

计算机网络编程

第16章 包过滤防火墙程序设计

信息工程学院 方徽星

fanghuixing@hotmail.com

大纲

- 设计目的
- 相关知识
- 例题分析

1. 设计目的

- 防火墙是网络安全技术中的重要组成部分
- 通过包过滤防火墙程序设计
 - 了解防火墙的基本概念与主要功能
 - 掌握网络层包过滤技术的设计思路与编程方法

2. 相关知识：网络安全的重要性

- 海莲花（ OceanLotus ）是高度组织化的、专业化的境外国家级黑客组织**
- 自2012年4月起针对中国政府的海事机构、海域建设部门、科研院所和航运企业,展开了精密组织的网络攻击**
- 是有国外政府支持的APT(高级持续性威胁)行动**

2. 相关知识：网络安全的重要性

- 2013年3月，中国解放军报报道，美国曾利用“震网”蠕虫病毒攻击伊朗的铀浓缩设备，已经造成伊朗核电站推迟发电**
- 近500万网民、及多个行业的领军企业遭此病毒攻击**

2. 相关知识：网络安全的重要性

- 杜天禹通过植入木马等方式，非法侵入山东省2016年普通高等学校招生考试信息平台网站，窃取2016年山东省高考考生个人信息64万余条，并对外出售牟利**
- 陈文辉等人使用所购的上述信息实施电信诈骗，拨打诈骗电话1万余次，骗取他人钱款20余万元，造成山东省临沂市罗庄区高考考生徐玉玉死亡**

