信息安全科技创新

2023年6月

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

基于内核模块的包过滤防火墙

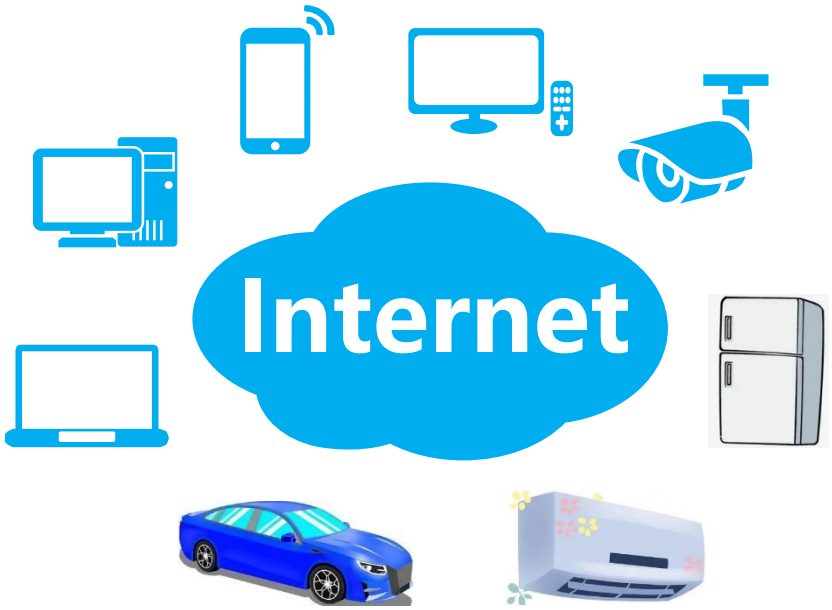
姚立红

2023年6月

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目录 Contents

1.计算机网络相关概念



- **怎么找到对方?**

  IP地址（202.120.2.119,……）

- **与设备上哪个应用通信?**

  端口号

- **怎么交互?**
  - QQ、微信聊天
  - 网页浏览
  - 电子邮件
  - .....

→ 应用层协议
- 专用协议
- 通用协议

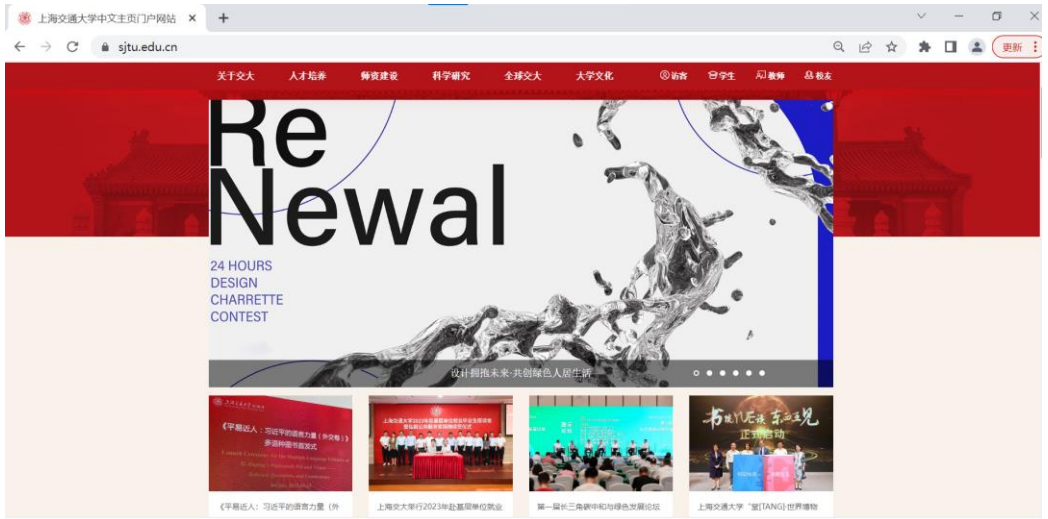
网络分层体系结构

- **应用层**(application layer)：提供应用程序便捷的网络服务调用
- **传输层**(transport layer)：将数据从源端口发送到目的端口（进程到进程）
- **网络层**(network layer)：将数据包跨越网络从源设备发送到目的设备（host to host）
- **数据链路层**(data link layer)：实现相邻（Neighboring）网络实体间的数据传输
- **物理层**(physical layer)：如何在连接各种计算机的传输媒体上传输数据比特流

