

[趣谈网络协议](#)[首页](#) | [🔍](#)

第26讲 | 云中的网络安全：虽然不是土豪，也需要基本安全和保障

2018-07-16 刘超

**朗读人：刘超**

时长14:31 大小6.66M



在今天的內容开始之前，我先卖个关子。文章结尾，我会放一个**超级彩蛋**，所以，今天的內容你一定要看到最后哦！

上一节我们看到，做一个小区物业维护一个大家共享的环境，还是挺不容易的。如果都是自觉遵守规则的住户那还好，如果遇上不自觉的住户就会很麻烦。

就像公有云的环境，其实没有你想的那么纯净，各怀鬼胎的黑客到处都是。扫描你的端口呀，探测一下你启动的什么应用啊，看一看是否有各种漏洞啊。这就像小偷潜入小区后，这儿看看，那儿瞧瞧，窗户有没有关严了啊，窗帘有没有拉上啊，主人睡了没，是不是时机潜入室内啊，等等。

假如你创建了一台虚拟机，里面明明跑了一个电商应用，这是你非常重要的一个应用，你会把它进行安全加固。这台虚拟机的操作系统里，不小心安装了另外一个后台应用，监听着一个端口，而你的警觉性没有这么高。

虚拟机的这个端口是对着公网开放的，碰巧这个后台应用本身是有漏洞的，黑客就可以扫描到这个端口，然后通过这个后台应用的端口侵入你的机器，将你加固好的电商网站黑掉。这就像你买了一个五星级的防盗门，卡车都撞不开，但是厕所窗户的门把手是坏的，小偷从厕所里面就进来了。

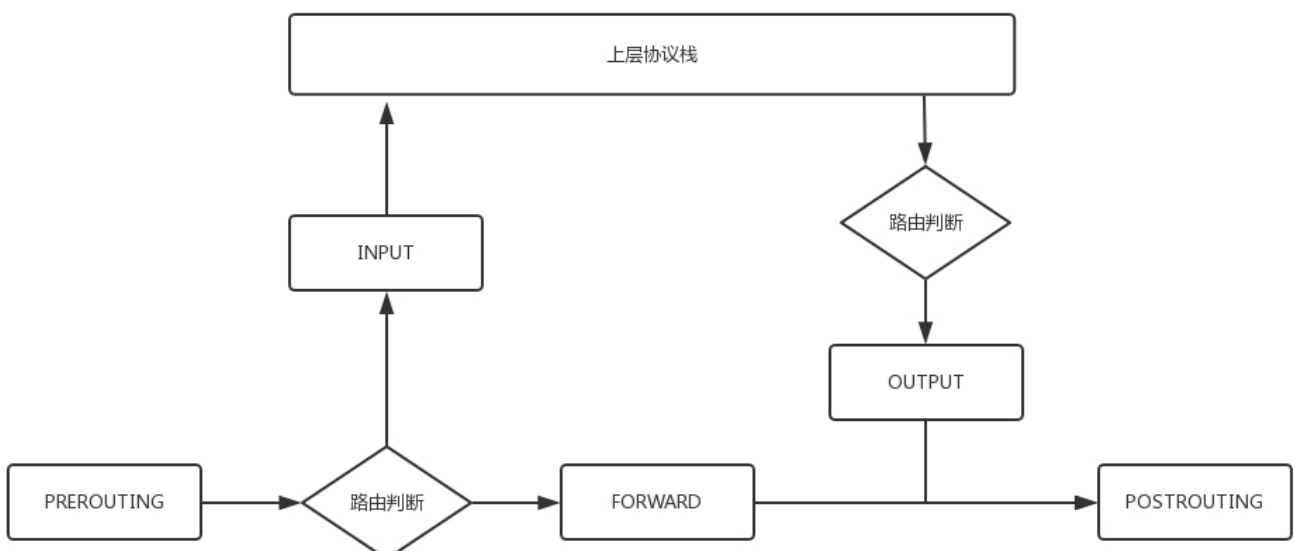
所以**对于公有云上的虚拟机，我的建议是仅仅开放需要的端口，而将其他的端口一概关闭。这个时候，你只要通过安全措施守护好这个唯一的入口就可以了。**采用的方式常常是用**ACL**（Access Control List，访问控制列表）来控制 IP 和端口。

设置好了这些规则，只有指定的 IP 段能够访问指定的开放接口，就算有个有漏洞的后台进程在那里，也会被屏蔽，黑客进不来。在云平台上，这些规则的集合常称为**安全组**。那安全组怎么实现呢？

