上章回顾

1. 如何在SSH服务器的配置文件中限制root用户登录?
    a. vim /etc/ssh/sshd_config
        i. PermitRootLogin no
2. 构建密钥对验证的SSH服务的步骤有哪几步?
    a. 客户机通过ssh-keygen -t rsa生成秘钥对，默认存储在当前用户家目录的.ssh目录下
    b. 客户机公钥通过scp方式发送到服务器
    c. 服务端将公钥文件内容写入到密钥对登录的用户的宿主目录下的.ssh目录中，并且需要添加到authorized_keys文件中
    d. 客户机通过指定的用户名称验证登录
3. TCP Wrappers的策略配置格式是什么?
    a. 服务名称 : IP地址
    b. 通过两个文件控制着策略的信息
        i. /etc/hosts.allow：允许列表
        ii. /etc/hosts.deny：拒绝列表
    c. 先看允许在看拒绝，如果都没有则默认允许

本章内容

- 熟悉Linux防火墙的表、链结构
- 理解数据包匹配的基本流程
- 学会编写iptables规则

一、防火墙概述

1、防火墙是一种协助确保信息安全的网络设备，通过一定的规则，允许或限制数据包的通过；防火墙主要分为两类:

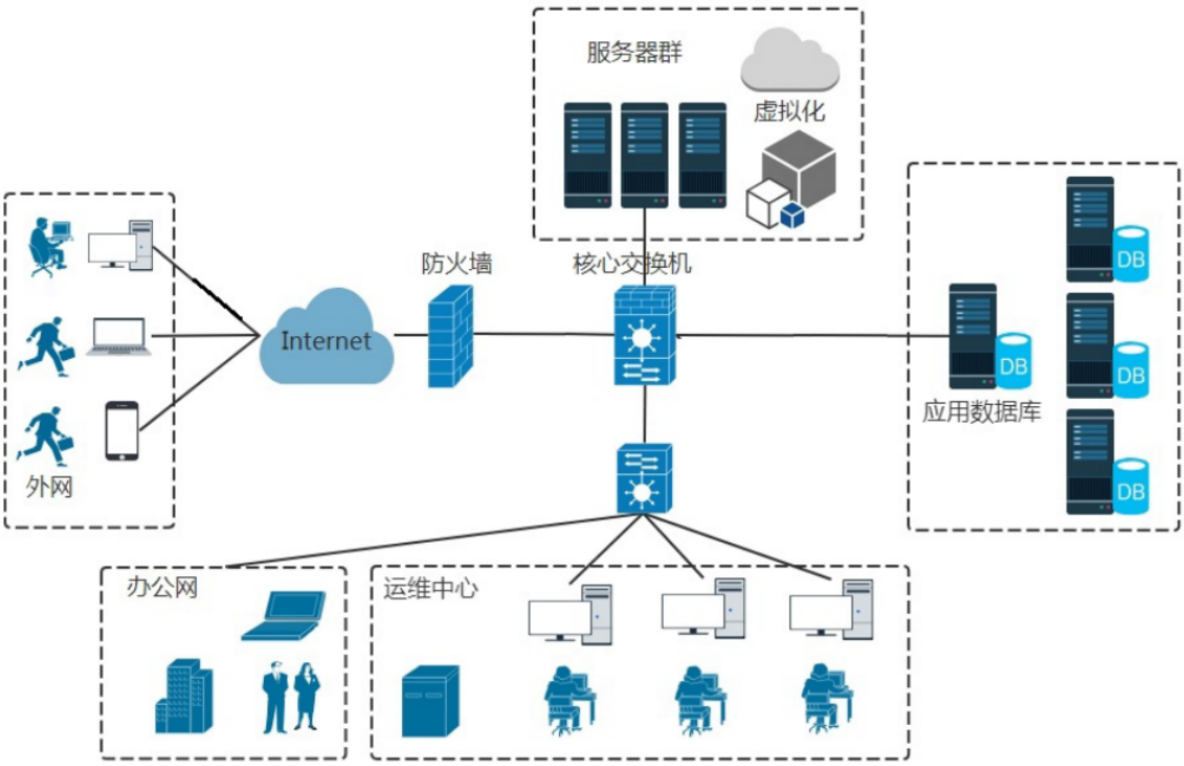
1）逻辑防火墙:

