Netfilter & iptables 原理

Linux爱好者 2025年03月24日 19:21 浙江

以下文章来源于Linux内核那些事 ，作者songsong001

Linux内核那些事

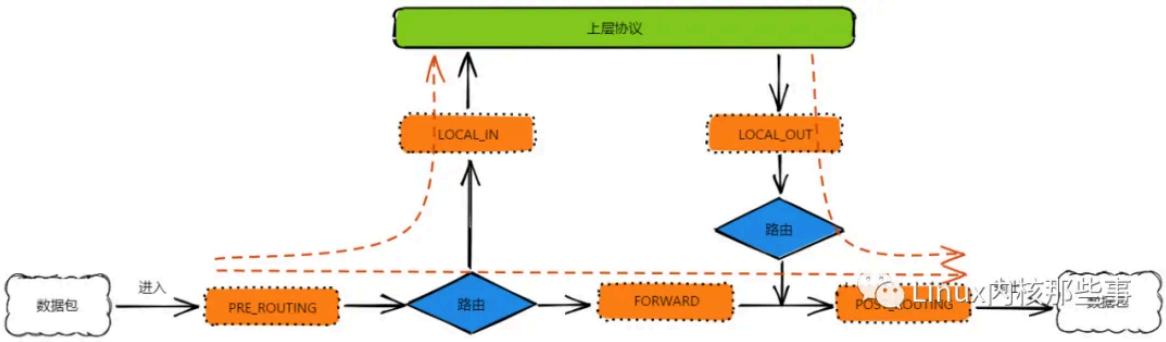
以简单的方式介绍 Linux 内核的原理，以通俗的语言分析 Linux 内核的实现。如果你没...

Netfilter 可能了解的人比较少，但是 iptables 用过 Linux 的都应该知道。本文主要介绍 Netfilter 与 iptables 的原理。

什么是 Netfilter

Netfilter 顾名思义就是网络过滤器，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，有名的 iptables 就是建立在 Netfilter 之上。

Netfilter 通过向内核协议栈中不同的位置注册 钩子函数（Hooks） 来对数据包进行过滤或者修改操作，这些位置称为 挂载点，主要有 5 个：PRE_ROUTING、LOCAL_IN、FORWARD、LOCAL_OUT 和 POST_ROUTING，如下图所示：



这 5 个 挂载点 的意义如下：

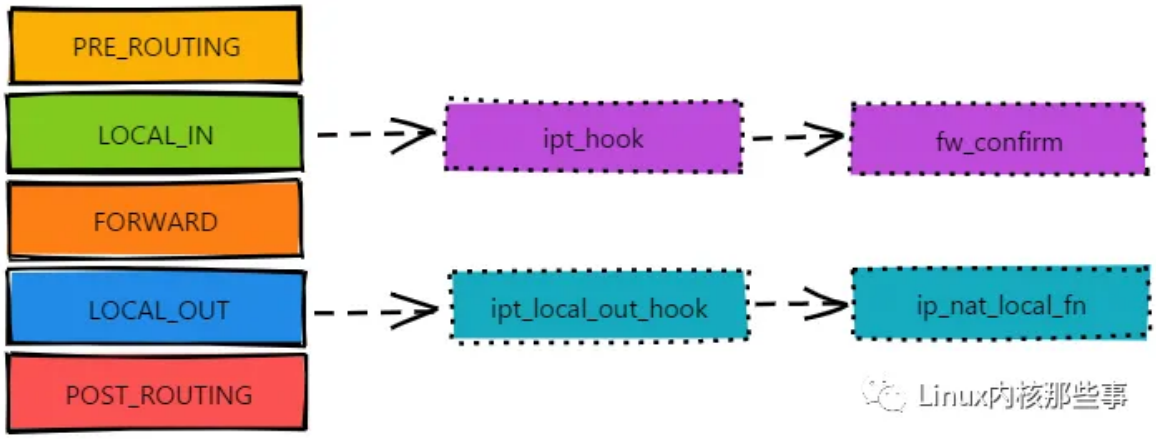
- PRE_ROUTING：路由前。数据包进入IP层后，但还没有对数据包进行路由判定前。
- LOCAL_IN：进入本地。对数据包进行路由判定后，如果数据包是发送给本地的，在上送数据包给上层协议前。
- FORWARD：转发。对数据包进行路由判定后，如果数据包不是发送给本地的，在转发数据包出去前。
- LOCAL_OUT：本地输出。对于输出的数据包，在没有对数据包进行路由判定前。