我们来复习一下，当一个网络包进入一台机器的时候，都会做什么事情。

首先拿下 MAC 头看看，是不是我的。如果是，则拿下 IP 头来。得到目标 IP 之后呢，就开始进行路由判断。在路由判断之前，这个节点我们称为**PREROUTING**。如果发现 IP 是我的，包就应该是我的，就发给上面的传输层，这个节点叫作**INPUT**。如果发现 IP 不是我的，就需要转发出去，这个节点称为**FORWARD**。如果是我的，上层处理完毕完毕后，一般会返回一个处理结果，这个处理结果会发出去，这个节点称为**OUTPUT**，无论是 FORWARD 还是 OUTPUT，都是路由判断之后发生的，最后一个节点是**POSTROUTING**。

整个过程如图所示。



整个包的处理过程还是原来的过程，只不过为什么要格外关注这**五个节点**呢？

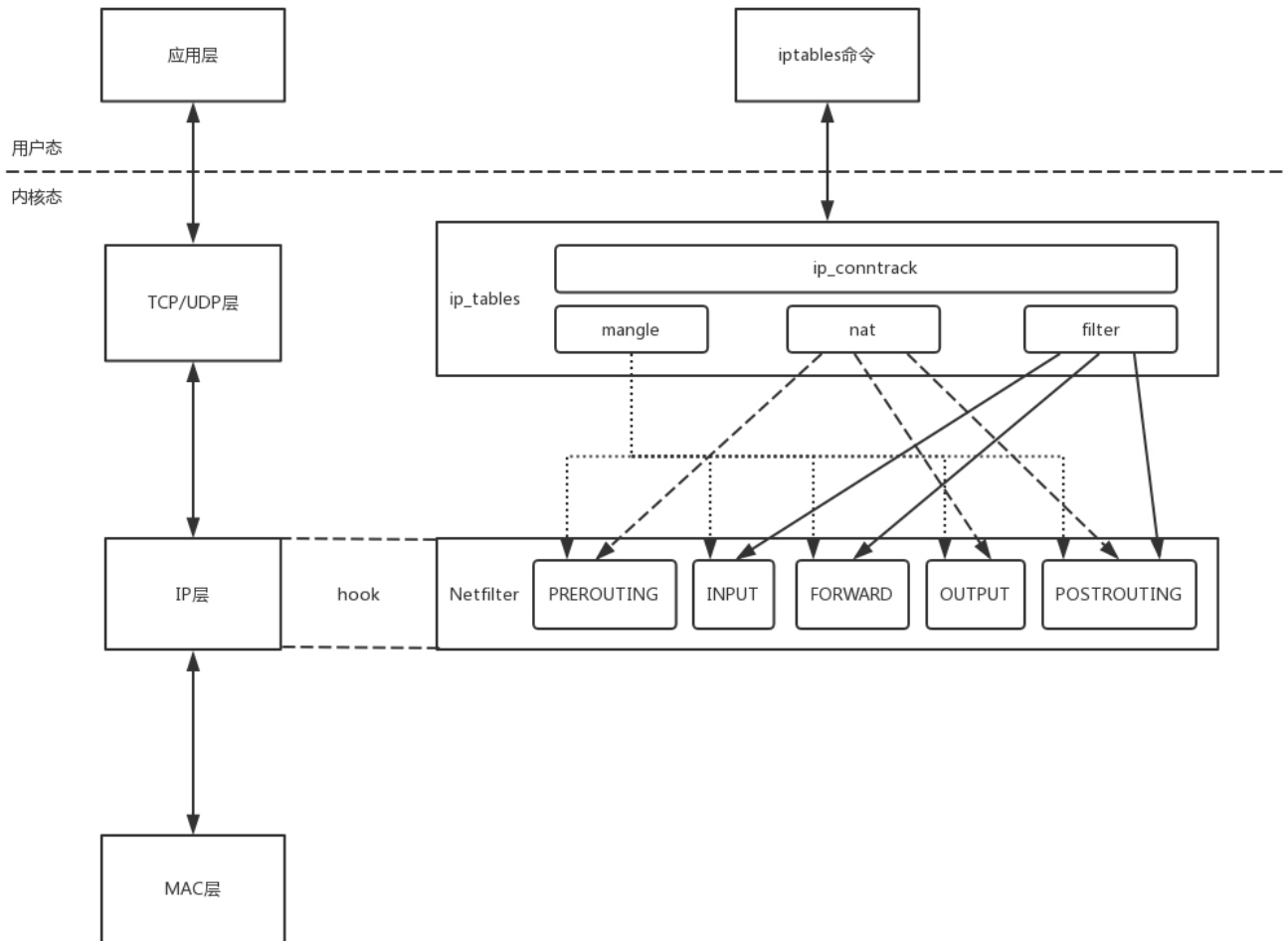
是因为在 Linux 内核中，有一个框架叫 Netfilter。它可以在这些节点插入 hook 函数。这些函数可以截获数据包，对数据包进行干预。例如做一定的修改，然后决策是否接着交给 TCP/IP 协议栈处理；或者可以交回给协议栈，那就是**ACCEPT**；或者过滤掉，不再传输，就是**DROP**；还有就是**QUEUE**，发送给某个用户态进程处理。