2. 相关知识：网络安全的重要性

- 2017年5月12日，一种名为“Wannacry”的勒索病毒袭击全球150多个国家和地区，影响领域包括政府部门、医疗服务、公共交通、邮政、通信和汽车制造业



2. 相关知识：网络安全的重要性

• 网络安全概念

- 既包括用于解决网络应用中的安全威胁的各种技术或管理手段
- 也包括这些安全威胁本身以及相关活动

网络安全法获高票通过 明确加强个人信息保护

十二届全国人大常委会第二十四次会议11月7日上午经表决通过了《中华人民共和国网络安全法》

2015年6月

十二届全国人大常委会第十五次会议对网络安全法草案进行首次审议

2016年6月

十二届全国人大常委会第二十二次会议对网络安全法草案进行第二次审议

2016年10月31日

网络安全法草案提交十二届全国人大常委会第二十四次会议进行第三次审议

网络安全法的出台先后经过了全国人大常委会的三次审议

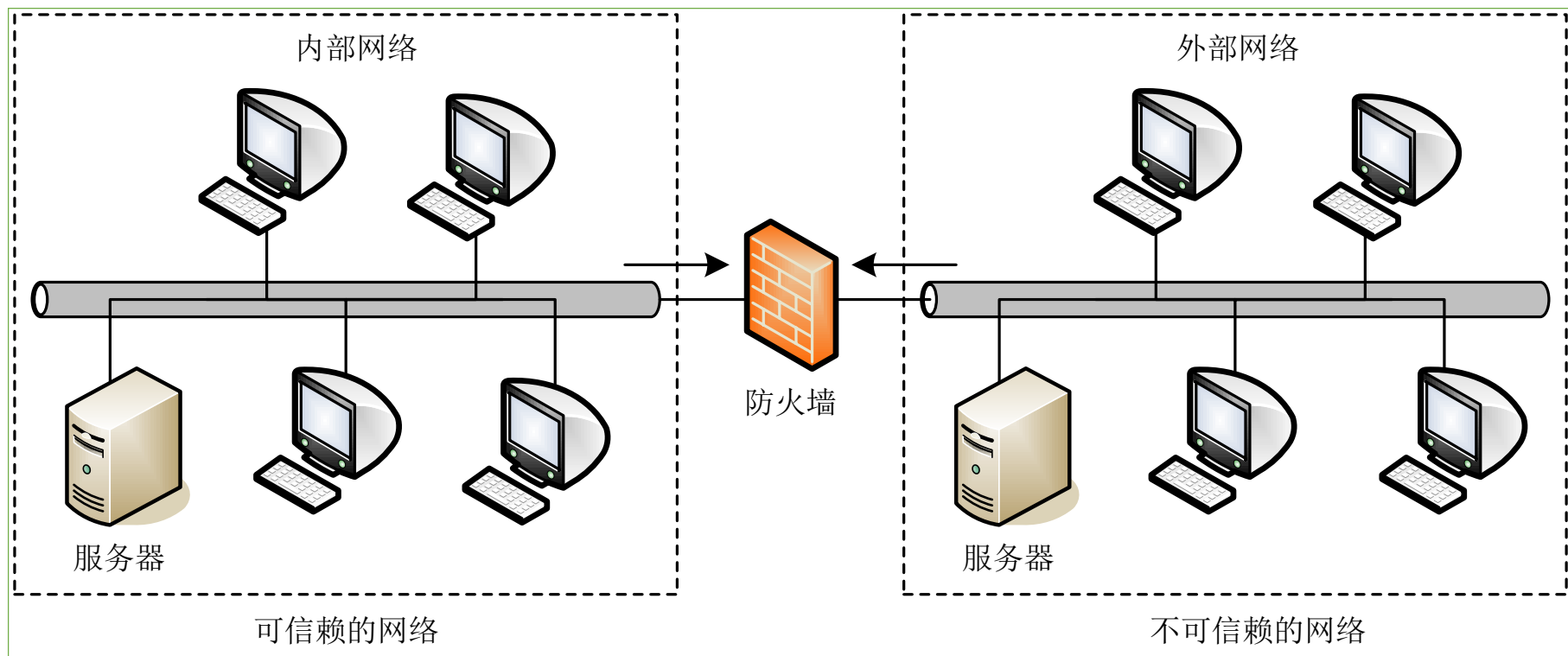
网络安全法共有7章79条
内容上有6方面突出亮点

- 1 明确了网络空间主权的原则
- 2 明确了网络产品和服务提供者的安全义务
- 3 明确了网络运营者的安全义务
- 4 进一步完善了个人信息保护规则
- 5 建立了关键信息基础设施安全保护制度
- 6 确立了关键信息基础设施重要数据跨境传输的规则

