Iptables 面试

目录

一、 什么是 iptables

iptables 主要工作在网络层，可以通过规则控制数据包的进出方向、源/目的地址、源/目的端口等信息，来实现对包的过滤、转发、重定向等操作。

二、 iptables 中有几个基本概念

1、Chain（链）

链是 iptables 规则的容器，用于存放一组规则。Linux 内核中预定义了五个链：INPUT、FORWARD、OUTPUT、PREROUTING 和 POSTROUTING。

它们的功能分别如下：

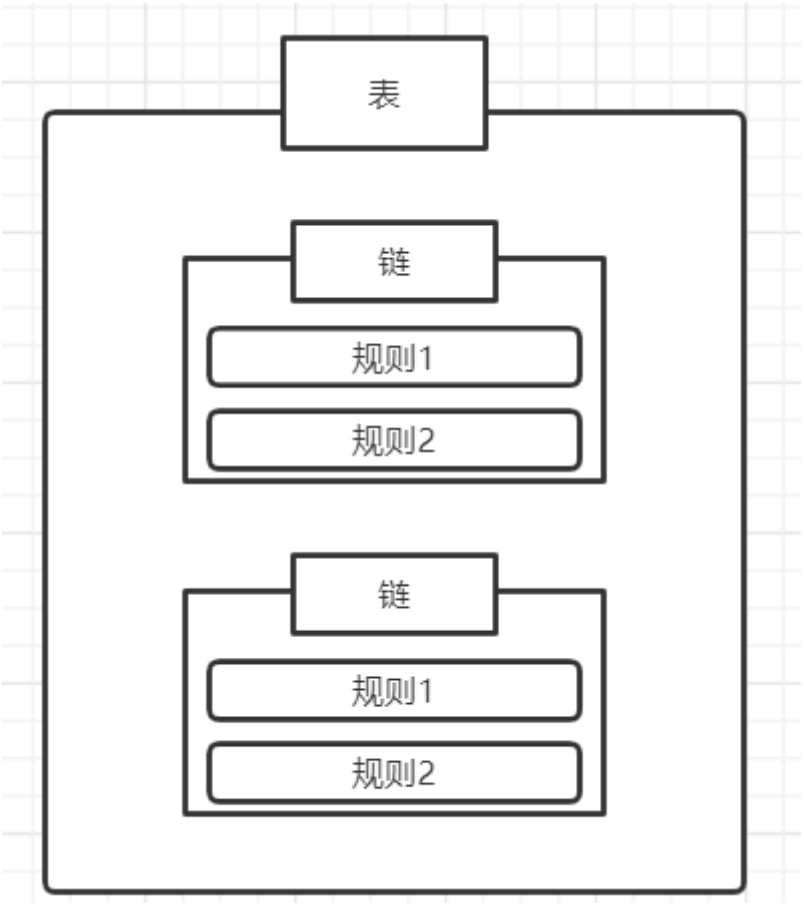
链	**功能**
INPUT	区配进入本机，并且目标 IP 是本机地址的数据包；
FORWARD	匹配由本机进行转发的数据包；
OUTPUT	匹配由本机发出的数据包
PREROUTING	匹配刚到达本机的所有数据包，可用于修改目的地址，通常用于实现 DNAT 功能。

POSTROUTING	匹配即将离开本机的数据包，可用于修改源地址，通常用于实现 SNAT 功能。

2. Rule（规则）

规则定义了对数据包进行处理的具体动作，由匹配条件和动作两部分组成。匹配条件可以是数据包的源地址、目的地址、端口等信息，动作可以是 ACCEPT（允许通过）、DROP(丢弃数据包）、REJECT（拒绝）等。配置规则时需要指定所属的表和链，如果不指定表则默认为 filter 表。3. Table（表）

表是存放链的容器，每个表包含若干个链。iptables 中有四个预定义的表：filter、nat、mangle 和 raw，其中 filter 和 nat 表是被使用最多的，分别用于数据包过滤和网络地址转换功能。



三、 iptables 的使用方法

1. 显示规则

要查看当前的 iptables 规则，可以使用以下命令：

```
$ iptables -L
```

这将显示所有规则，包括输入、输出和转发规则。通过观察规则列表，您可以了解当前网络流量的策略。

2. 添加规则

要添加一个规则，可以使用以下命令：

```
iptables -A <chain> <rule>
```

其中，**<chain>**指定要添加规则的链，例如 **INPUT**、**OUTPUT** 或 **FORWARD**。**<rule>**是规则本身，可以是允许或拒绝特定类型的流量。

例如，如果您想允许 SSH 连接（TCP 端口 22），可以使用以下命令：

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

这将在 INPUT 链中添加一个规则，允许 TCP 端口 22 的流量通过。

3. 删除规则

要删除一个规则，可以使用以下命令：

```
iptables -D <chain> <rule>
```

其中，**<chain>**和**<rule>**是要删除的规则的链和规则本身。例如，如果您要删除前面添加的 SSH 规则，可以使用以下命令：

```
iptables -D INPUT -p tcp --dport 22 -j ACCEPT
```

这将删除输入链中允许 TCP 端口 22 的流量通过的规则。

4. 修改规则

要修改一个规则，可以先删除旧规则，再添加新规则，或者使用以下命令修改规则：

```
iptables -R <chain> <rule_number> <new_rule>
```

其中，**<chain>**是要修改的规则的链，**<rule_number>**是要修改的规则的编号（可以在 iptables -L 命令的输出中找到），**<new_rule>**是要添加的新规则。

