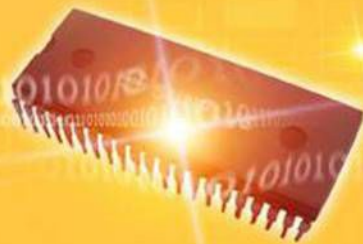


嵌入式系统工程师



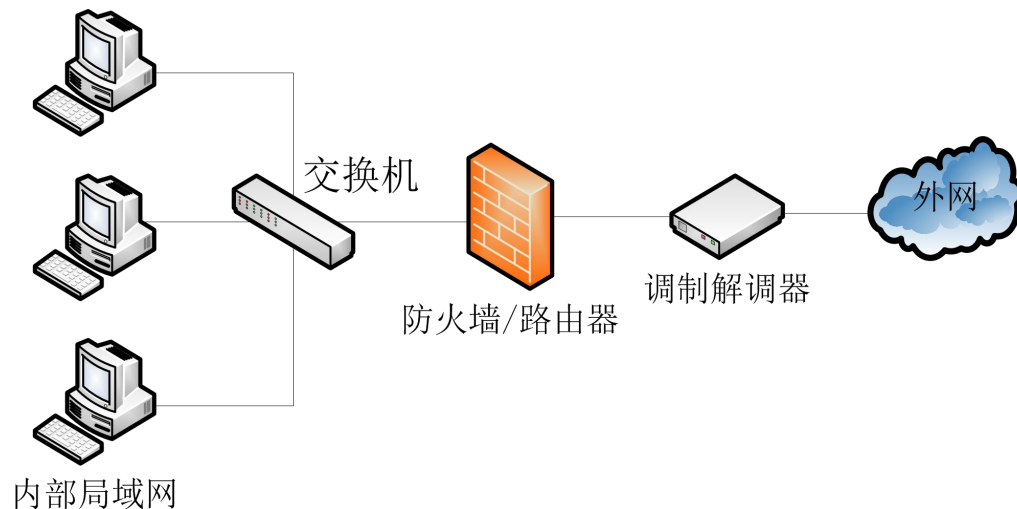
Linux防火墙

- 认识防火墙
- 防火墙的一般网络布线示意
- 防火墙的使用限制
- Linux的数据包过滤软件:iptables
- 设置单机防火墙实例

- 认识防火墙
- 防火墙的一般网络布线示意
- 防火墙的使用限制
- Linux的数据包过滤软件:iptables
- 设置单机防火墙实例

➤ 防火墙的定义

- 防火墙被定义成一个或一组设备，它在网络之间执行访问控制策略



➤ 防火墙的分类

- 硬件防火墙、软件防火墙

➤ 防火墙最重要的任务

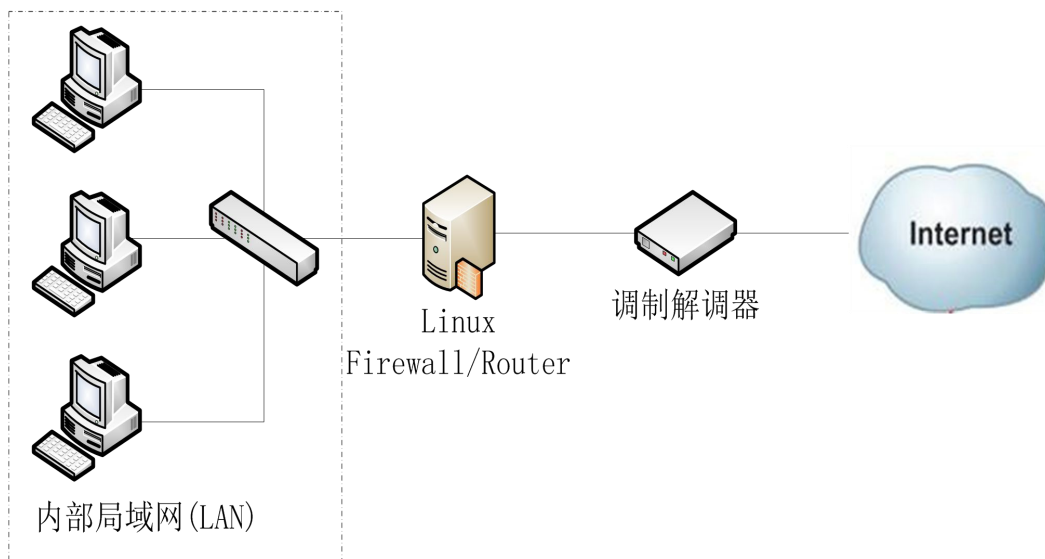
- 切割被信任(如子域)与不被信任(如 Internet)的网段
- 划分出可提供Internet的服务与必须受保护的服务
- 分析出可接受与不可接受的数据包状态