示例：访问交大主页产生的网络报文（wireshark）

```
无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . : fe80::9d01:1e6f:c01c:4544%10
   IPv4 地址 . . . . . . . . . . . . : 192.168.1.21
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.1.1
```



o.	Time	Source	Destination	Protocol	Length	Info
65	5.498536	192.168.1.21	202.120.2.119	TCP	66	56423 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_
66	5.502776	202.120.2.119	192.168.1.21	TCP	66	443 → 56423 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1460 WS
67	5.503033	192.168.1.21	202.120.2.119	TCP	54	56423 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0

```
> Frame 65: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{392EAD92-FF25-4AB2-A11C-82169DC5A8D5}, id 0
v Ethernet II, Src: IntelCor_b3:d0:f8 (5c:5f:67:b3:d0:f8), Dst: 04:f9:f8:fd:3f:d3 (04:f9:f8:fd:3f:d3)
  > Destination: 04:f9:f8:fd:3f:d3 (04:f9:f8:fd:3f:d3)
  > Source: IntelCor_b3:d0:f8 (5c:5f:67:b3:d0:f8)
    Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 192.168.1.21, Dst: 202.120.2.119
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0x0fe4 (4068)
  > Flags: 0x40, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0x9c33 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.21
    Destination Address: 202.120.2.119
```

```
Transmission Control Protocol, Src Port: 56423, Dst Port: 443, Seq: 0, Len: 0
    Source Port: 56423
    Destination Port: 443
    [Stream index: 14]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3371187163
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window: 64240
    [Calculated window size: 64240]
    Checksum: 0x0284 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
  > [Timestamps]
```