这个比较难理解，经常用在内部负载均衡，就是过来的数据一会儿传给目标地址 1，一会儿传给目标地址 2，而且目标地址的个数和权重都可能变。协议栈往往处理不了这么复杂的逻辑，需要写一个函数接管这个数据，实现自己的逻辑。

有了这个 Netfilter 框架就太好了，你可以在 IP 转发的过程中，随时干预这个过程，只要你能实现这些 hook 函数。

一个著名的实现，就是**内核模块 ip_tables**。它在这五个节点上埋下函数，从而可以根据规则进行包的处理。按功能可分为四大类：连接跟踪（conntrack）、数据包的过滤（filter）、网络地址转换（nat）和数据包的修改（mangle）。其中连接跟踪是基础功能，被其他功能所依赖。其他三个可以实现包的过滤、修改和网络地址转换。

在用户态，还有一个你肯定知道的客户端程序 iptables，用命令行来干预内核的规则。内核的功能对应 iptables 的命令行来讲，就是**表和链**的概念。



iptables 的表分为四种：raw→mangle→nat→filter。这四个优先级依次降低，raw 不常用，所以主要功能都在其他三种表里实现。每个表可以设置多个链。

filter 表处理过滤功能，主要包含三个链：

INPUT 链：过滤所有目标地址是本机的数据包；

FORWARD 链：过滤所有路过本机的数据包；

OUTPUT 链：过滤所有由本机产生的数据包。

nat 表主要是处理网络地址转换，可以进行 Snat（改变数据包的源地址）、Dnat（改变数据包的目标地址），包含三个链：

PREROUTING 链：可以在数据包到达防火墙时改变目标地址；

OUTPUT 链：可以改变本地产生的数据包的目标地址；

POSTROUTING 链：在数据包离开防火墙时改变数据包的源地址。

mangle 表主要是修改数据包，包含：

PREROUTING 链；

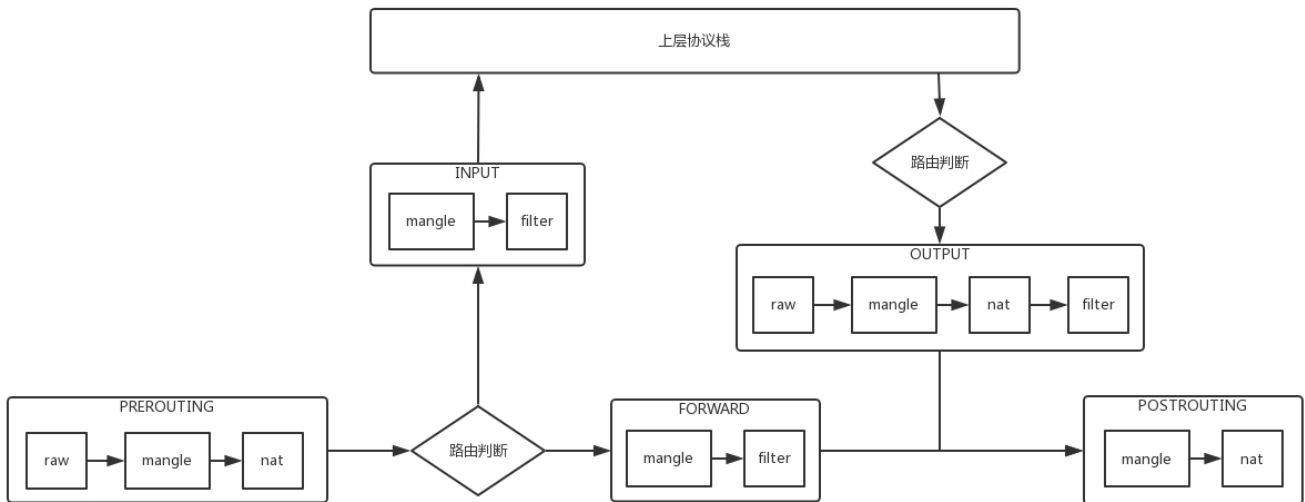
INPUT 链；

FORWARD 链；

OUTPUT 链；