- POST_ROUTING：路由后。对于输出的数据包，在对数据包进行路由判定后。

从上图可以看出，路由判定是数据流向的关键点。

- 第一个路由判定通过查找输入数据包 IP头部 的目的 IP地址 是否为本机的 IP地址，如果是本机的 IP地址，说明数据是发送给本机的。否则说明数据包是发送给其他主机，经过本机只是进行中转。
- 第二个路由判定根据输出数据包 IP头部 的目的 IP地址 从路由表中查找对应的路由信息，然后根据路由信息获取下一跳主机（或网关）的 IP地址，然后进行数据传输。

通过向这些 挂载点 注册钩子函数，就能够对处于不同阶段的数据包进行过滤或者修改操作。由于钩子函数能够注册多个，所以内核使用链表来保存这些钩子函数，如下图所示：



如上图所示，当数据包进入本地（LOCAL_IN 挂载点）时，就会相继调用 ipt_hook 和 fw_confirm 钩子函数来处理数据包。另外，钩子函数还有优先级，优先级越小越先执行。

正因为挂载点是通过链表来存储钩子函数，所以挂载点又被称为 链，挂载点对应的链名称如下所示：

- LOCAL_IN 挂载点：又称为 INPUT链。
- LOCAL_OUT 挂载点：又称为 OUTPUT链。
- FORWARD 挂载点：又称为 PORWARD链。
- PRE_ROUTING 挂载点：又称为 PREROUTING链。
- POST_ROUTING 挂载点：又称为 POSTOUTING链。

什么是 iptables

iptables 是建立在 Netfilter 之上的数据包过滤器，也就是说，iptables 通过向 Netfilter 的挂载点上注册钩子函数来实现对数据包过滤的。iptables 的实现比较复杂，所以先要慢慢介绍一下它的一些基本概念。

表

从 iptables 这个名字可以看出，它一定包含了 表 这个概念。表 是指一系列规则，可以看成是规则表。iptables 通过把这些规则表挂载在 Netfilter 的不同链上，对进出内核协议栈的数据包进行过滤或者修改操作。