- 网络防火墙:
    - 这种防火墙通常放置在被保护网络的边界，例如企业内部网络和互联网之间。
    - 主要作用是为了隔离风险区域（Internet）与安全区域（局域网）的连接，同时不会妨碍安全区域到风险区域的访问。
    - 为了保证安全防御作用，所有进出被保护网络的通信数据流都必须经过防火墙，所有通过防火墙的数据流都必须经过安全策略的过滤或者防火墙的授权
- 主机防火墙:
    - 这种防火墙安装在每个网络节点上，控制每个传入和传出的数据包。
    - 它主要负责监控与保护该主机相关的所有网络连接，确保任何未经授权的访问都被阻止。

2）物理防火墙:

- 硬件防火墙:
    - 硬件防火墙是一种物理设备，它独立于主机设备，有着自己的专用资源，不会占用主机设备的任何CPU或RAM。
    - 硬件防火墙由于其专用的硬件和操作系统，通常具有更好的稳定性和更高的安全性，因此大多数企业用户会选择使用硬件防火墙

- 软件防火墙：
  - 通过在主机上安装特定的软件程序来实现
  - 软件防火墙的初始成本相对较低，对于预算有限的用户来说是一个不错的选择。



二、Netfilter和iptables

1、Netfilter和iptables是一套在Linux上用于实现包过滤和网络转发的高效工具。这套工具由两个部分组成：Netfilter框架和iptables工具

2、Netfilter是一个由Linux内核提供的工作在内核空间的网络数据包处理框架，其主要功能就是对进出内核协议栈的数据包进行过滤或者修改

3、iptables则是工作在用户空间的工具，它负责与内核的Netfilter模块进行通信，以实现对netfilter的配置和管理。

三、iptables介绍

1、iptables是一个Linux内核防火墙工具，用于配置和管理Linux系统上的网络数据包过滤规则。

2、它允许用户根据预设的规则对进入和离开系统的网络数据包进行筛选和操作。

3、iptables通过定义一系列的规则来控制网络数据包的传输，这些规则可以基于源IP地址、目标IP地址、传输协议（TCP、UDP等）、端口号等条件进行匹配。

4、iptables的主要功能包括：

- 允许/拒绝数据包的传输：根据规则的设置，iptables可以允许或拒绝网络数据包的传输，从而实现对网络访问的控制。
- 记录日志：iptables可以配置为记录匹配规则的数据包的日志，以便于分析和审计。
- 重定向数据包：通过iptables的NAT（网络地址转换）功能，可以将数据包重定向到指定的IP地址或端口。
- 流量整形：iptables可以用于控制网络流量的速率，以防止网络拥塞和拒绝服务攻击。

5、iptables的工作规则

1）iptables的规则（rules）是按照一定的顺序进行匹配的，一旦找到匹配的规则，就会执行相应的操作。

2）这些规则由网络管理员预定义，通常是一些条件，例如数据包头符合特定的条件

3）这些规则被存储在内核空间的信息包过滤表中，并指定了源地址、目的地址、传输协议（如TCP、UDP、ICMP）和服务类型（如HTTP、FTP和SMTP）等。

4）当数据包与规则匹配时，iptables就根据规则所定义的方法来处理这些数据包，如放行（accept）、拒绝（reject）和丢弃（drop）等。

5）如果没有匹配的规则，iptables会根据默认策略进行处理。

6）默认策略也可以是允许所有数据包通过、拒绝所有数据包或者根据其他条件进行转发。

四、iptables的表、链结构

1、在 Linux 系统中，iptables 通过表（tables）、链（chains）以及规则（rules）协同工作，实现对数据包精细且全面的管理。

2、规则链

1）在iptables中，链（Chain）是一个非常重要的概念，它是用来对IP数据包进行过滤、修改、转发、重定向以及丢弃等操作的一系列规则的集合。

2）每个链都包含一系列的规则，这些规则按照特定的顺序进行排列，每条规则包含一个匹配条件和一个对应的动作。

3）当IP数据包与链中的规则匹配时，就会执行相应的动作。

4）iptables中的链总共分为 5 种

- **INPUT链：处理进入本机的数据包。**
- **OUTPUT链：处理由本机发出的数据包。**
- **FORWARD链：处理经过本机的数据包，例如，路由器接收到一个目的地址不是自己的数据包，这个数据包会被送到FORWARD链进行处理**
- **PREROUTING链：处理路由决策之前的数据包，通常用于修改目的地地址。**
- **POSTROUTING链：处理路由决策之后的数据包，通常用于修改源地址。**