该法自2017年6月1日起施行

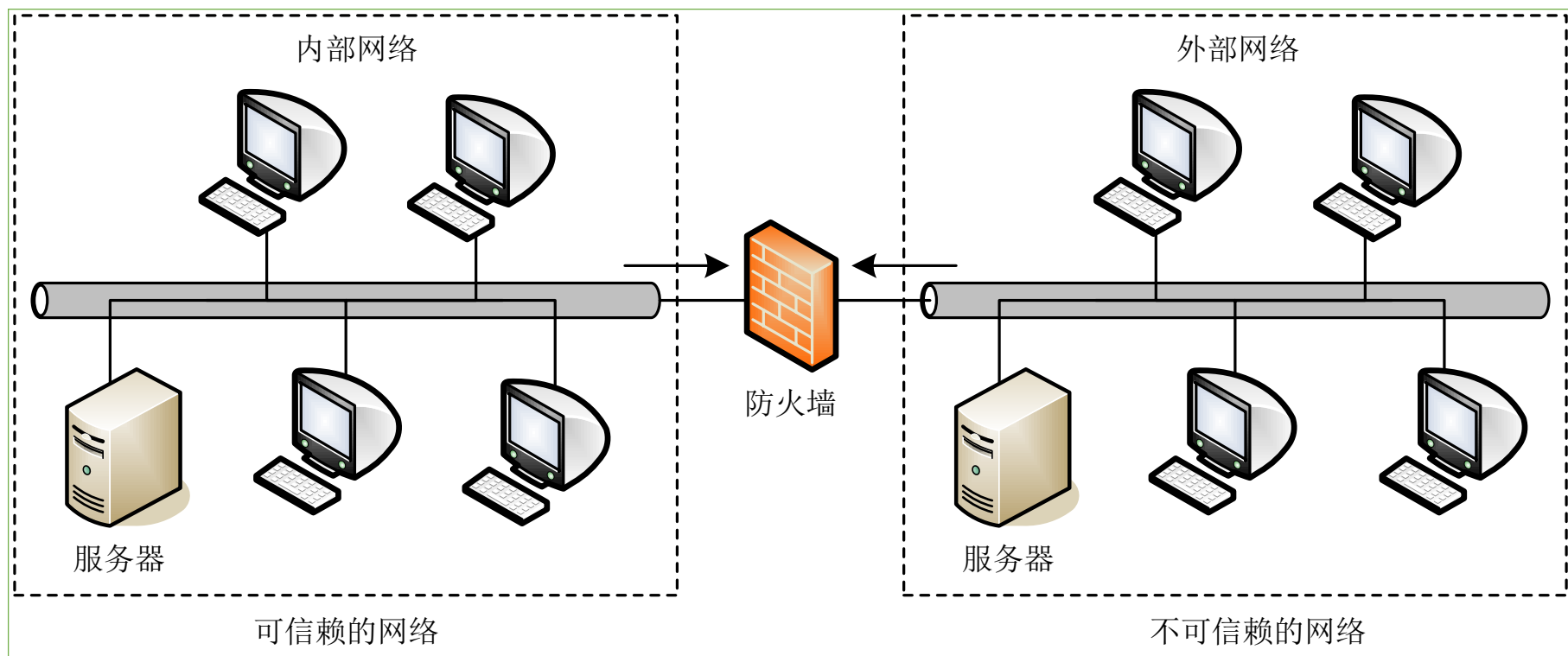
2. 相关知识：防火墙的基本概念

- 防火墙是在计算机网络之间执行控制策略的系统，包括专用的硬件设备与软件系统



2. 相关知识：防火墙的基本概念

- 防火墙的目的是保护内部网络资源不被外部非授权用户使用，防止内部网络受到外部非法用户的攻击



2. 相关知识：防火墙的基本概念

- **防火墙的控制策略主要包括**

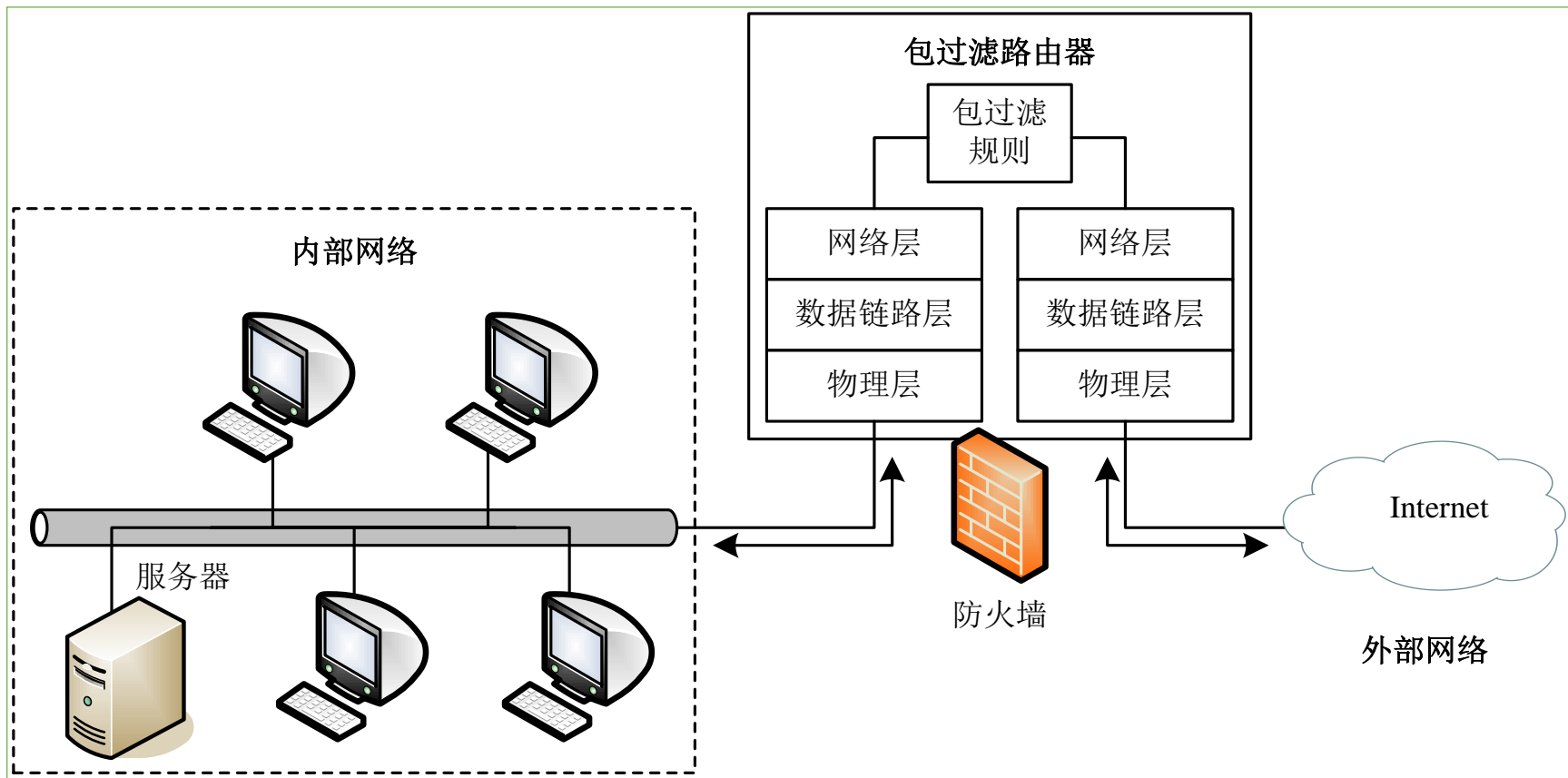
- **服务控制：确定访问的网络服务，FTP、E-mail等**
- **方向控制：确定访问网络服务的方向，内部<->外部**
- **用户控制：确定访问网络服务的（内部）用户**
- **行为控制：确定访问网络的形式，如访问部分信息**

2. 相关知识：防火墙的分类方法

- 根据结构与实现技术的不同，主要分为
 - 包过滤路由器：在**网络层**实现的防火墙系统，建立在**路由技术**基础上
 - 应用级网关：在**应用层**实现的防火墙，建立在**代理技术**的基础上

2. 相关知识：防火墙的分类方法

- **包过滤路由器**：检查IP分组，根据包过滤规则执行相应操作



2. 相关知识：防火墙的分类方法

- 包过滤路由器过滤的内容主要包括：
 - 网络层的头部信息，例如IP地址、优先级与协议类型等
 - 传输层的头部信息，例如端口号与TCP控制标记（SYN、ACK、FIN、RST）等