➤ 你需不需要防火墙？

- 理论上需要，但你必须知道系统哪些数据与服务需要保护、针对需要受保护的服务来设置防火墙规则

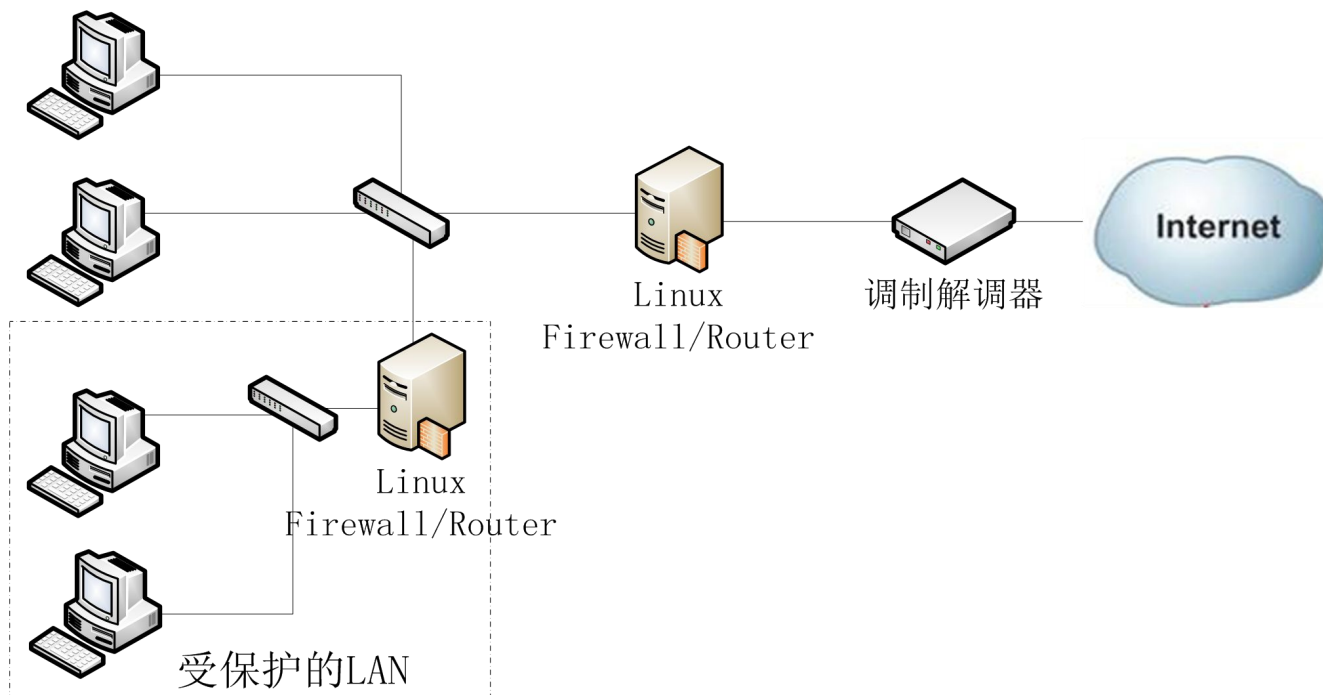
- 认识防火墙
- 防火墙的一般网络布线示意
- 防火墙的使用限制
- Linux的数据包过滤软件:iptables
- 设置单机防火墙实例