3、规则表

1）由于每个规则链中都设置了很多条规则，这样就可以把不同的规则链组合成能够完成某一特定功能的集合分类，这个集合分类就可以称为表

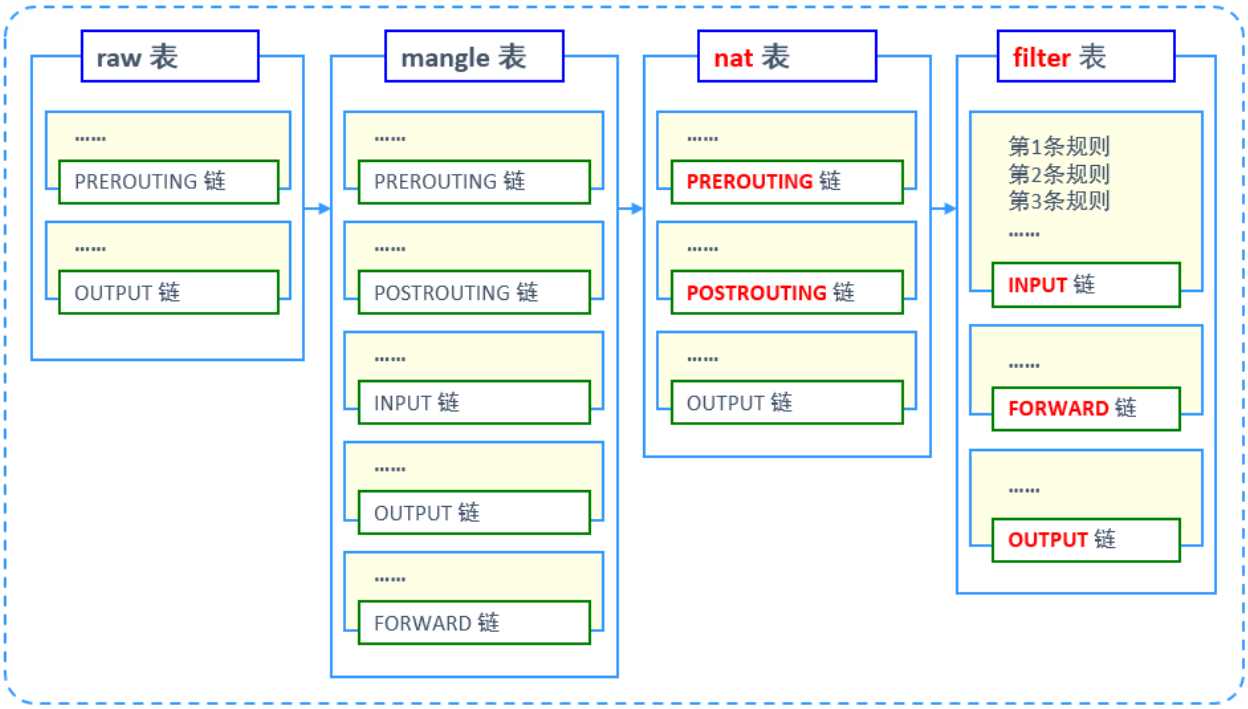
2）**在iptables中共用 4 张表：**

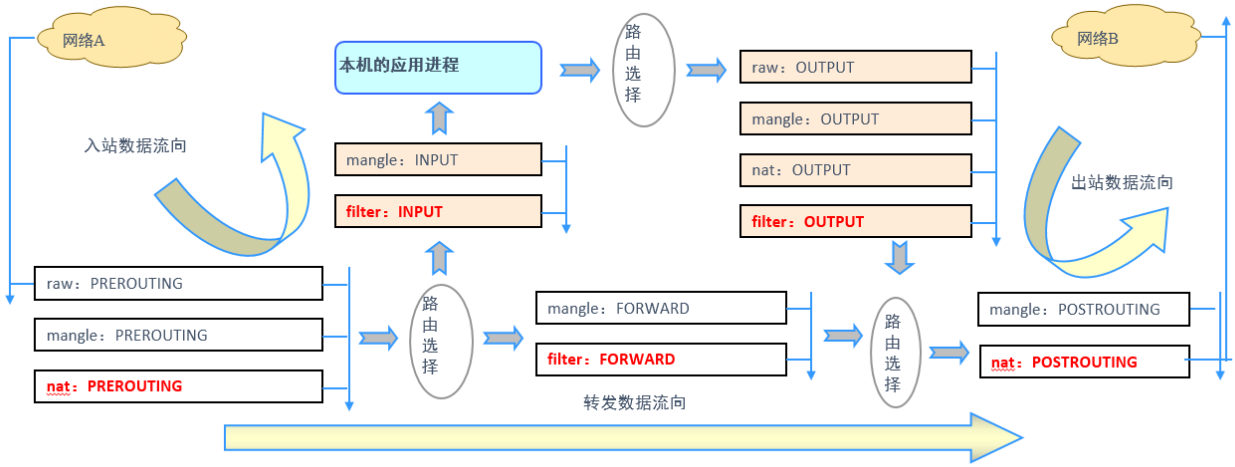
- **filter表：主要用于确定是否放行该数据包，也就是进行包过滤。同样也是默认处理的表**
- **nat表：用于修改数据包中的源、目标IP地址或端口，实现网络地址转换的功能。**
- **mangle表：表主要用于对数据包的 IP 首部信息进行修改，这些信息包含 TTL（Time To Live，生存时间）、TOS（Type of Service，服务类型）等。它能够对数据包进行标记，这些标记可在后续的规则匹配中使用。**
  - 修改 TTL：可以调整数据包在网络中的生存时间，避免数据包在网络中无限循环。
  - iptables -t mangle -A PREROUTING -j TTL --ttl-set 64

- **raw表：raw 表的主要作用是决定数据包是否要进行状态跟踪。在 iptables 中，状态跟踪机制会对数据包的连接状态进行记录和管理，这会消耗一定的系统资源。raw 表可以让某些数据包跳过状态跟踪，从而提高性能。**
  - 对特定的数据包跳过状态跟踪：对于一些不需要进行状态跟踪的数据包，如 ICMP 错误消息、多播数据包等，可以使用 raw 表让它们跳过状态跟踪，减少系统资源的消耗。

4、表和链之间的关系

- filter表可以使用的链：INPUT,FORWARD,OUTPUT
- nat表中可以使用的链：PREROUTING,OUTPUT ,POSTROUTING,INPUT
- mangle 表中可以使用的链：PREROUTING,INPUT,FORWARD,OUTPUT,POSTROUTING