2. 相关知识：防火墙的分类方法

- 实现包过滤的关键是指定相应的包过滤规则

规则	源地址	目的地址	传输层协议	端口号	操作
1	任意	202.113.1.2	TCP	80	允许
2	任意	202.113.1.3	TCP	25	允许
3	任意	202.113.1.4	UDP	53	允许
4	任意	任意其他地址	任意	任意	丢弃

2. 相关知识：防火墙的分类方法

- **包过滤路由器主要优点**

- **结构简单**
- **便于管理**
- **造价低廉**

**包过滤针对的是网络层与传输层的数据
不需要客户机和服务器程序做任何修改**

2. 相关知识：防火墙的分类方法

- **包过滤路由器的主要缺点**
 - **规则配置比较困难，需熟悉各种协议及相关特征**
 - **包过滤建立在IP地址或端口号基础上，只能控制到主机级而不能达到用户级**
 - **不能阻止某些类型的网络攻击**
 - **DDoS：常通过向服务器提交大量请求，使服务器超负荷，从而拒绝服务**
 - **IP欺骗攻击：伪造源IP地址,以便冒充其他系统或发件人的身份**

2. 相关知识：防火墙的分类方法

- **应用级网关：在应用层实现的防火墙，通常是一台具有应用程序访问控制功能的主机**
 - **处理的数据包是应用层数据**
 - **核心技术是应用访问控制规则**
 - **如果应用级网关允许某个数据包通过，则将数据包转发给相应主机**
 - **如果拒绝某个数据包通过，则丢弃数据包并通知相应的发送方**

2. 相关知识：防火墙系统结构

- **防火墙通常由包过滤路由器与应用级网关作为基本单元，采用多级结构与多种组合方式**
 - **包过滤路由器通常用字符S来表示**
 - **堡垒主机是指一台运行应用级网关软件的主机，通常用B来表示**
 - **单接口堡垒主机：有一个网络接口，可连接一个子网，通常用B1来表示**
 - **双接口堡垒主机：有两个网络接口，可连接两个子网，通常用**

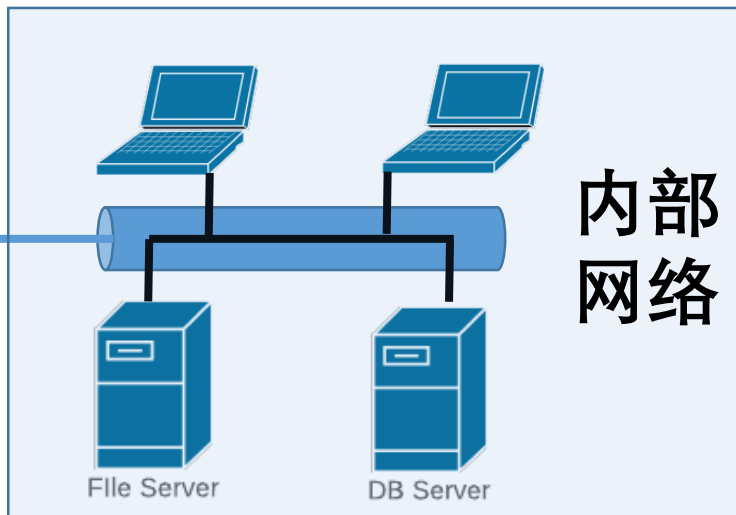
2. 相关知识：防火墙系统结构

- **根据主机面向的服务对象方面的差异，内部网络主机可分为**
 - **普通主机，如内部用户的工作主机**
 - **对内服务器，如文件服务器、数据库服务器**
 - **对外服务器，如Web服务器、FTP服务器**

内防火墙



Firewall



内部网络

外部用户

非军事区



Firewall



Internet



外防火墙



Web Server



FTP Server

**非军事区：将对外服务器
放置在该区域**

3. 例题分析：设计要求

- 编写包过滤程序，捕获与分析网络中的IP包，并且将符合条件的IP包信息显示在控制台上
- 只设置两条简单的包过滤规则，并且不丢弃符合条件的IP包
- 两条包过滤规则
 - 目的地址为192.168.0.1，协议类型为UDP：允许通过
 - 源地址为192.168.0.1，协议类型为UDP：拒绝通过

3. 例题分析：设计要求

- 具体要求

- 命令行程序

PackFilter packet_sum

- 将部分字段内容显示在控制台上

源IP地址： xx. xx. xx. xx

目的IP地址： xx. xx. xx. xx

协议类型： UDP

操作类型： 允许或拒绝

...

3. 例题分析：关键问题

- 初始化Socket结构

- 为了通过网卡截获网络中传输的IP包，需创建原始套接字
- 调用setsockopt，设置套接字选项，允许处理IP头部

套接字句柄 参数层次：IP协议 参数值占用的字节数

```
setsockopt(sock, IPPROTO_IP,  
           IP_HDRINCL, (char *) &flag, sizeof(flag));
```

需要设置的参数 参数值：true
表示用户自己
处理IPv4头部

The diagram illustrates the components of the `setsockopt` function call. It features five blue arrows pointing from descriptive text to specific parts of the code:
1. An arrow from '套接字句柄' (Socket handle) points to the `sock` parameter.
2. An arrow from '参数层次：IP协议' (Parameter level: IP protocol) points to the `IPPROTO_IP` parameter.
3. An arrow from '参数值占用的字节数' (Number of bytes occupied by the parameter value) points to the `sizeof(flag)` parameter.
4. An arrow from '需要设置的参数' (Parameter to be set) points to the `IP_HDRINCL` option.
5. An arrow from '参数值：true 表示用户自己处理IPv4头部' (Parameter value: true indicates user handles IPv4 header) points to the `&flag` argument.