- 单一网络，仅有一个路由器
 - 只要管理这一台防火墙主机就可以很轻易的将来自Internet的不良网络数据包阻挡掉

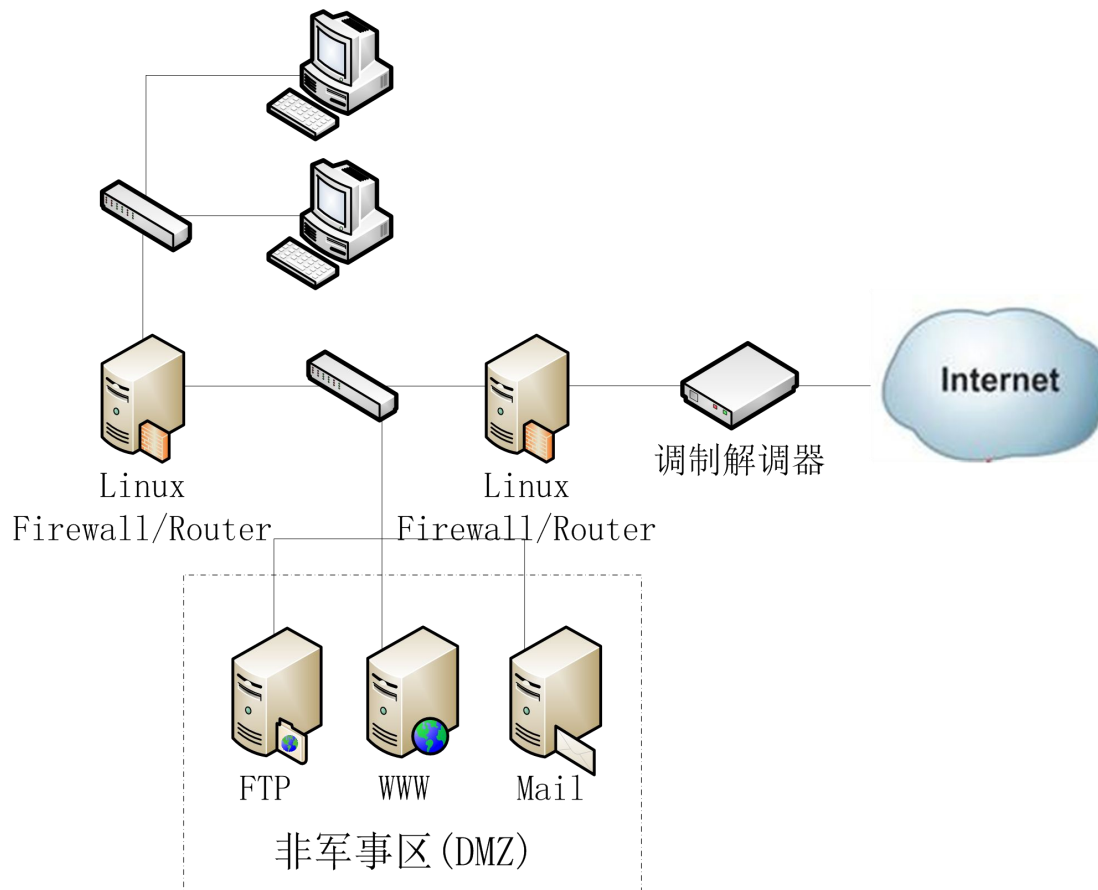


- 如果入侵从LAN进入，那咋办？

- 内部网络包含安全性更高的子网，需要内部防火墙切开子网



➤ 架设在防火墙后端的主机服务器



- 认识防火墙
- 防火墙的一般网络布线示意
- 防火墙的使用限制
- Linux的数据包过滤软件:iptables
- 设置单机防火墙实例