2. 网络防火墙功能

基本概念

- **抽象**：防止网络攻击
- **功能**：按一定的策略管理网络访问，即进行网络访问控制、审计及告警等。

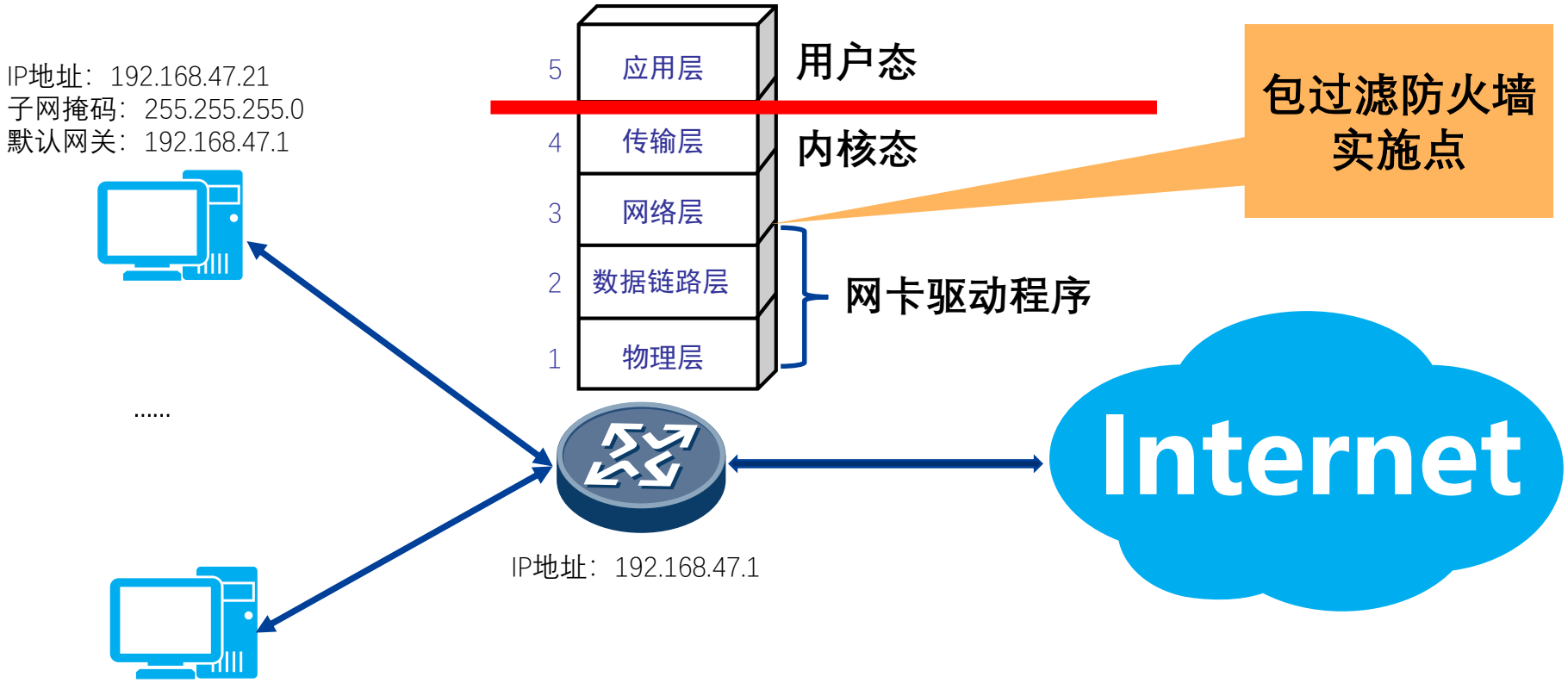
逻辑结构

- **访问决策**：依据访问控制规则，做出当前网络访问是正当的、应该放行的，或不正当的、需阻断的判决。
- **访问实施**：对行程的访问控制判决，对一特定的网络访问进行控制，即放行或阻断该网络访问。

（1）包过滤防火墙

- **包过滤防火墙的安放点**：通常嵌入在连接内外网的路由器（或网关）上实现 。
- **包过滤防火墙的工作原理**：
  - 包过滤操作是在网络层实现。
  - 根据数据包的源IP地址、目标IP地址、协议类型（TCP、UDP、ICMP等）、源端口、目的端口、ICMP消息类型等报头信息及数据包传输方向等信息来，判断是否允许数据包通过。

包过滤防火墙的实施点


IP地址：192.168.47.21
子网掩码：255.255.255.0
默认网关：192.168.47.1
.....
5 应用层
4 传输层
3 网络层
2 数据链路层
1 物理层
用户态
内核态
网卡驱动程序
包过滤防火墙
实施点
IP地址：192.168.47.1
Internet

包过滤防火墙的特点

- **接入方便：**包过滤防火墙工作在**网络层**，与应用层不相关，用户不需要改变客户端的任何应用程序或设置，也无需对内部网络用户进行相关使用培训，因而很容易接入到现存的网络环境中。
- **速度快：**包过滤防火墙工作在网络层，至多分析所对应的传输层协议（即获得端口等相关的属性信息），协议处理比较简单，所以处理包的速度比应用代理防火墙快。
- **实现简单：**包过滤防火墙实现相对简单，甚至可以集成到原有的路由器中。目前很多网络路由器都有IP包过滤功能，它们在逻辑上可以认为是包过滤防火墙。

（2）应用代理防火墙

应用代理防火墙的原理：

- 应用代理防火墙采取的是一种代理机制，它可以为每一种应用服务建立一个专门的代理；
- 内外网之间不直接通信，需先经过应用代理防火墙的审核，审核通过后再由应用代理防火墙代为连接。