3. 例题分析：关键问题

```
//创建原始套接字
SOCKET sock = socket(AF_INET, SOCK_RAW, IPPROTO_IP);
BOOL flag = true;
setsockopt(sock, IPPROTO_IP, IP_HDRINCL,
           (char *)&flag, sizeof(flag));
//填充套接字地址
sockaddr_in host_addr;
host_addr.sin_family = AF_INET;
host_addr.sin_port = htons(6000);
host_addr.sin_addr = *(in_addr *)
                    pHostIP->h_addr_list[0];
bind(sock, (PSOCKADDR)&host_addr, sizeof(host_addr));
```

3. 例题分析：关键问题

- 接收所有IP包
 - 想要截获经过网卡的所有IP包，需要调用 `WSAIoctl()` 函数将网卡设置为混杂模式
 - 当接收IP包中的协议类型与原始套接字匹配，IP包内容被复制到套接字缓冲区

```
DWORD dwBufferLen[10];
DWORD dwBufferInLen = 1;
DWORD dwBytesReturned = 0;
WSAIoctl(sock, IO_RCVALL, &dwBufferInLen,
          sizeof(dwBufferInLen), &dwBufferLen,
          sizeof(dwBufferLen), &dwBytesReturned,
          NULL, NULL);
//接收所有IP包
while (i < packsum)
{
    recv(sock, buffer, 65535, 0);
    ...
}
```

3. 例题分析：关键问题

- **检查包过滤规则**

- **接收到IP包后，需根据过滤规则分析IP头部字段**
- **可利用结构体来定义包过滤规则，并填充**
- **根据每条规则检查IP头部字段，执行相应操作**

//包过滤规则结构体

```
typedef struct
```

```
{
```

```
    char SourceAddr[16]; //源IP地址
```

```
    char DestinAddr[16]; //目的IP地址
```

```
    unsigned short SourcePort; //源端口号
```

```
    unsigned short DestinPort; //目的端口号
```

```
    unsigned char Protocol; //协议类型
```

```
    bool Operation; //操作类型
```

```
}filter_table;
```

```
//填写包过滤规则（2项）
filter_table filter[2];
//第一条
memset(filter[0].SourceAddr, 0, 16);

//设置IP地址
memcpy(filter[0].SourceAddr,
        "192.168.0.1", strlen("192.168.0.1"));

//设置协议类型
filter[0].Protocol = IPPROTO_UDP;
filter[0].Operation = REJECT_OPT;
```

```
//第二条
```

```
memset(filter[1].DestinAddr, 0, 16);
```

```
//设置IP地址
```

```
memcpy(filter[1].DestinAddr,  
        "192.168.0.1", strlen("192.168.0.1"));
```

```
//设置协议类型
```