- 设置了防火墙也不能保证网络就一定安全
- 防火墙不能有效阻止病毒或木马程序



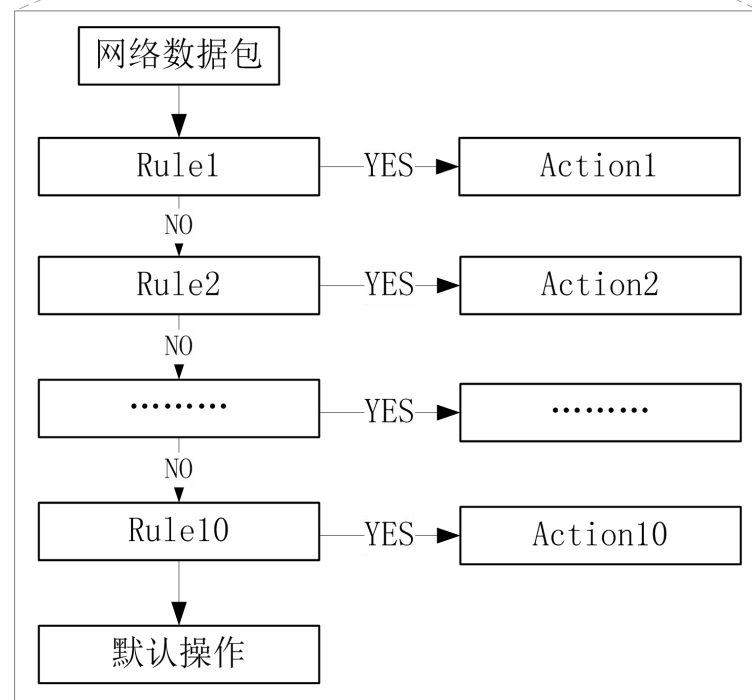
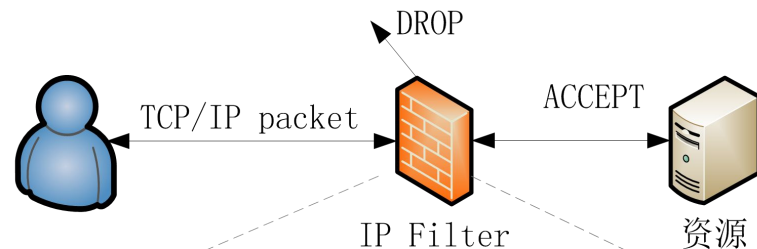
- 防火墙对于来自内部LAN的攻击无能为力



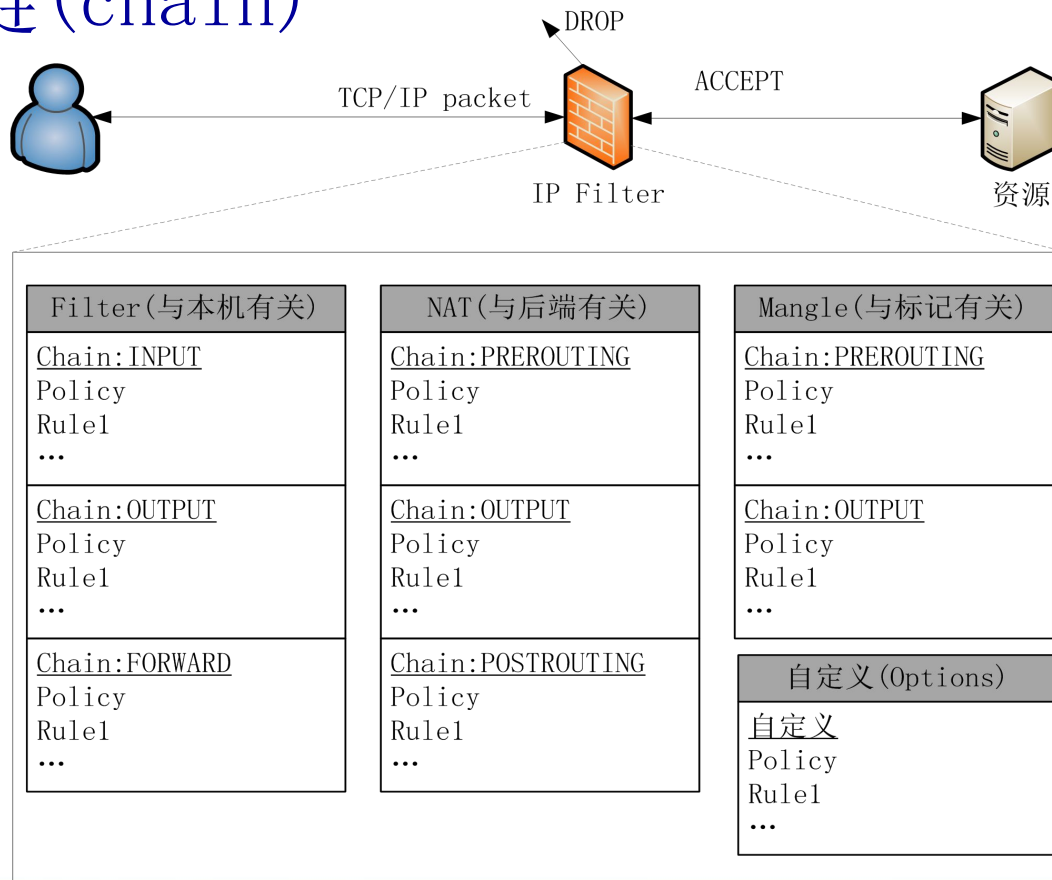
- 认识防火墙
- 防火墙的一般网络布线示意
- 防火墙的使用限制
- **Linux的数据包过滤软件:iptables**
- 设置单机防火墙实例