——不给内、外网的计算机任何直接会话的机会，从而避免了外部攻击者入侵内部网络

应用代理防火墙的优缺点

- **优点**：实现基于网络会话的连接控制并产生相应的日志记录——安全性好。
- **缺点**
  - 效率低
  - 需要客户端设置

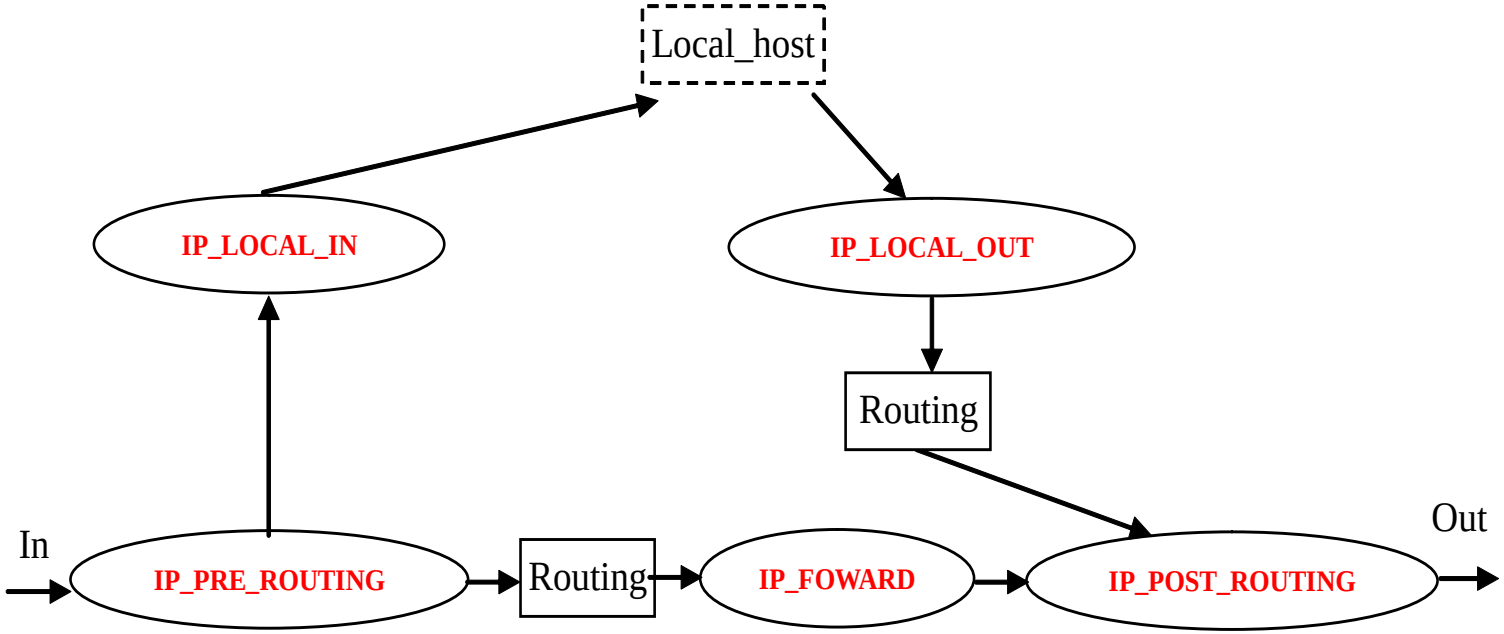
3. Linux的Netfilter机制

Netfilter的基本概念

- Netfilter机制的核心是一个开放式的IP数据包处理框架，该框架对外提供了操纵和处理IP数据的统一接口，编程人员可以利用该接口实现对IP包的控制以及其它新的处理方式。
- 一方面Linux系统自身借助该机制实现一些常见的IP包处理方式（iptables），包括重新实现了其内核包过滤防火墙（称之为Linux内置防火墙），及其它相关的处理功能，如网络地址转换NAT等。
- 另一方面，第三方的软件开发者可以基于Netfilter提供的IP报文处理接口，开发相应的网络工具，包括网络防火墙、NAT、网络审计等。

Netfilter的核心思想

- 在网络IP协议层IP数据包的处理流程中，总结出几个关键点（即钩子点），这些关键点提供了多种可能的IP数据包处理方式和开放接口。
- 安全管理员不但可以配置Netfilter以不同的方式处理IP数据包，也可利用Netfilter所提供的开放接口在IP数据包的协议处理流程中实现新的IP包处理方式。