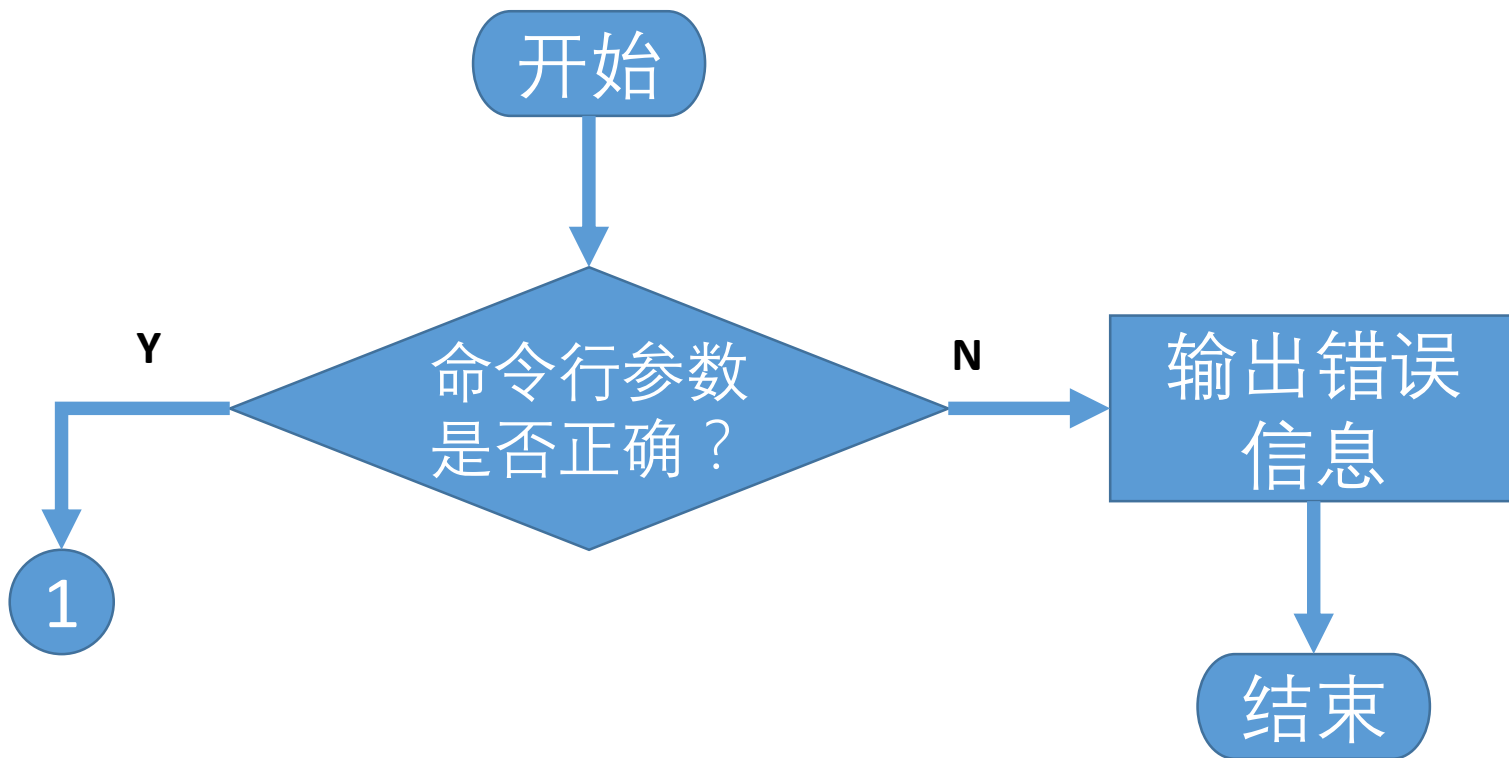
```
filter[1].Protocol = IPPROTO_UDP;
```

```
filter[1].Operation = PERMIT_OPT;
```

```
//检验包过滤规则1
//比较IP包中的源IP地址与规则中的源IP地址
if (strcmp(source_ip, filter[0].SourceAddr) == 0)
{
    //源IP地址相同
    if (ip.Protocol == filter[0].Protocol)
    {
        //协议相同
        ...
    }
}
```