iptables 定义了 4 种表，每种表都有其不同的用途：

1. Filter表

Filter表 用于过滤数据包。是 iptables 的默认表，因此如果你配置规则时没有指定表，那么就默认使用 Filter表，它分别挂载在以下 3 个链上：

- INPUT链
- OUTPUT链
- PORWARD链

2. NAT表

NAT表 用于对数据包的网络地址转换(IP、端口)，它分别挂载在以下 3 个链上：

- PREROUTING链
- POSTOUTING链
- OUTPUT链

3. Mangle表

Mangle表 用于修改数据包的服务类型或TTL，并且可以配置路由实现QOS，它分别挂载在以下 5 个链上：

- PREROUTING链
- INPUT链

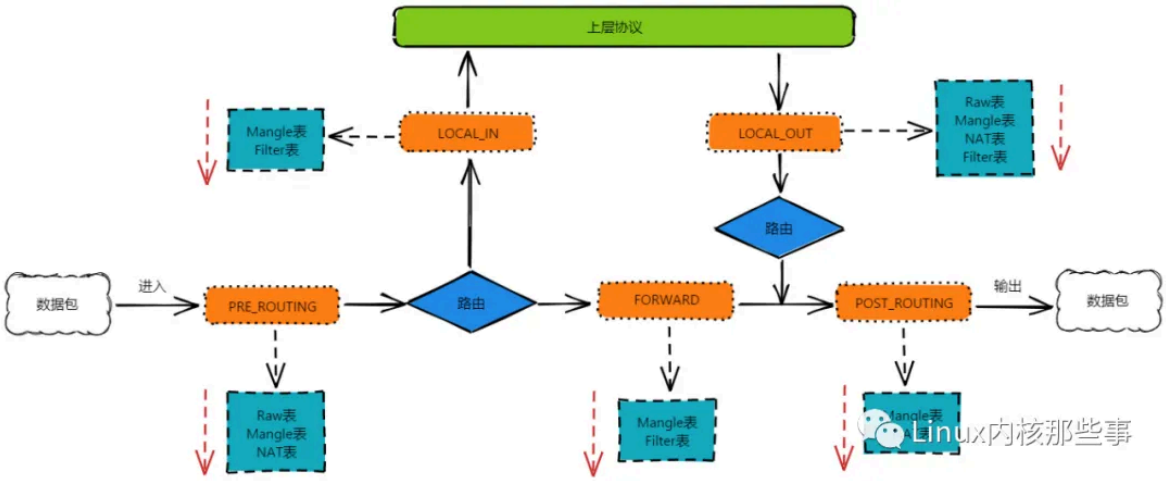
- PORWARD链
- OUTPUT链
- POSTOUTING链

4. Raw表

Raw表 用于判定数据包是否被状态跟踪处理，它分别挂载在以下 2 个链上：

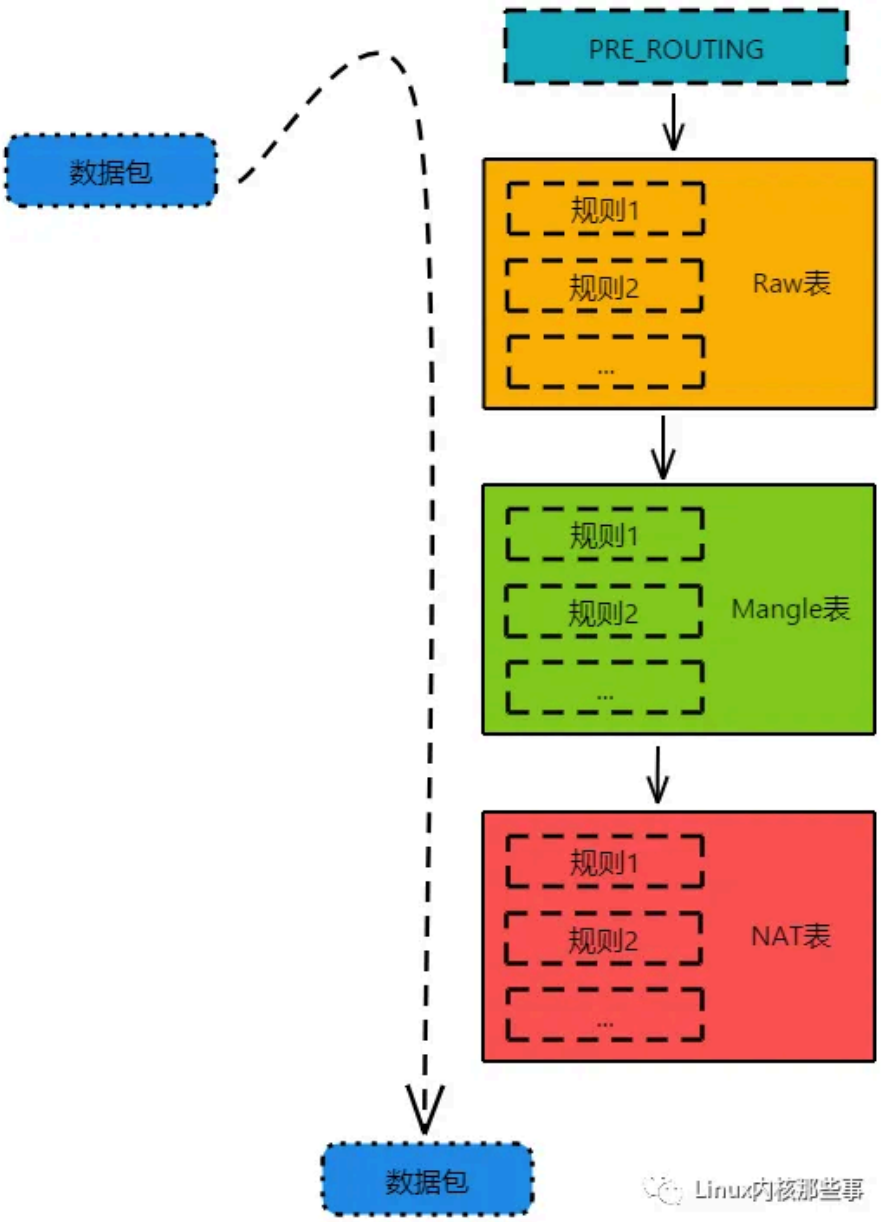
- PREROUTING链
- OUTPUT链

我们通过下图来展示各个表所挂载的链：



上图展示了，数据包从网络中进入到内核协议栈的过程中，要执行的 iptables 规则，如果在执行某条 iptables 规则失败后，会直接把数据包丢弃，不会继续执行下面的规则。