POSTROUTING 链。

将 iptables 的表和链加入到上面的过程图中，就形成了下面的图和过程。



1. 数据包进入的时候，先进 mangle 表的 PREROUTING 链。在这里可以根据需要，改变数据包头内容之后，进入 nat 表的 PREROUTING 链，在这里可以根据需要做 Dnat，也就是目标地址转换。
2. 进入路由判断，要判断是进入本地的还是转发的。
3. 如果是进入本地的，就进入 INPUT 链，之后按条件过滤限制进入。
4. 之后进入本机，再进入 OUTPUT 链，按条件过滤限制出去，离开本地。
5. 如果是转发就进入 FORWARD 链，根据条件过滤限制转发。
6. 之后进入 POSTROUTING 链，这里可以做 Snat，离开网络接口。

有了 iptables 命令，我们就可以在云中实现一定的安全策略。例如我们可以处理前面的偷窥事件。首先我们将所有的门都关闭。

复制代码

```
1 iptables -t filter -A INPUT -s 0.0.0.0/0.0.0.0 -d X.X.X.X -j DROP
```


-s 表示源 IP 地址段，-d 表示目标地址段，DROP 表示丢弃，也即无论从哪里来的，要想访问我这台机器，全部拒绝，谁也黑不进来。

但是你发现坏了，ssh 也进不来了，都不能远程运维了，可以打开一下。

 复制代码

```
1 iptables -I INPUT -s 0.0.0.0/0.0.0.0 -d X.X.X.X -p tcp --dport 22 -j ACCEPT
```

如果这台机器是提供的是 web 服务，80 端口也应该打开，当然一旦打开，这个 80 端口就需要很好的防护，但是从规则角度还是要打开。

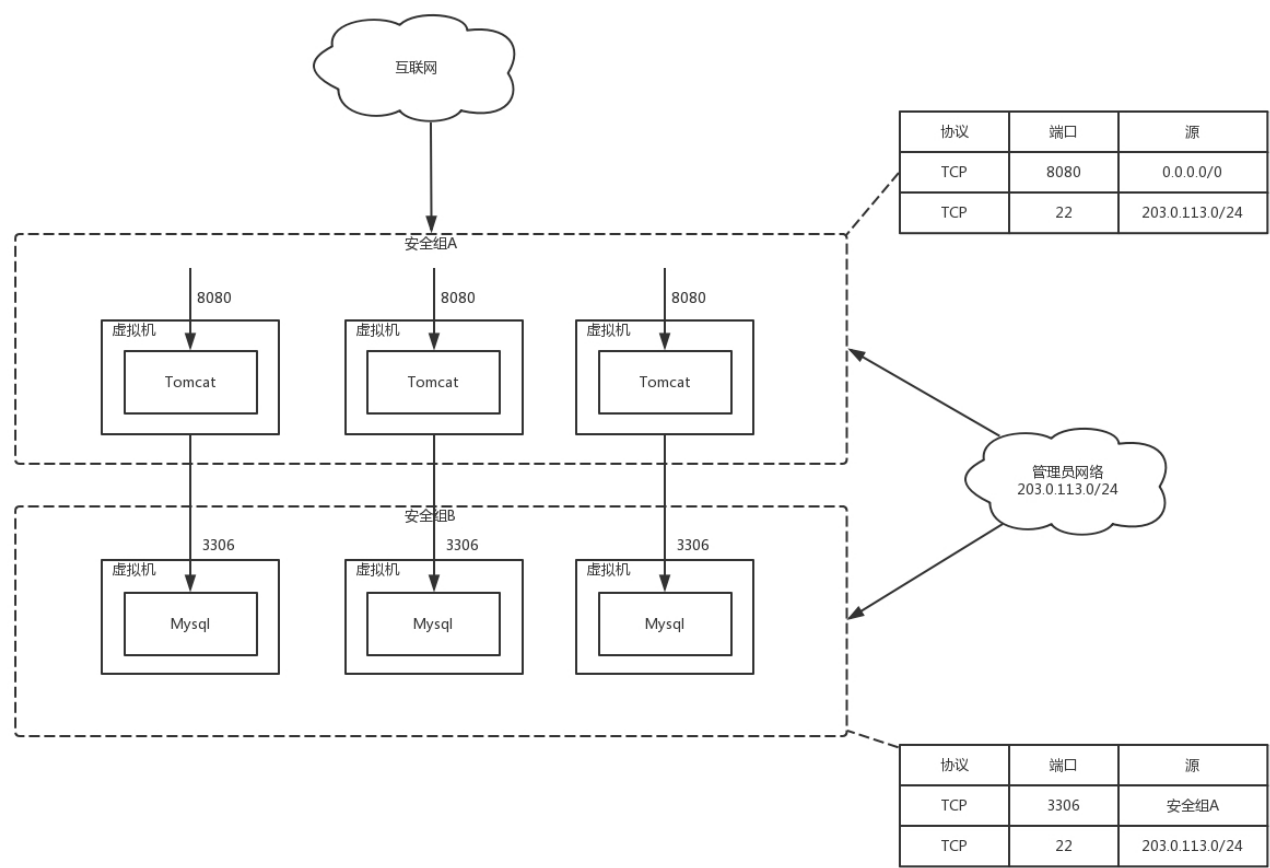
 复制代码

```
1 iptables -A INPUT -s 0.0.0.0/0.0.0.0 -d X.X.X.X -p tcp --dport 80 -j ACCEPT
```