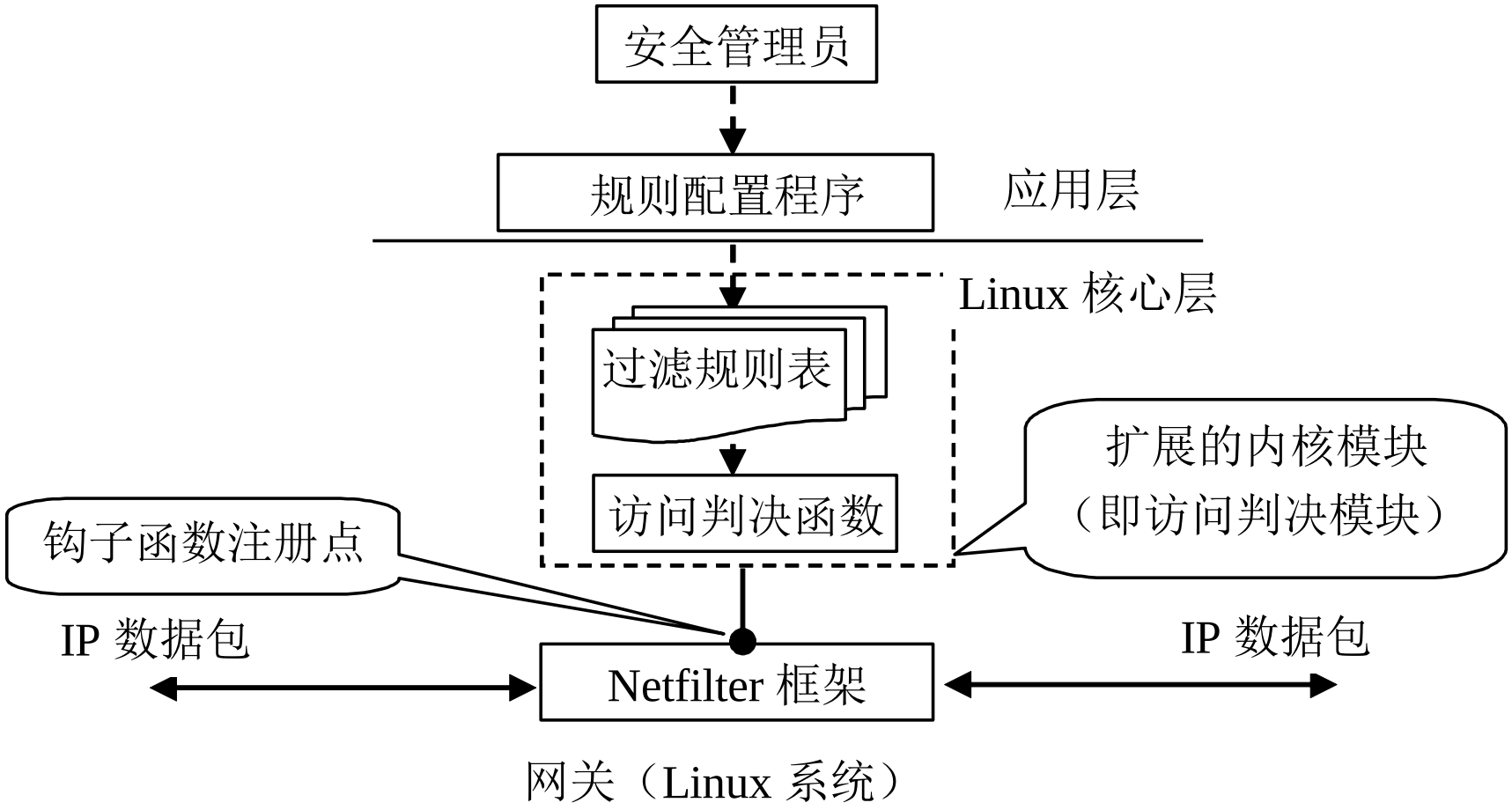
IP数据包的处理流程中的五个钩子点

开放处理方式

- 意味着可在Netfilter机制基础上自己开发新的报文处理方式，Netfilter将IP报文的处理权交给新开发的报文处理方式。
- 开放处理方式包括两种具体方式：
  - **钩子函数方式**：Netfilter机制为每种网络协议(IPv4、IPv6等)定义一套钩子，在数据报流过协议栈的某钩子点，挂接在该钩子上的钩子函数将会被调用。
  - **队列输出方式**：Netfilter提供队列输出功能，在这些关键点将所经过的IP数据包通过一定的方式直接交给应用层，程序员可以在应用层开发应用软件对这些数据包进行完全自主的处理，如丢弃、修改等。