➤ iptables根据数据包的分析资料“对比”预先定义的规则内容

➤ 对比结果符合Rule1，此时这个网络数据包就会进行Action1的动作，而不会理会后续的Rule2、Rule3等规则了



- iptables有多个表格(table)。而每个表格又有多个链(chain)



- Filter (过滤器): 与本机数据有关
 - INPUT: 主要与想要进入Linux本机的数据包有关
 - OUTPUT: 主要与Linux本机所要送出的数据包有关
 - FORWARD: 与本机无关, 传送数据到后端的计算机中
- NAT (地址转换): 主要用来进行来源和目的地的ip或port的转换
 - PREROUTING: 在进行路由判断之前所要进行的规则
 - POSTROUTING: 在进行路由判断之后所要进行的规则
 - OUTPUT: 与发出去的数据包有关
- Mangle (破坏者): 主要与特殊的数据包的路由标志有关 (很少使用)

- 1. 规则的查看
- iptables [-t tables] [-L] [-nv]
 - -t:后面接table, 例如nat或filter, 若省略则使用filter
 - -L:列出目前的table的规则
 - -n:不进行IP与HOSTNAME的反查, 这样显示速度快
 - -v:列出更多的信息(数据包的位数、相关的网络接口)
- iptables-save会列出完整的防火墙规则(推荐)

➤ 1. 规则的查看

```
Terminal
[~]iptables -t nat -L -n
Chain PREROUTING (policy ACCEPT)
target      prot opt source      destination
Chain INPUT (policy ACCEPT)
target      prot opt source      destination
Chain OUTPUT (policy ACCEPT)
target      prot opt source      destination
Chain POSTROUTING (policy ACCEPT)
target      prot opt source      destination
[~]
```

target:代表进行的操作,ACCEPT接受,REJECT拒绝,DROP丢弃

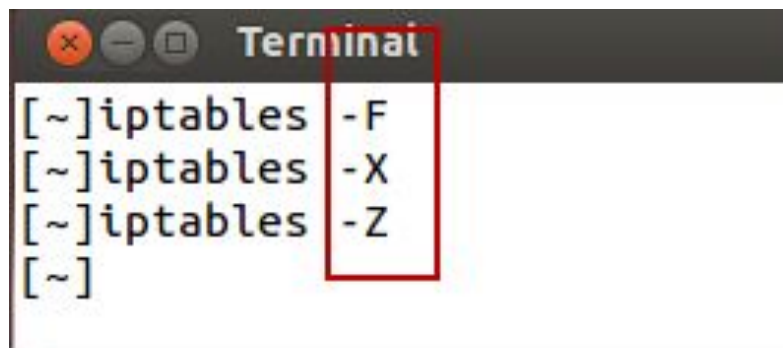
prot:数据包协议,主要有TCP、UDP、ICMP

opt:额外选项说明

source:对来源IP进行限制

destination:对来目标IP进行限制

- 2. 规则的清除
- iptables [-t tables] [-FXZ]
 - -F:清除所有已定制的规则
 - -X:除掉所有用户“自定义”的chain
 - -Z:将所有的chain的计数与流量统计都归零
- 例：清除本机防火墙(filter)的所有规则