这样就搞定了，其他的账户都封死，就一个防盗门可以进出，只要防盗门是五星级的，就比较安全了。

这些规则都可以在虚拟机里，自己安装 iptables 自己配置。但是如果虚拟机数目非常多，都要配置，对于用户来讲就太麻烦了，能不能让云平台把这部分工作做掉呢？

当然可以了。在云平台上，一般允许一个或者多个虚拟机属于某个安全组，而属于不同安全组的虚拟机之间的访问以及外网访问虚拟机，都需要通过安全组进行过滤。

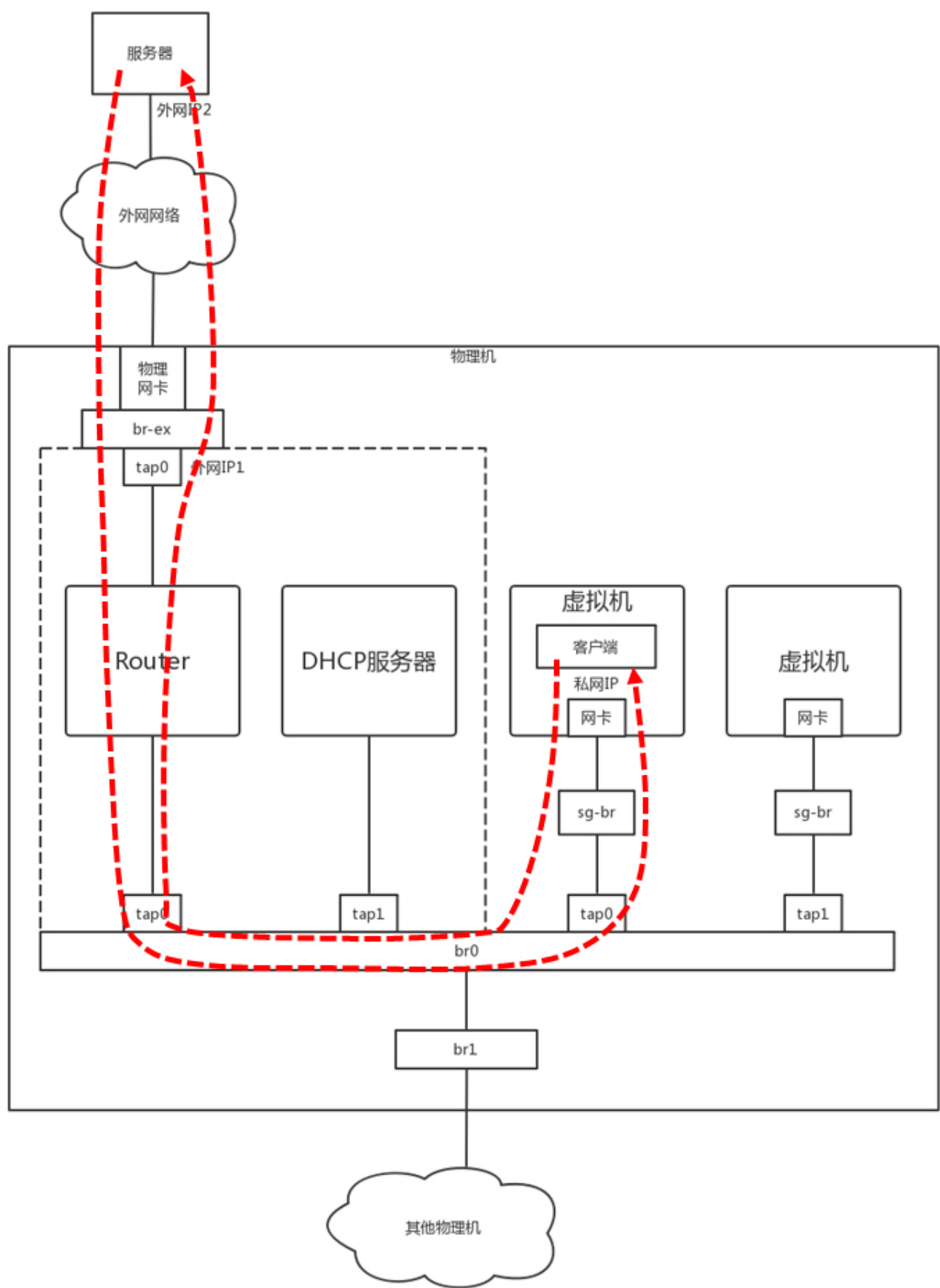


例如图中，我们会创建一系列的网站，都是前端在 Tomcat 里面，对外开放 8080 端口。数据库使用 MySQL，开放 3306 端口。

为了方便运维，我们创建两个安全组，将 Tomcat 所在的虚拟机放在安全组 A 里面。在安全组 A 里面，允许任意 IP 地址 0.0.0.0/0 访问 8080 端口，但是对于 ssh 的 22 端口，仅仅允许管理员网段 203.0.113.0/24 访问。

我们将 MySQL 所在的虚拟机在安全组 B 里面。在安全组 B 里面，仅仅允许来自安全组 A 的机器访问 3306 端口，但是对于 ssh 的 22 端口，同样允许管理员网段 203.0.113.0/24 访问。