- raw表中可以使用的链：PREROUTING,OUTPUT





- 网络处理表
  - raw 表：主要用于处理免除连接跟踪机制的数据包，图中显示它在 PREROUTING 和 OUTPUT 链中起作用。
  - mangle 表：用于对数据包进行修改，如修改 TTL、TOS 等，图中显示它在 PREROUTING、INPUT、FORWARD、OUTPUT 和 POSTROUTING 链中起作用。
  - nat 表：用于网络地址转换，图中显示它在 PREROUTING、OUTPUT 和 POSTROUTING 链中起作用。
  - filter 表：用于过滤数据包，决定是否允许数据包通过，图中显示它在 INPUT、FORWARD 和 OUTPUT 链中起作用。
- 数据流向
  - 入站数据流向（从网络 A 到本机）

- 数据包首先经过 PREROUTING 链处理，涉及 raw、mangle 和 nat 表。
- 接着进行路由选择，根据路由结果，数据包可能进入本机的应用进程或被转发。
- 如果数据包进入本机的应用进程，它会经过 INPUT 链处理，涉及 mangle 和 filter 表。

- 出站数据流向（从本机到网络 B）
  - 数据包从本机的应用进程出发，首先经过 OUTPUT 链处理，涉及 raw、mangle、nat 和 filter 表。
  - 然后进行路由选择，最后数据包通过网络 B 离开本机。
- 转发数据流向（从网络 A 经过本机到网络 B）
  - 数据包从网络 A 进入本机后，首先经过 PREROUTING 链处理，涉及 raw、mangle 和 nat 表。
  - 接着进行路由选择，数据包进入 FORWARD 链处理，涉及 mangle 和 filter 表。
  - 最后经过 POSTROUTING 链处理，涉及 mangle 和 nat 表，然后通过网络 B 离开本机。