A screenshot of a terminal window titled "Terminal". The window shows a series of commands entered at the prompt: [~]iptables -F, [~]iptables -X, [~]iptables -Z, and [~]. A red rectangular box highlights the options -F, -X, and -Z in the first three lines.

```
[~]iptables -F
[~]iptables -X
[~]iptables -Z
[~]
```

➤ 3. 定义默认策略(policy)

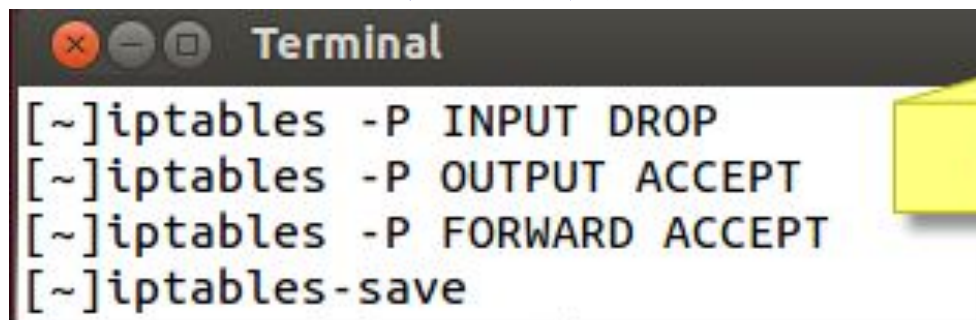
➤ iptables [-t nat] -p [INPUT, OUTPUT, FORWARD] [ACCEPT, DROP]

➤ -P: 定义策略(Policy), P为大写

➤ ACCEPT: 该数据包可接受

➤ DROP: 该数据包直接丢弃, 不会让client知道为何丢弃

➤ 例: 将本机的INPUT设置为DROP, 其他设置为ACCEPT注意先清除所有规则



```
Terminal
[~]iptables -P INPUT DROP
[~]iptables -P OUTPUT ACCEPT
[~]iptables -P FORWARD ACCEPT
[~]iptables-save
```

用ping测试

➤ 4. IP、网络及接口设备的防火墙设置

```
iptables [-AI 链名] [-io 网络接口] [-p 协议] [-s 来源IP/网域] \
```

```
> [-d 目标IP/网域] -j [ACCEPT|DROP|REJECT|LOG]
```

选项与参数:

-AI 链名: 针对某的链进行规则的 "插入" 或 "累加"

-A : 新增加一条规则, 该规则增加在原本规则的最后面。

例如原本已经有四条规则, 使用 -A 就可以加上第五条规则!

-I : 插入一条规则。如果没有指定此规则的顺序, 默认是插入变成第一条规则。

例如原本有四条规则, 使用 -I 则该规则变成第一条, 而原本四条变成 2~5 号链 :
有 INPUT, OUTPUT, FORWARD 等, 此链名称又与 -io 有关, 请看底下。

-io 网络接口: 设定封包进出的接口规范

-i : 封包所进入的那个网络接口, 例如 eth0, lo 等接口。需与 INPUT 链配合;

-o : 封包所传出的那个网络接口, 需与 OUTPUT 链配合;

-p 协定: 设定此规则适用于哪种封包格式

主要的封包格式有: tcp, udp, icmp 及 all 。

-s 来源 IP/网域: 设定此规则之封包的来源项目, 可指定单纯的 IP 或包括网域, 例如:

IP : 192.168.0.100

网域: 192.168.0.0/24, 192.168.0.0/255.255.255.0 均可。

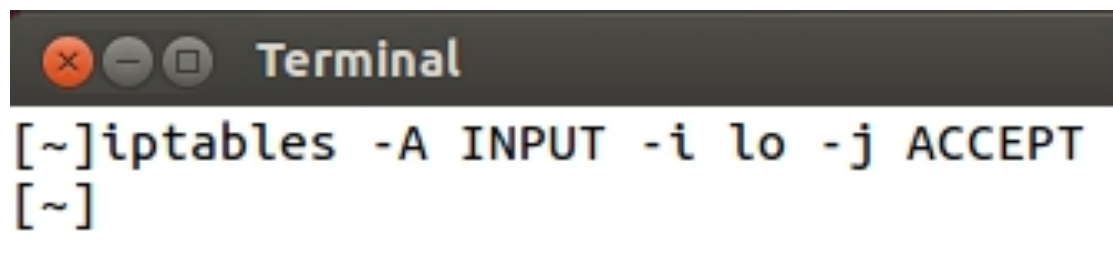
若规范为『不许』时, 则加上 ! 即可, 例如:

-s ! 192.168.100.0/24 表示不许 192.168.100.0/24 之封包来源;

-d 目标 IP/网域: 同 -s , 只不过这里指的是目标的 IP 或网域。

-j : 后面接动作, 主要的动作有接受 (ACCEPT)、丢弃 (DROP)、拒绝 (REJECT) 及记录 (LOG)

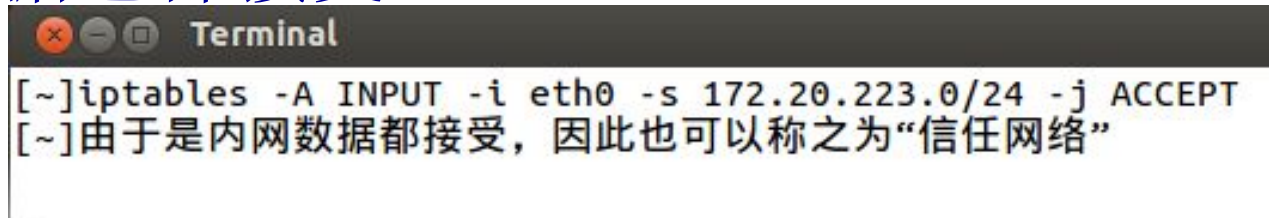
- 例1：设置lo成为受信任的设备，亦即进出lo的数据包都予以接受



```
Terminal  
[~]iptables -A INPUT -i lo -j ACCEPT  
[~]
```

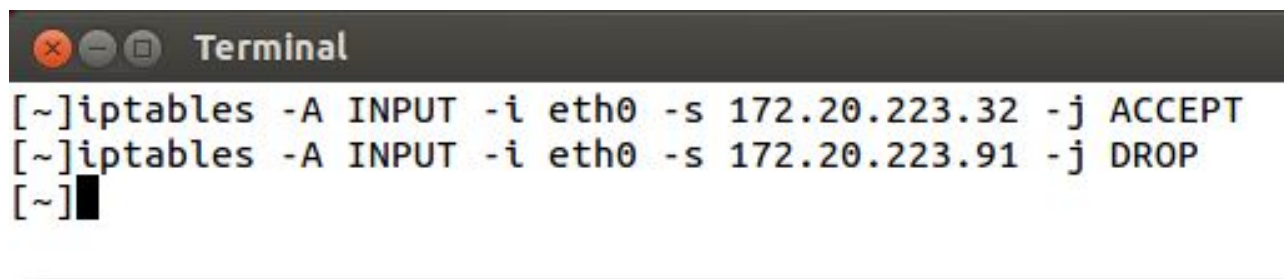
仔细看上面并没有列出 `-s`, `-d` 等等的规则，这表示：不论封包来自何处或去到哪里，只要是来自 `lo` 这个界面，就予以接受！这个观念挺重要的，就是没有指定的项目，则表示该项目完全接受的意思！

- 例2：只要来自内网的(172.20.223.0/24)的数据包都接受



```
Terminal
[~]iptables -A INPUT -i eth0 -s 172.20.223.0/24 -j ACCEPT
[~]由于是内网数据都接受，因此也可以称之为“信任网络”
```