这些安全组规则都可以自动下发到每个在安全组里面的虚拟机上，从而控制一大批虚拟机的安全策略。这种批量下发是怎么做到的呢？你还记得这幅图吗？



两个 VM 都通过 tap 网卡连接到一个网桥上，但是网桥是二层的，两个 VM 之间是可以随意互通的，因而需要有一个地方统一配置这些 iptables 规则。

可以多加一个网桥，在这个网桥上配置 iptables 规则，将在用户在界面上配置的规则，放到这个网桥上。然后在每台机器上跑一个 Agent，将用户配置的安全组变成 iptables 规则，配置在这个网桥上。

安全问题解决了，iptables 真强大！别忙，iptables 除了 filter，还有 nat 呢，这个功能也非常重要。

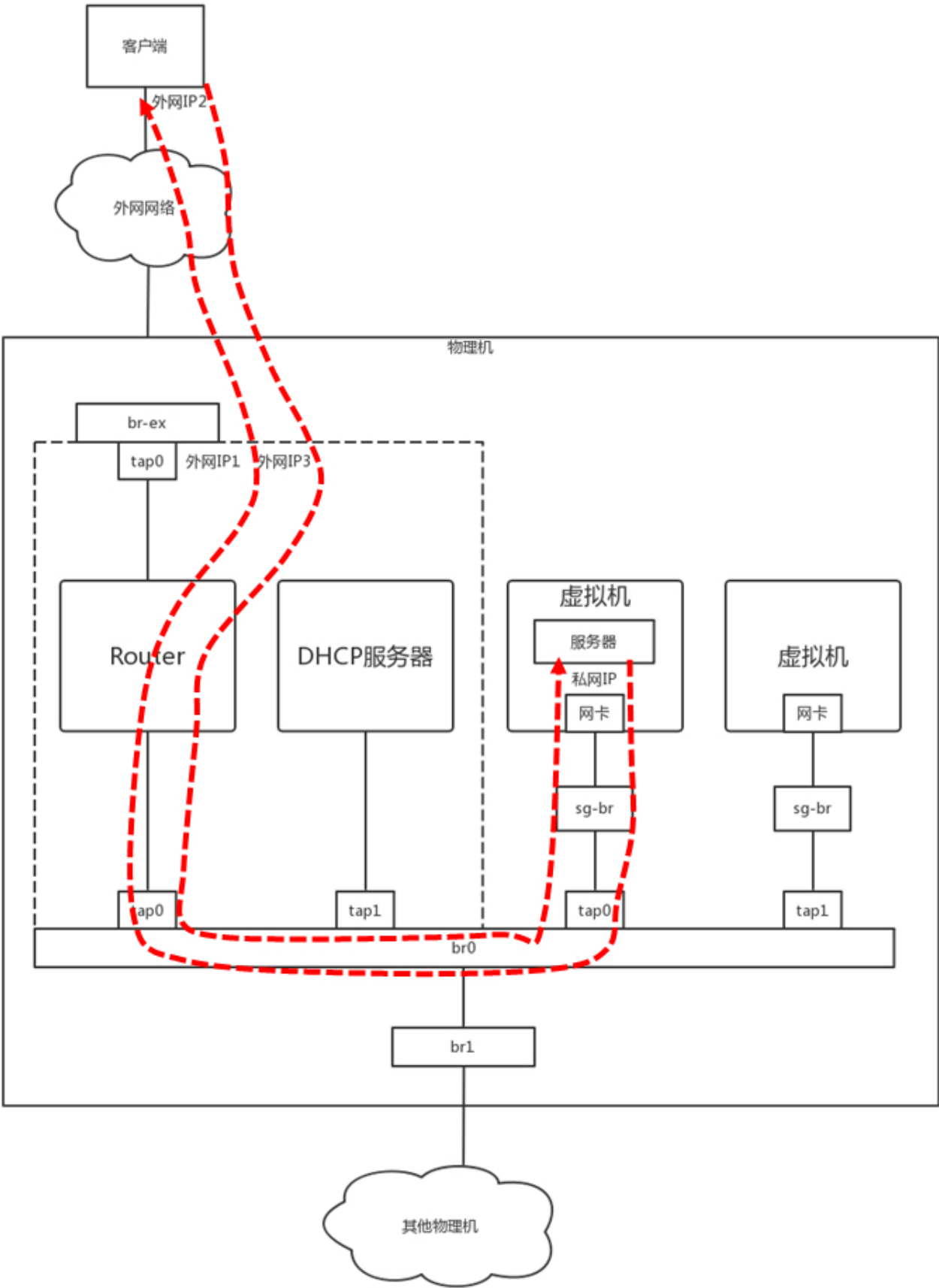
前面的章节我们说过，在设计云平台的时候，我们想让虚拟机之间的网络和物理网络进行隔离，但是虚拟机毕竟还是要通过物理网和外界通信的，因而需要在出物理网的时候，做一次网络地址转换，也即 nat，这个就可以用 iptables 来做。