- 路由选择
  - 路由选择是一个关键步骤，它决定了数据包是进入本机的应用进程还是被转发。
  - 图中在多个节点都有路由选择步骤，如在 PREROUTING 后、本机的应用进程前后以及 FORWARD 链前后。

5、iptables的简单命令

1、语法：iptables 选项

- -L：查看规则
- -n：以数字形式显示地址，端口等信息
- -t：指定表名
- -F：清空规则

五、编写防火墙规则

1、语法结构

1）iptables [-t 表名] [-I 链名] [-p 条件匹配] [-j 控制类型]

2、注意事项

1）当不指定表名时，默认指定filter表

2）不指定链名时，默认指表中的所有链

3）除非设置链的默认策略，否则必须指定匹配条件

	table	command	chain	Parameter & Xmatch	target
iptables	-t filter nat mangle raw	-A -D -L -F -P -I -R -n	INPUT FORWARD OUTPUT PREROUTING POSROUTING	-p tcp -s -d --sport --dport --dports -m tcp -m state -m multiport	-j ACCEPT DROP REJECT DNAT SNAT

3、指定规则策略

1）黑名单：没有被拒绝的流量默认被允许，这个情况下管理员必须应对每一种Internet威胁，指定新的规则，不建议使用

2）白名单：仅允许手动定义数据流，这种较为保守，根据日常工作需求，针对性开放数据流，目前几乎都是用白名单方式

4、指定规则的思路

1）选择一张表，此表决定了数据报文的处理方式

2）选择一条链，此链决定了数据报文流经哪些位置

3）选择格式的条件，此条件决定了对数据报文做何种条件的匹配

4）选择处理数据报文的动作，制定相应的防火墙规则

六、iptables基础应用

1、yum安装iptables-services

1）Centos 7 中系统默认存在iptables命令，但是仅为简单的查询和操作的命令，不包含配置文件

2）完成后将会自定生成配置文件，便于配置保存

```
[root@localhost ~] yum -y install iptables-services
[root@localhost ~] systemctl start iptables
[root@localhost ~] systemctl enable iptables
Created symlink from /etc/systemd/system/basic.target.wants/iptables.service to /usr/lib/systemd/system/iptables.service.
[root@localhost ~] iptables-save > /etc/sysconfig/iptables    #保存配置
[root@localhost ~] iptables-restore < /etc/sysconfig/iptables    #重载配置
```

七、编写防火墙规则

1、控制类型

- **ACCEPT：允许通过**
- **DROP：直接丢弃，不给出任何回应**
- **REJECT：拒绝通过，必要时会给出提示**
- **LOG：记录日志信息，然后传给下一条规则继续匹配**

2、添加、查看、删除规则

1）添加规则

- **-A：在链的末尾追加一条规则（append）**
- **-I：在链的开头（或指定序号）插入一条规则（insert）**

```
[root@localhost ~] iptables -t filter -A INPUT -p tcp -j ACCEPT

[root@localhost ~] iptables -I INPUT -p udp -j ACCEPT
```

```
[root@localhost ~] iptables -I INPUT 2 -p icmp -j ACCEPT
```

2）**查看规则列表**

- **-L：列出所有的规则条目**
- **-n：以数字形式显示地址、端口等信息**
- **-v：以更详细的方式显示规则信息**
- **--line-numbers：查看规则时，显示规则序号**

```
[root@localhost ~] iptables -L --line-number
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

1    ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
2    ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
3    ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
```

3）删除清空规则

- **-D：删除链内指定序号（或内容）的一条规则**
- **-F：清空指定表中所有的规则（默认是filter）**

```
[root@localhost ~] iptables -t filter -D INPUT 3
[root@localhost ~] iptables -nL INPUT
ChainINPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp -- 0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp --  0.0.0.0/0            0.0.0.0/0
[root@localhost ~] iptables -F
[root@localhost ~] iptables -t nat -F
[root@localhost ~] iptables -t mangle -F
[root@localhost ~] iptables -t raw -F
```

3、设置默认策略

- **-P：为指定的链设置默认规则**

> 默认策略仅可以设置：ACCEPT、DROP

```
[root@localhost ~] iptables -t filter -P FORWARD DROP
[root@localhost ~] iptables -P OUTPUT ACCEPT
```