拿其中一个链来看，如下图所示：



也就是说，当数据包从网络中进入到内核协议栈后，在路由判定前会分别执行 Raw表、Mangle 表 和 NAT表 中的规则。如果在执行规则时，某一条规则拒绝了数据包，那么数据包便会被丢弃，从而不会继续执行下面的规则。

添加 iptables 规则

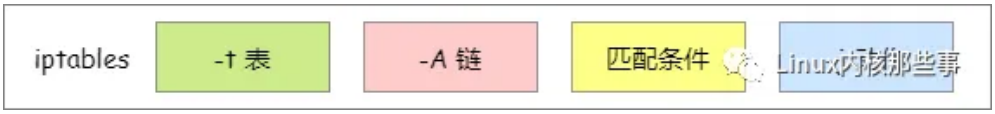
上面介绍了 iptables 的原理，下面主要介绍怎么向 iptables 中添加规则。要向 iptables 中添加规则，可以使用 iptables 命令，其使用格式如下：

```
iptables [选项 参数] ...
```

可选的选项如下：

```
-t <表>：指定要操纵的表；
-A <链>：向规则链中添加条目；
-D <链>：从规则链中删除条目；
-I <链>：向规则链中插入条目；
-R <链>：替换规则链中的条目；
-L：显示规则链中已有的条目；
-F：清楚规则链中已有的条目；
-Z：清空规则链中的数据包计算器和字节计数器；
-N：创建新的用户自定义规则链；
-P：定义规则链中的默认目标；
-h：显示帮助信息；
-p：指定要匹配的数据包协议类型；
-s：指定要匹配的数据包源ip地址；
-j <动作>：指定要进行的动作行为；
-i <网络接口>：指定数据包进入本机的网络接口；
-o <网络接口>：指定数据包要离开本机所使用的网络接口。
--dport <端口>：匹配目标端口号。
--sport <端口>：匹配来源端口号。
```

iptables 规则的选项比较多，一般来说，一条 iptables 规则主要由四个部分组成，如下图所示：



- 第一部分可以通过 -t 选项来指定操作的表，如 filter、nat、mangle 或 raw。
- 第二部分可以通过 -A、-D、-I 或 -R 选项来指定操作的链，如 INPUT、OUTPUT、FORWARD、PREROUTING 或 POSTOUTING。
- 第三部分主要设置规则的匹配条件，如匹配源IP地址或者端口等。
- 第四部分主要设置规则匹配成功后进行的动作，如接收或拒绝等。

第一和第二部分比较简单，我们详细介绍一下第三和第四部分。

匹配条件

匹配条件 分为 基本匹配条件 与 扩展匹配条件，基本匹配条件包括 源IP地址 和 目标IP地址 等，扩展匹配条件包括 源端口 和 目标端口 等。

处理动作

处理动作 是指当匹配条件成功后要进行的一系列操作过程，动作也可以分为 基本动作 和 扩展动作。

此处列出一些常用的动作：

- ACCEPT：允许数据包通过。
- DROP：直接丢弃数据包，不给任何回应信息。
- REJECT：拒绝数据包通过，必要时会给数据发送端一个响应的信息，客户端刚请求就会收到拒绝的信息。
- SNAT：源IP地址转换。
- MASQUERADE：是SNAT的一种特殊形式，适用于动态IP上。
- DNAT：目标IP地址转换。
- REDIRECT：在本机做端口映射。
- LOG：在 /var/log/messages 文件中记录日志信息，然后将数据包传递给下一条规则，也就是说除了记录以外不对数据包做任何其他操作，仍然让下一条规则去匹配。

下面我们通过几个简单的例子来阐明 iptables 命令的使用：

1. 允许本地回环接口(即运行本机访问本机)

```
iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT  # 不指定表名时, 默认
```

2. 允许访问80端口

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

3. 禁止数据转发

```
iptables -A FORWARD -j REJECT
```

4. 禁止IP段访问

```
iptables -I INPUT -s 124.45.0.0/16 -j DROP   # 禁止IP段从123.45.0.1到123.4
```

5. 查看已添加的 iptables 规则

```
iptables -L -n -v
```

总结

本文主要介绍了 Netfilter 与 iptables 的原理，并且还介绍了 iptables 命令的简单使用。由于 iptables 是一个复杂的系统，所以本文不能完整的介绍其所有功能，有兴趣的可以继续查阅其他相关的资料。

推荐阅读 — 点击标题可跳转

1、进程怎么绑定 CPU

2、线上 Linux CPU 100% 故障排查总结

3、Cursor重磅上线Claude Max，工具调用一次0.05美元，充值实测一波