我们学过，IP 头里面包含源 IP 地址和目标 IP 地址，这两种 IP 地址都可以转换成其他地址。转换源 IP 地址的，我们称为 Snat；转换目标 IP 地址的，我们称为 Dnat。

你有没有思考过这个问题，TCP 的访问都是一去一回的，而你在你家里连接 WIFI 的 IP 地址是一个私网 IP，192.168.1.x。当你通过你们家的路由器访问 163 网站之后，网站的返回结果如何能够到达你的笔记本电脑呢？肯定不能通过 192.168.1.x，这是个私网 IP，不具有公网上的定位能力，而且用这个网段的人很多，茫茫人海，怎么能够找到你呢？

所以当你从你家里访问 163 网站的时候，在你路由器的出口，会做 Snat 的，运营商的出口也可能做 Snat，将你的私网 IP 地址，最终转换为公网 IP 地址，然后 163 网站就可以通过这个公网 IP 地址返回结果，然后再 nat 回来，直到到达你的笔记本电脑。

云平台里面的虚拟机也是这样子的，它只有私网 IP 地址，到达外网网口要做一次 Snat，转换成为机房网 IP，然后出数据中心的时候，再转换为公网 IP。



这里有一个问题是，在外网网口上做 Snat 的时候，是全部转换成一个机房网 IP 呢，还是每个虚拟机都对应一个机房网 IP，最终对应一个公网 IP 呢？前面也说过，公网 IP 非常贵，虚拟机也很多，当然不能每个都有单独的机房网和公网 IP 了，于是这种 Snat 是一种特殊的 Snat，MASQUERADE（地址伪装）。

这种方式下，所有的虚拟机共享一个机房网和公网的 IP 地址，所有从外网网口出去的，都转换成为这个 IP 地址。那又一个问题来了，都变成一个公网 IP 了，当 163 网站返回结果的时候，给谁呢，再 nat 成为哪个私网的 IP 呢？

这就是 Netfilter 的连接跟踪（conntrack）功能了。对于 TCP 协议来讲，肯定是上来先建立一个连接，可以用“源 / 目的 IP+ 源 / 目的端口”唯一标识一条连接，这个连接会放在 conntrack 表里面。当时是这台机器去请求 163 网站的，虽然源地址已经 Snat 成公网 IP 地址了，但是 conntrack 表里面还是有这个连接的记录的。当 163 网站返回数据的时候，会找到记录，从而找到正确的私网 IP 地址。

这是虚拟机做客户端的情况，如果虚拟机做服务器呢？也就是说，如果虚拟机里面部署的就是 163 网站呢？

这个时候就需要给这个网站配置固定的物理网的 IP 地址和公网 IP 地址了。这时候就需要显示的配置 Snat 规则和 Dnat 规则了。

当外部访问进来的时候，外网网口会通过 Dnat 规则将公网 IP 地址转换为私网 IP 地址，到达虚拟机，虚拟机里面是 163 网站，返回结果，外网网口会通过 Snat 规则，将私网 IP 地址转换为那个分配给它的固定的公网 IP 地址。