4、常用管理选项汇总

类别	选项	用途
添加新的规则	-A	在链的末尾追加一条规则
	-I	在链的开头（或指定序号）插入一条规则
查看规则列表	-L	列出所有的规则条目
	-n	以数字形式显示地址、端口等信息
	-v	以更详细的方式显示规则信息
	--line-numbers	查看规则时，显示规则的序号
删除、清空规则	-D	删除链内指定序号（或内容）的一条规则
	-F	清空指定表中所有的规则（默认是filter）
设置默认策略	-P	为指定的链设置默认规则

5、规则的匹配条件

- **常见的通用匹配条件**
  - **协议匹配：-p 协议名**
  - **地址匹配：-s 源地址、-d 目的地址**
  - **接口匹配：-i 入站网卡、-o 出站网卡**

```
[root@localhost ~] iptables -I INPUT -p icmp -j DROP
[root@localhost ~] iptables -A FORWARD ! -p icmp -j ACCEPT
[root@localhost ~] iptables -A FORWARD -s 192.168.1.11 -j REJECT
[root@localhost ~] iptables -I INPUT -s 10.20.30.0/24 -j DROP
[root@localhost ~] iptables -A INPUT -i ens33 -s 192.168.0.0/16 -j DROP
[root@localhost ~] iptables -A INPUT -i ens33 -s 10.0.0.0/8 -j DROP
[root@localhost ~] iptables -A INPUT -i ens33 -s 172.16.0.0/12 -j DROP
```

- **常用的隐含匹配条件**
  - **端口匹配：--sport 源端口、--dport 目的端口**
  - **TCP标记匹配：--tcp-flags 检查范围 被设置的标记**

- **ICMP类型匹配：--icmp-type ICMP类型**

> ICMP 消息类型：
>
> - 8 代表 Echo Request（回显请求），也就是我们通常使用 ping 命令时发送的请求包。
> - 0 代表 Echo Reply（回显应答），即对 ping 请求的响应包。
> - 3 代表 Destination Unreachable（目的不可达）消息，当数据包无法到达目的地时，路由器或目标主机可能会发送此类消息。

```
[root@localhost ~] iptables -A FORWARD -s 192.168.4.0/24 -p udp --dport 53 -j ACCEPT
[root@localhost ~] iptables -A INPUT -p tcp --dport 20:21 -j ACCEPT
[root@localhost ~] iptables -A INPUT -p icmp --icmp-type 8 -j DROP
[root@localhost ~] iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT
[root@localhost ~] iptables -A INPUT -p icmp --icmp-type 3 -j ACCEPT
[root@localhost ~] iptables -A INPUT -p icmp -j DROP
```

- **常用的显式匹配条件**
  - **多端口匹配：**
    - **-m multiport --sport 源端口列表**
    - **-m multiport --dport 目的端口列表**
  - **IP范围匹配：-m iprange --src-range | --dst-range IP范围**
  - **MAC地址匹配：-m mac --mac-source MAC地址**
  - **状态匹配：-m state --state 连接状态**

```
[root@localhost ~] iptables -A INPUT -p tcp -m multiport --dport 25,80,110,143 -j ACCEPT
[root@localhost ~] iptables -A FORWARD -p tcp -m iprange --src-range 192.168.4.21-192.168.4.28 -j ACCEPT
[root@localhost ~] iptables -A INPUT -m mac --mac-source 00:0c:29:c0:55:3f -j DROP
[root@localhost ~] iptables -I INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
```

类别	条件类型	用法
通用匹配	协议匹配	-p 协议名
	地址匹配	-s 源地址、-d 目的地址
	接口匹配	-i 入站网卡、-o 出站网卡
隐含匹配	端口匹配	--sport 源端口、--dport 目的端口
	TCP标记匹配	--tcp-flags 检查范围 被设置的标记
	ICMP类型匹配	--icmp-type ICMP类型
显式匹配	多端口匹配	-m multiport --sport | --dport端口列表
	IP范围匹配	-m iprange --src-range IP范围
	MAC地址匹配	-m mac --mac-source MAC地址
	状态匹配	-m state --state 连接状态

八、iptables filter表应用实例

1、安装ssh、vsftpd、httpd服务

2、清空所有规则，并且将filter表中的INPUT链默认改为DROP，仅允许ssh、vsftpd、httpd服务通行

```
[root@localhost ~] yum -y install vsftpd httpd
[root@localhost ~] systemctl start vsftpd
[root@localhost ~] systemctl start httpd
[root@localhost ~] netstat -anpt | grep httpd
tcp6       0      0 :::80                   :::*                    LISTEN
1400/httpd
[root@localhost ~] netstat -anpt | grep vsftpd
tcp6       0      0 :::21                   :::*                    LISTEN
1393/vsftpd

[root@localhost ~] iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
```

```
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
```