- 例3：只要是来自172.20.223.32就接受，但是来自172.20.223.91的数据包就丢弃



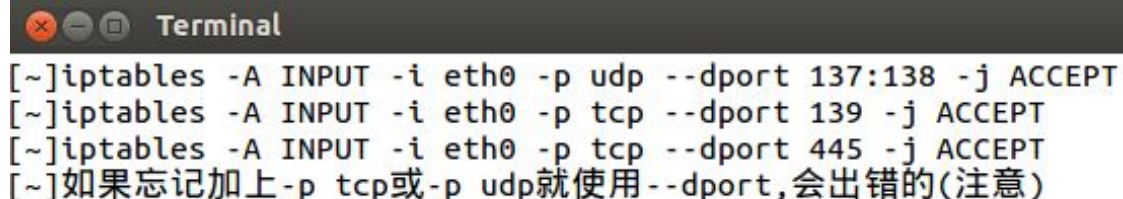
```
Terminal
[~]iptables -A INPUT -i eth0 -s 172.20.223.32 -j ACCEPT
[~]iptables -A INPUT -i eth0 -s 172.20.223.91 -j DROP
[~]
```


➤ 5. 针对端口的防火墙设置

```
iptables [-A 链] [-i 网络接口] [-p tcp,udp] \  
> [-s 来源IP/网域] [--sport 端口范围] \  
> [-d 目标IP/网域] [--dport 端口范围] -j [ACCEPT|DROP|REJECT]  
选项与参数:
```

--sport 端口范围: 限制来源的端口号,
端口号可以是连续的, 例如 1024:65535
--dport 端口范围: 限制目标的端口号。

➤ 例1: 想连接到本机的udp port 137, 138 tcp port 139, 445就放行



```
Terminal  
[~]iptables -A INPUT -i eth0 -p udp --dport 137:138 -j ACCEPT  
[~]iptables -A INPUT -i eth0 -p tcp --dport 139 -j ACCEPT  
[~]iptables -A INPUT -i eth0 -p tcp --dport 445 -j ACCEPT  
[~]如果忘记加上-p tcp或-p udp就使用--dport,会出错的(注意)
```


➤ 6. 对mac与state的防火墙设置

```
iptables -A INPUT [-m state] [--state 状态]
```

选项与参数:

-m : 一些 iptables 的外挂模块, 主要常见的有:

state : 状态模块

mac : 网络卡硬件地址 (hardware address)

--state : 一些封包的状态, 主要有:

INVALID : 无效的封包, 例如数据破损的封包状态

ESTABLISHED: 已经联机成功的联机状态;

NEW : 想要新建立联机的封包状态;

RELATED : 这个最常用! 表示这个封包是与我们主机发送出去的封包有关

- 例1：只要已建立或相关封包就予以通过，只要是不合法封包就丢弃

```
Terminal
[~]iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
[~]iptables -A INPUT -m state --state INVALID -j DROP
[~]
```

- 例2：针对局域网络内的 aa:bb:cc:dd:ee:ff 主机放行

```
Terminal
[~]iptables -A INPUT -m mac --mac-source aa:bb:cc:dd:ee:ff -j ACCEPT
[~]
```

选项与参数：
--mac-source : 就是来源主机的 MAC

- 认识防火墙
- 防火墙的一般网络布线示意
- 防火墙的使用限制
- Linux的数据包过滤软件:iptables
- 设置单机防火墙实例

➤ 超简单的客户端防火墙（实例）

1. 规则归零：清除所有已经存在的规则
2. 默认策略：除了 INPUT 这个自定义链设为 DROP 外，其他为预设 ACCEPT
3. 信任本机：由于 lo 对本机来说是相当重要的，因此 lo 必须设定为信任装置
4. 回应数据包：让本机主动向外要求而响应的封包可以进入本机 (ESTABLISHED, RELATED)
5. 信任用户：这是非必要的，如果你想要让区网的来源可用你的主机资源时



值得信赖的教育品牌

Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com