类似的规则如下：

源地址转换 (Snat): `iptables -t nat -A -s 私网 IP -j Snat --to-source 外网 IP`

目的地址转换 (Dnat): `iptables -t nat -A -PREROUTING -d 外网 IP -j Dnat --to-destination 私网 IP`

到此为止 iptables 解决了非法偷窥隐私的问题。

小结

好了，这一节就讲到这里了，我们来总结一下。

云中的安全策略的常用方式是，使用 iptables 的规则，请记住它的五个阶段，PREROUTING、INPUT、FORWARD、OUTPUT、POSTROUTING。

iptables 分为四种表，raw、mangle、nat、filter。其中安全策略主要在 filter 表中实现，而虚拟网络和物理网络地址的转换主要在 nat 表中实现。

最后，给你留两个思考题。

1. 这一节中重点讲了 iptables 的 filter 和 nat 功能，iptables 还可以通过 QUEUE 实现负载均衡，你知道怎么做吗？
2. 这一节仅仅讲述了云中偷窥的问题，如果是一个合法的用户，但是不自觉抢占网络通道，应该采取什么策略呢？

我们的专栏更新到第 26 讲，不知你掌握得如何？是不是有很多问题想要跟我面对面探讨呢？这里就有一个机会。

今天晚上 8:30，我会在极客时间 APP 里做一个直播，主题是“技术人如何在技术浪潮中线性成长？”，我会把我们讲过的网络协议作为案例，在直播中展开讲解，也会分享我从业多年来的心得体会。你可以直接在这里留言提问，也可以准备好问题在直播的时候和我交流。

欢迎你来观看直播！我们晚上见！



© 版权归极客邦科技所有，未经许可不得转载

上一篇 第25讲 | 软件定义网络：共享基础设施的小区物业管理办法

下一篇 第27讲 | 云中的网络QoS：邻居疯狂下电影，我该怎么办？

精选留言

写留言



rtoday

2018-07-18

10

关于网络安全
不知道课程结束后
可否拨冗谈一下...

展开



固态U盘

2018-07-16

6

多年都没具体弄清楚iptables,这一篇文章解惑了，多谢刘老师。



大树

2018-07-17

4

老师好，直播错过了，有没有回看呢？



timgise

2018-07-16

4

期待



小宇宙

2018-08-24

2

k8s的kube-proxy 就是利用的iptables 做流量转发和负载均衡的，service 利用nat 将相应的流量转发到对应的pod中，另外iptables 有一个probability特性，可以设置probability的百分比是多少，从而实现负载均衡

作者回复: 对的

**balancer**

2018-07-27

👍 1

老师，后面能不能加点对网络的检测，排查案例，课程就完整了

**勤劳的小胖子-libo**

2018-07-16

👍 1

这章总结太给力了，很喜欢♡

里面的机房网ip就是物理网ip地址吧？

**stany**

2018-07-16

👍 1

刘老师好，我也是网易的同学，关于表和链的那张图，我在别的资料上，看到在 output 部分和 forward 部分之后汇聚的链路上，还有一个路由策略 routing decision 的？但是这里不能传图，传不上来给你看。

**人丑就应多读书**

2018-11-07

👍 0

老师我有个问题，之前讲了gslb的原理，那么本地负载均衡slb是通过netfilter进行的实现吗？

展开 ∨

作者回复：负载均衡实现方法比较多，lvs, haproxy都可以

**周飞**

2018-10-17

👍 0

感谢老师的讲解，以前只是了解nat的大体原理，现在对整个过程有了更深入的了解。同时对云主机的安全策略和实现原理有个深入的认识，这门课非常值得学生和程序员学习。

展开 ∨

**Sinderlar**

2018-08-01

👍 0

看完后对iptables 配置特别理解...明白为啥这么配置.....而不是复制改端口配置.....

**学徒王小明**

2018-07-21

👍 0

老师，运营商做Snat是将ipv4转ipv6吗？

作者回复: 不是的，转成互联网的公网ip

**Hurt**

2018-07-16

👍 0

👍👍 期待

**魏峰**

2018-07-16

👍 0

提纲挈领，把以前学的东西都融汇贯通了。如果想要把这些知识往深里面学学，看看有啥好的建议没

展开 ▾

**梦想启动的蜗牛**

2018-07-16

👍 0

直接看彩蛋😁

**Jay**

2018-07-16

👍 0

看直播啦