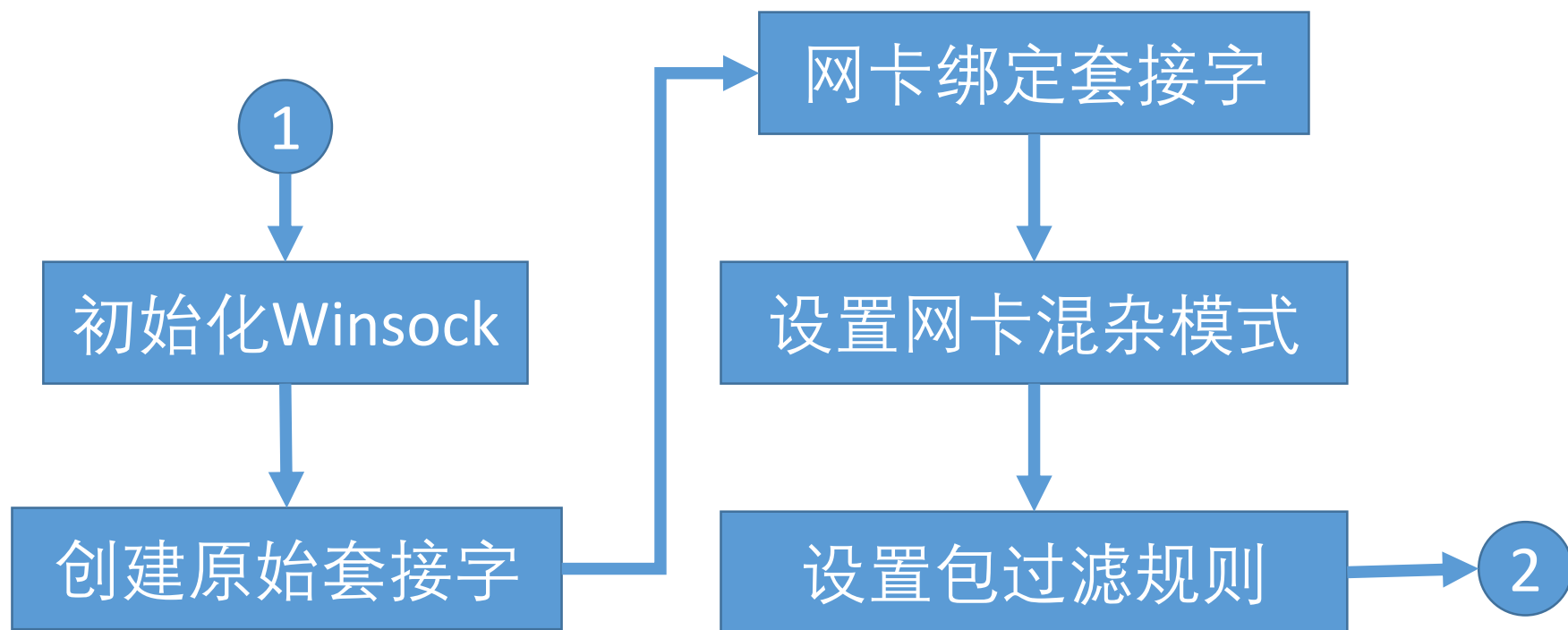

3. 例题分析：关键问题

- 程序流程图



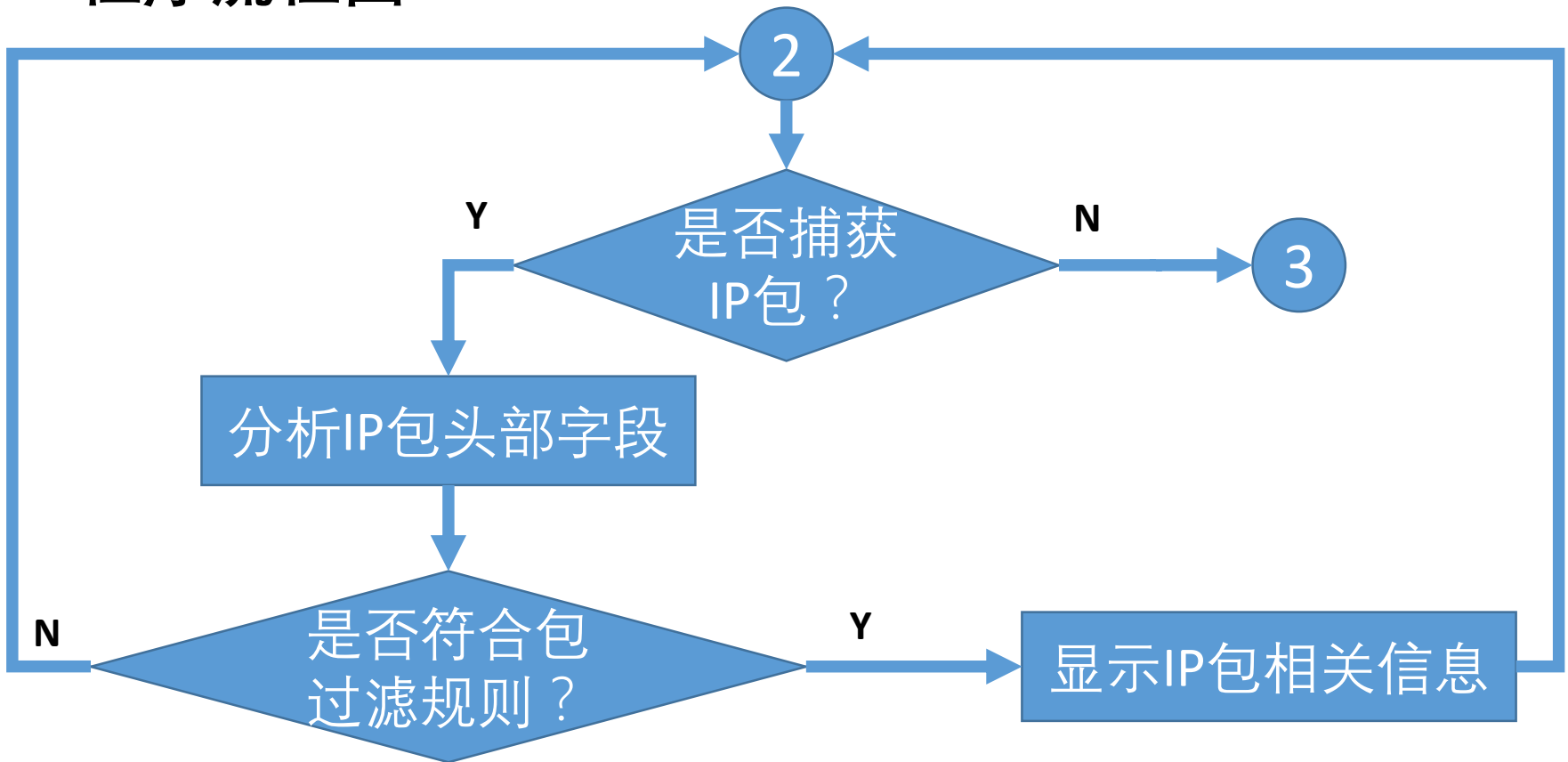
3. 例题分析：关键问题

- 程序流程图



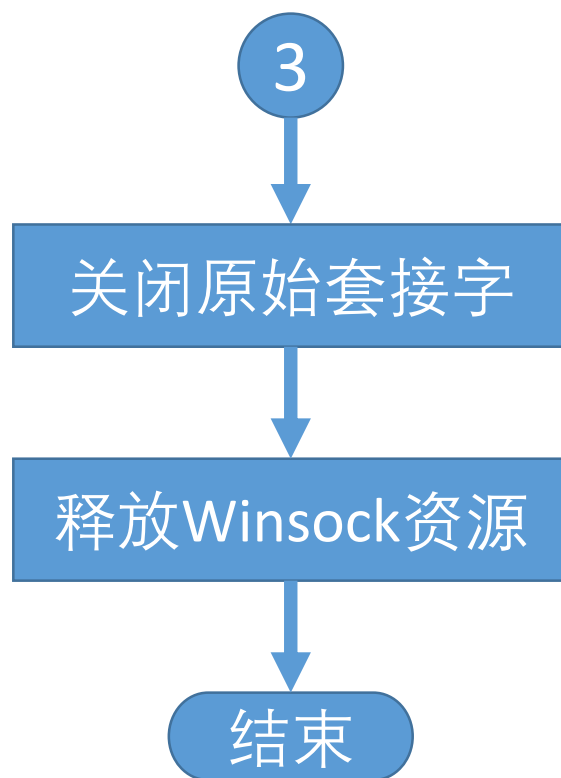
3. 例题分析：关键问题

- 程序流程图

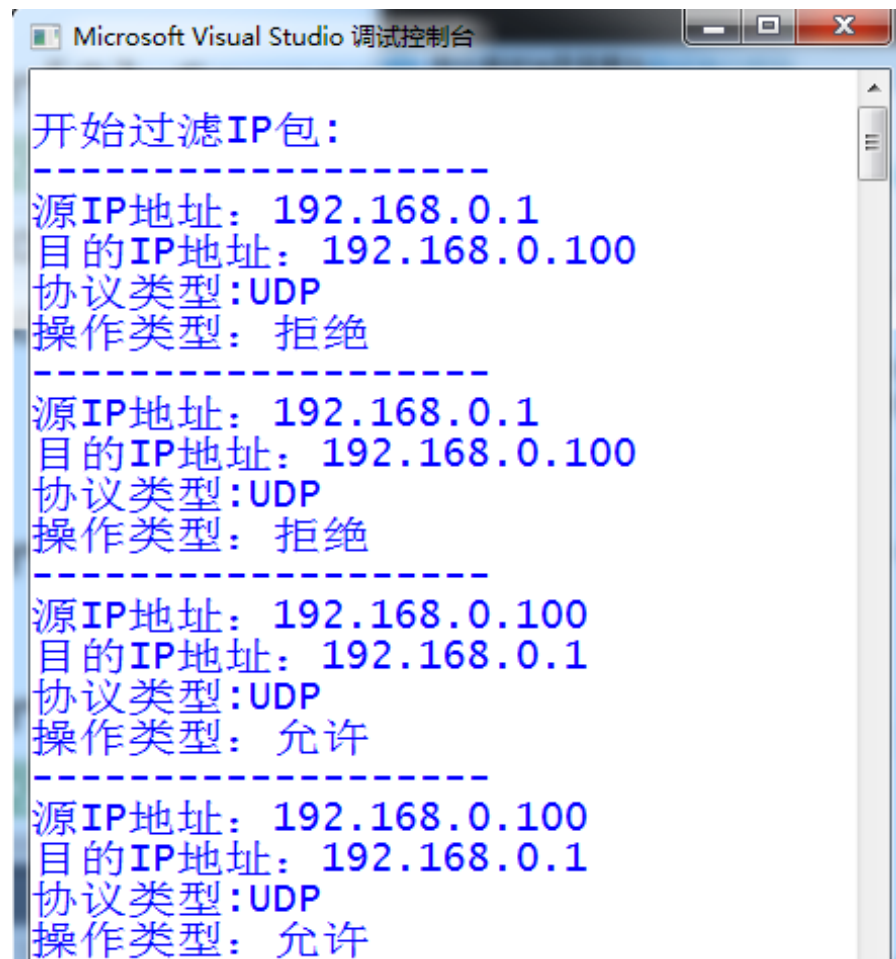


3. 例题分析：关键问题

- 程序流程图



程序演示



```
Microsoft Visual Studio 调试控制台

开始过滤IP包：
-----
源IP地址： 192.168.0.1
目的IP地址： 192.168.0.100
协议类型：UDP
操作类型： 拒绝
-----
源IP地址： 192.168.0.1
目的IP地址： 192.168.0.100
协议类型：UDP
操作类型： 拒绝
-----
源IP地址： 192.168.0.100
目的IP地址： 192.168.0.1
协议类型：UDP
操作类型： 允许
-----
源IP地址： 192.168.0.100
目的IP地址： 192.168.0.1
协议类型：UDP
操作类型： 允许
```

本章小结

- **设计目的**
 - 了解防火墙基本概念与主要功能
 - 掌握网络层包过滤技术的设计思路与编程方法
- **相关知识**
 - 网络安全的重要性
 - 防火墙基本概念
 - 防火墙分类
 - 防火墙系统结构
- **例题分析**
 - 初始化Socket结构、检查包过滤规则、流程图