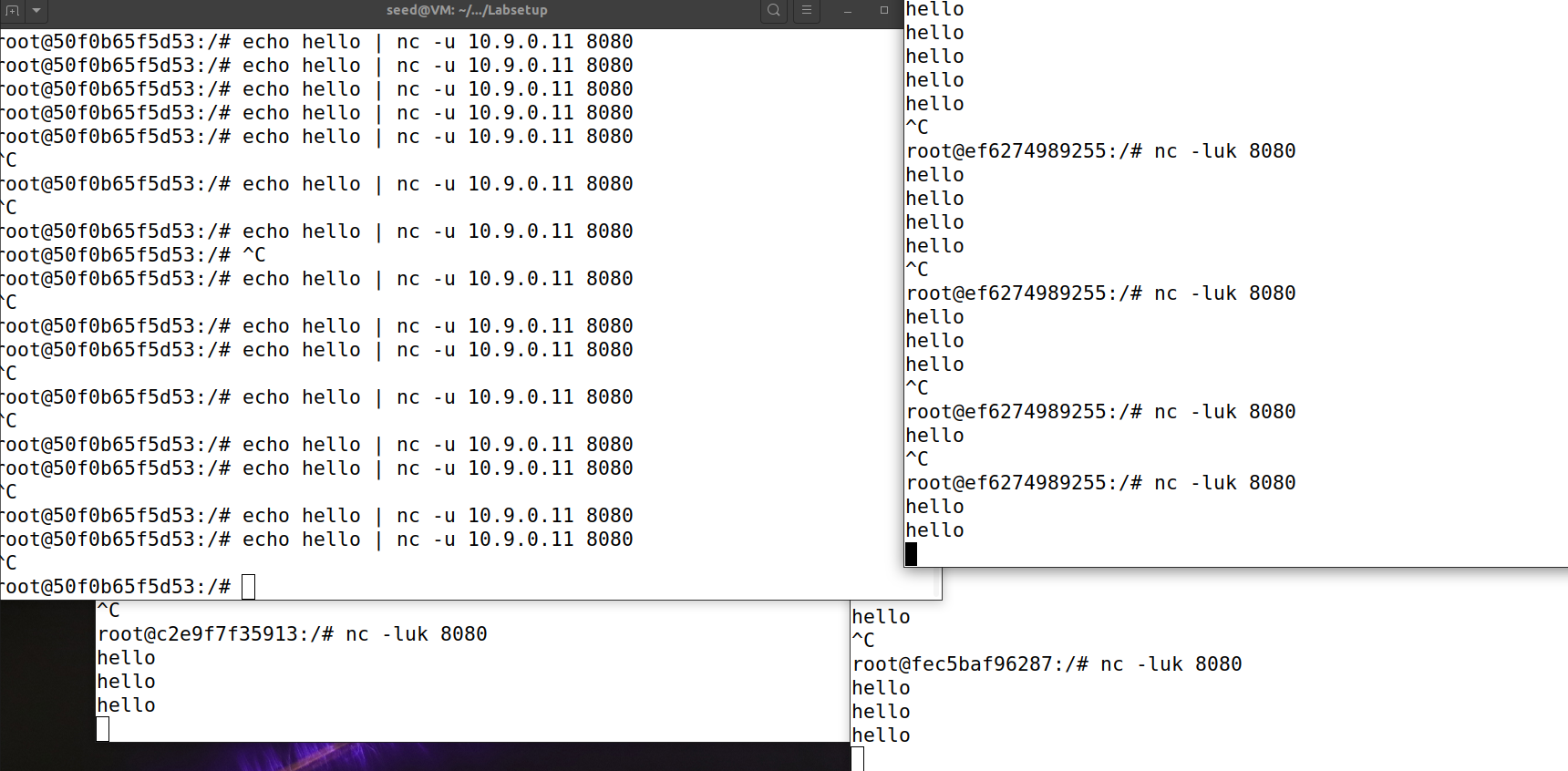
**可以看到结果平均**

****

**以概率的代码**

****

**结果也可以看到是乱序接收的但总的来说是平均的**

****