3、问题一：本机无法访问其他主机以及本机

1）通过添加状态模块完成数据包之间的响应状态

- NEW：表示一个新的连接请求，是每一条连接中的第一个数据包所处的状态，与具体使用的协议无关。
  - 例如，当你在浏览器中输入一个网址并按下回车键，此时你的计算机向服务器发送的第一个请求数据包就处于 NEW 状态，它代表着一个全新的网络连接开始被建立。
- ESTABLISHED：当使用 TCP 或 UDP 协议时，如果主机发送的第一个数据包成功通过防火墙，并且与目标主机之间完成了连接的建立过程（如 TCP 的三次握手），那么后续该主机发出和接收到的数据包都处于 ESTABLISHED 状态。只要发送并接到应答，一个数据包的状态就会从 NEW 变为 ESTABLISHED，并且这个状态会持续匹配这个连接后续的数据包。
  - 例如，在你成功打开一个网页后，浏览器与网页服务器之间后续用于传输网页内容、图片等数据的数据包都处于 ESTABLISHED 状态。
- RELATED状态：当一个数据包与处于 ESTABLISHED 状态的连接有关系时，就会被认为是 RELATED 状态。也就是说，一个连接要处于 RELATED 状态，首先必须有一个与之相关的 ESTABLISHED 连接。
  - 例如，当你通过 FTP 协议下载文件时，FTP 控制连接处于 ESTABLISHED 状态，而用于传输文件数据的连接则处于 RELATED 状态，因为它与控制连接是相关联的，是为了完成整个 FTP 下载任务而建立的
- INVALID状态：表示数据包的状态不能被识别属于哪个连接状态，或者与任何已建立的连接都没有关系。这种数据包通常是不合法的或者是异常的，一般会被防火墙拒绝。
  - 例如，当防火墙收到一个没有正确头部信息或者来源不明的数据包时，它可能会被标记为 INVALID 状态并被丢弃，以防止潜在的安全威胁。

```
[root@localhost ~] iptables -I INPUT -m state --state ESTABLISHED -j ACCEPT
```

4、 问题二：ftp客户端访问ftp服务器连接成功，但是无法获取数据

> --syn: 指定要匹配的标志位SYN，即TCP连接建立时发送的第一个数据包的标志

```
[root@localhost ~] iptables -F
[root@localhost ~] iptables -A INPUT -m state --state ESTABLISHED,RELATED -j
 ACCEPT
[root@localhost ~] iptables -A INPUT -i lo -j ACCEPT
[root@localhost ~] iptables -A INPUT -s 192.168.2.0/24 -j ACCEPT #允许内网任何访问
[root@localhost ~] iptables -A INPUT -p tcp --syn --dport 80 -j ACCEPT    #允许来自目标端口为80的TCP连接建立请求的数据包通过防火墙。
[root@localhost ~] iptables -A INPUT -p tcp --syn --dport 22 -j ACCEPT
[root@localhost ~] iptables -A INPUT -p tcp --syn --dport 21 -j ACCEPT
[root@localhost ~] iptables -A INPUT -j REJECT
```

```
[root@localhost ~] iptables-save > /etc/sysconfig/iptables
[root@localhost ~] modprobe nf_conntrack_ftp      #临时方法，添加连接追踪模块
[root@localhost ~] vim /etc/sysconfig/iptables-config     #修改配置文件，永久性生效
IPTABLES_MODULES="nf_conntrack_ftp"
```

小结：

- Linux防火墙默认包括哪几个表、哪几种链？
- 对于转发的数据包，会经过哪几种链的处理？
- 在同一个规则链内，规则匹配的特点是什么？

上机作业

一、部署服务

1、vsftpd、httpd、sshd

二、通过修改iptables规则限制服务的通信

1、除了这三个服务外其他所有流量均被拒绝

2、仅有一个客户机可以访问到这三个服务

3、将所有的条目进行保存

三、验证

1、查看iptables的规则条目　　图一

2、客户机和iptables服务器相互通信验证　　图二 —— 图四（httpd、vsftpd、sshd）

3、清空条目，然后重新加载