4. 内核模块包过滤防火墙原型

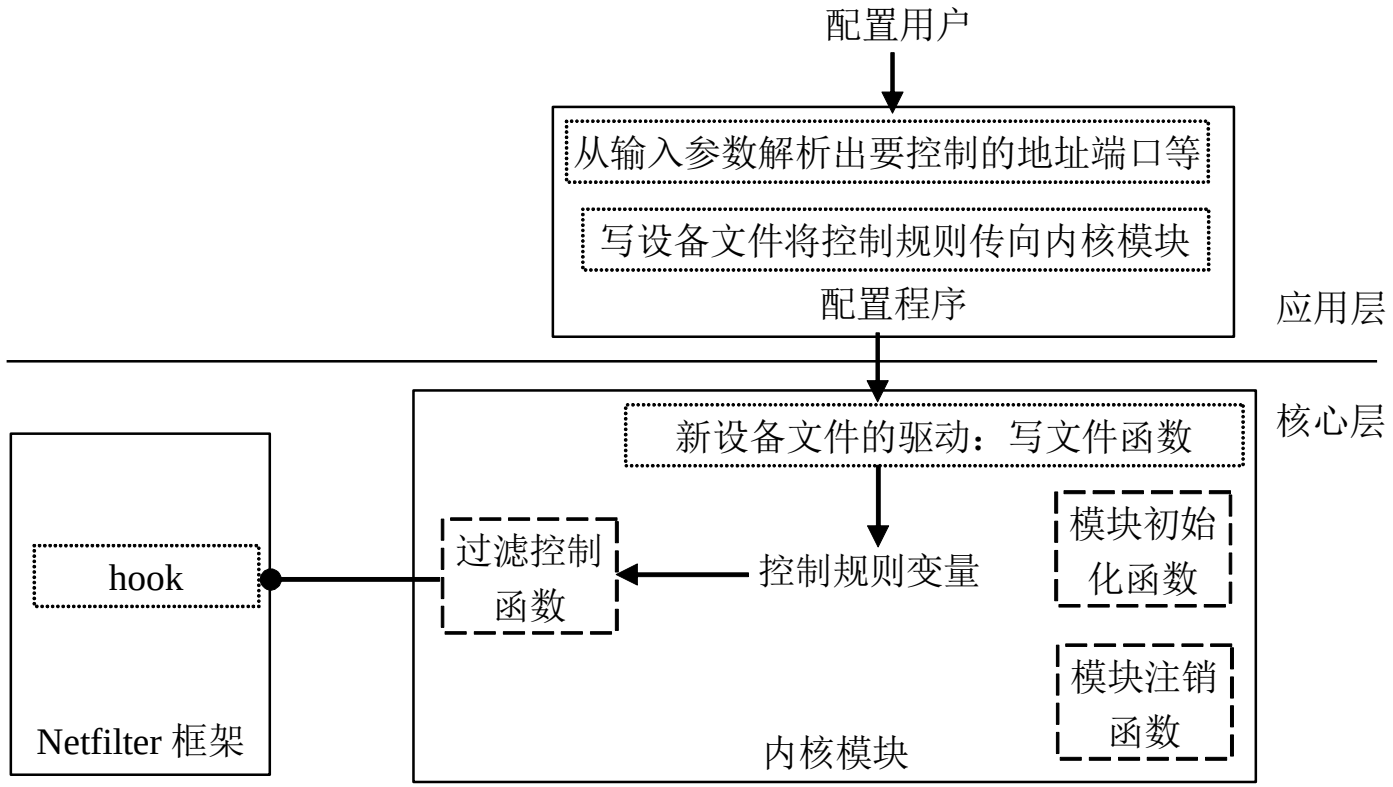
运行原理



实现结构

原型系统分为两个部分独立实现

- **配置程序**：运行在应用层，用来设置启用哪一条包过滤策略和配置相应的参数，包括要控制的IP地址和端口等；
- **Linux内核模块**：运行在内核层，完成包过滤防火墙的功能，该模块借助注册Netfilter 钩子函数的方式来实现对数据包的过滤和控制。



运行方式

- 插入内核模块
- 通过命令行参数，输入要被丢弃的数据包的协议类型，源、目的IP地址以及源、目的端口号

./configure –p protocol –x source_ip –y source_port –m dst_ip –n dst_port

各选项的具体含义如下：

- -p protocol 指明要控制的协议（或网络应用）类型，具体为 tcp、udp、ping三种之一；
- –x source_ip 指明要控制报文的源IP地址；
- –y dst_ip 指明要控制报文的目标IP地址；
- –m source_port 指明要控制报文的源端口地址；
- –n dst_port 指明要控制报文的目标端口；

——若某个选项省略则取其默认值为0，0表示对任意值相匹配，即表示控制的报文覆盖该选项的所有值域。

5. 内核包过滤防火墙扩展开发

背景：原型系统的控制功能简单

- 控制规则简单，只是依据通信双方的IP地址和端口进行检查和控制，实际上包过滤防火墙还可以依据其它要素进行控制。比如，基于时间段进行控制，比如在休息日时间不准从外网访问内网等。
- 以几个简单变量存储相应的控制信息，相当于只能支持一条包过滤规则。一个实用的包过滤防火墙需要同时支持多条包过滤规则。

开发目标：

- **检查和控制要素的扩展**。除实现基于IP地址、端口的检查和控制外，还能使基于时间段、网络接口、ICMP报文的子类型等进行报文的检查和控制。
- **多包过滤规则的扩展**。以表的形式存储包过滤规则，能够支持用户配置多条包过滤规则，使内核模块防火墙能够同时按多条包过滤规则进行报文检查和过滤控制。
- **友好的包过滤规则的配置和管理界面**。支持包过滤的规则导入、导出，添加、编辑、删除等基本功能。

谢谢！