例如，如果您要将输入链中允许 SSH 连接的规则更改为允许 HTTP 连接（TCP 端口 80），可以使用以下命令：

```
iptables -R INPUT 1 -p tcp --dport 80 -j ACCEPT
```

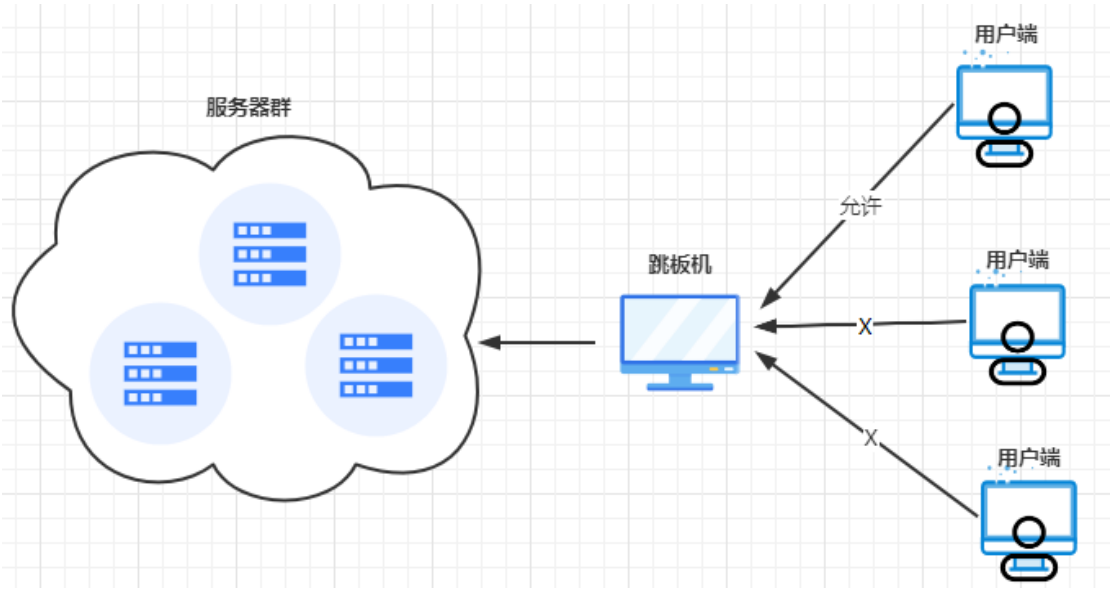
这将输入链中编号为 1 的规则更改为允许 TCP 端口 80 的流量通过。

iptables 是一个功能非常强大的工具，以上只是几个最简单的使用示例，还有很多高级选项和用法。限于文章篇幅，此处不做介绍，感兴趣的读者可自行查阅相关文档资料。

四、 iptables 的应用案例有哪些

这是之前在工作中的一个 iptables 应用案例，虽然场景简单但参考性较强，可便于举一反三的应用到其他场景中。首先说下场景：公司内部有几台跳板机，用于给到特定的人员登录访问服务器。由于之前出现过个别跳板机被黑客入侵的情况，现在需要对其进行安全加固。经过考虑以后，决定使用 iptables 的防火墙功能来实现，这类方案的特点是简单且成本低。

安全方面的需求很简单，需要对跳板机进行访问限制，只允许指定 IP 的机器访问该主机，从而达到阻止黑客入侵系统的目的。同时 ，需要保证跳板机可以正常访问业务系统。



本案例以操作系统 Centos7.x 为例来进行演示，该需求将使用 filter 表和其中的 INPUT 链来实现。

1. 关闭 firewalld 防火墙。

```
$ systemctl stop firewalld

$ systemctl disable firewalld
```

2. 安装 iptables-services ，并启用服务。

```
$ yum install iptables-services

$ systemctl enable iptables

$ systemctl start iptables
```

3. 清空现有 INPUT 链的规则。

```
$ iptables -F INPUT
```

4. 配置访问规则

```
 # 开放指定 IP 访问，此处 IP 根据实际情况变更

$ iptables  -A INPUT  -s 192.168.4.168 -j ACCEPT

# 配置状态为 RELATED、ESTABLISHED 的连接可通过，保证跳板机对外发起请求，对端的回复可通过。

$ iptables -A INPUT  -p tcp -m state --state RELATED,ESTABLISHED -j ACCEPT

# 开放 icmp 访问

$ iptables -A INPUT -p icmp -j ACCEPT

# 开放 dns 访问，用于域名解析

$ iptables -A INPUT -p udp --sport 53 -j ACCEPT

# 开放时间服务器访问，用于时间同步

$ iptables -A INPUT -p udp --sport 123 -j ACCEPT
```

注释：-A 用于添加规则到表；-s 用于指定源 IP；-p 用于指定协议；--sport 用于指定源端口，此处用于放行时间服务器和 DNS 服务器返回的数据包；--state RELATED,ESTABLISHED 用于指定连接的状态，用于放行目标机器返回的数据包。

5. 更改 INPUT 默认规则为 DROP，屏蔽其他所有连接。

```
$ iptables -P INPUT DROP
```

注意：该规则需要在上方其他规则配置完成后，才能执行。否则会导致机器无法连接。

6. 查看规则，已完成。

```
$ iptables -nL INPUT

Chain INPUT (policy DROP)

target     prot opt source               destination

ACCEPT     all  --  192.168.4.168        0.0.0.0/0

ACCEPT     tcp  -

-  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     udp  --  0.0.0.0/0            0.0.0.0/0           udp spt:53

ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp spt:123
```

7. 保存 iptables 配置，否则重启机器后会丢失。

```
$ service iptables save
```