防火墙 I

2012-12

课程内容

- 防火墙概述
- Linux 下防火墙简介
- Netfilter 与 iptables 关系
- Netfilter 的功能
- iptables 命令介绍
- 实验题目

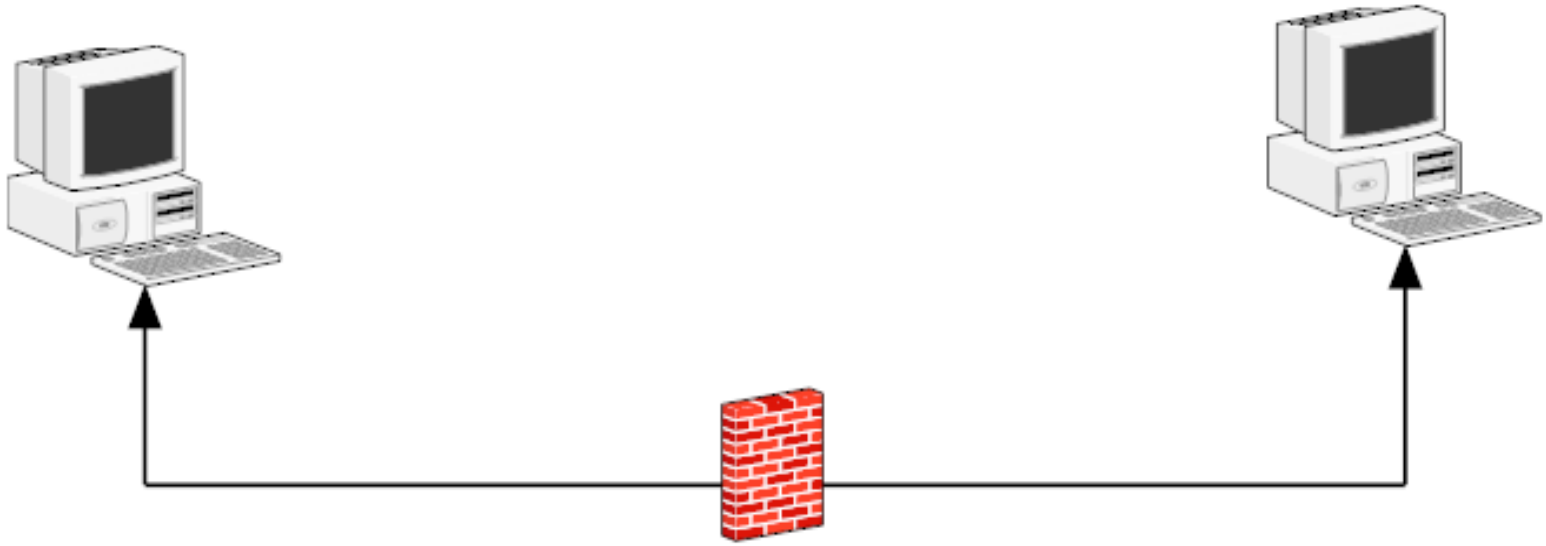
防火墙概述

- 引入防护墙的原因；
- 防火墙的定义；
- 防火墙的类型；

引入防火墙原因（1）

无防火墙时网络中计算机通讯情况

引入防火墙的原因（2）



有防火墙的网络通讯

防火墙的定义

防火墙：

是在两个网络间实现访问控制的一个或一组软件或硬件系统。

防火墙的类型

- 防火墙的存在形式：软件、硬件。
- 根据防范方式和侧重点的不同可分为两类：
  - 主机型
  - 网关型

Linux 下防火墙简介

- 2.0.X 内核　ipfwadm
- 2.2.X 内核　ipchains
- 2.4.X 内核　netfilter/iptables
- 2.6.16 以上　netfilter/iptables 重新设计

Netfilter 与 iptables 的关系

虽然 netfilter/iptables IP 信息包过滤系统作为一个整体看待，但是其实他们一个是该过滤系统的两个组件， netfilter 是内核的模块实现， iptables 是对上层操作工具．

- netfilter 组件也称为内核空间（ kernelspace ），是内核的一部分，由一些信息包过滤表组成，这些表包含内核用来控制信息包过滤处理的规则集。
- iptables 组件是一种工具，也称为用户空间（ userspace ），它使插入、修改和除去信息包过滤表中的规则变得容易。除非您正在使用 Red Hat Linux 7.1 或更高版本，否则需要从 netfilter.org 下载该工具并安装使用它。
- iptables 是一个管理内核包过滤的工具，可以加入、插入或删除核心包过滤表格中的规则。实际上真正来执行这些过滤规则的是 Netfilter 。 Netfiler 是 Linux 核心中一个通用架构，它提供一系列的表（ tables ），每个表由若干链（ chains ）组成，而每条链中可以由一条或数条规则（ rule ）组成．

Netfilter 功能

Filter ： 实现包过滤与状态防火墙的功能；

NAT ： 实现 NAT 的功能，实现数据包的地址转换，允许修改数据包的源和目标地址、端口等

Mangle ：借助这种机制对经过防火墙的数据包进行修改；

Raw ： 负责加快数据包穿越防火墙的速度，以此提高防火墙的性能；

Iptables 命令介绍

中文：

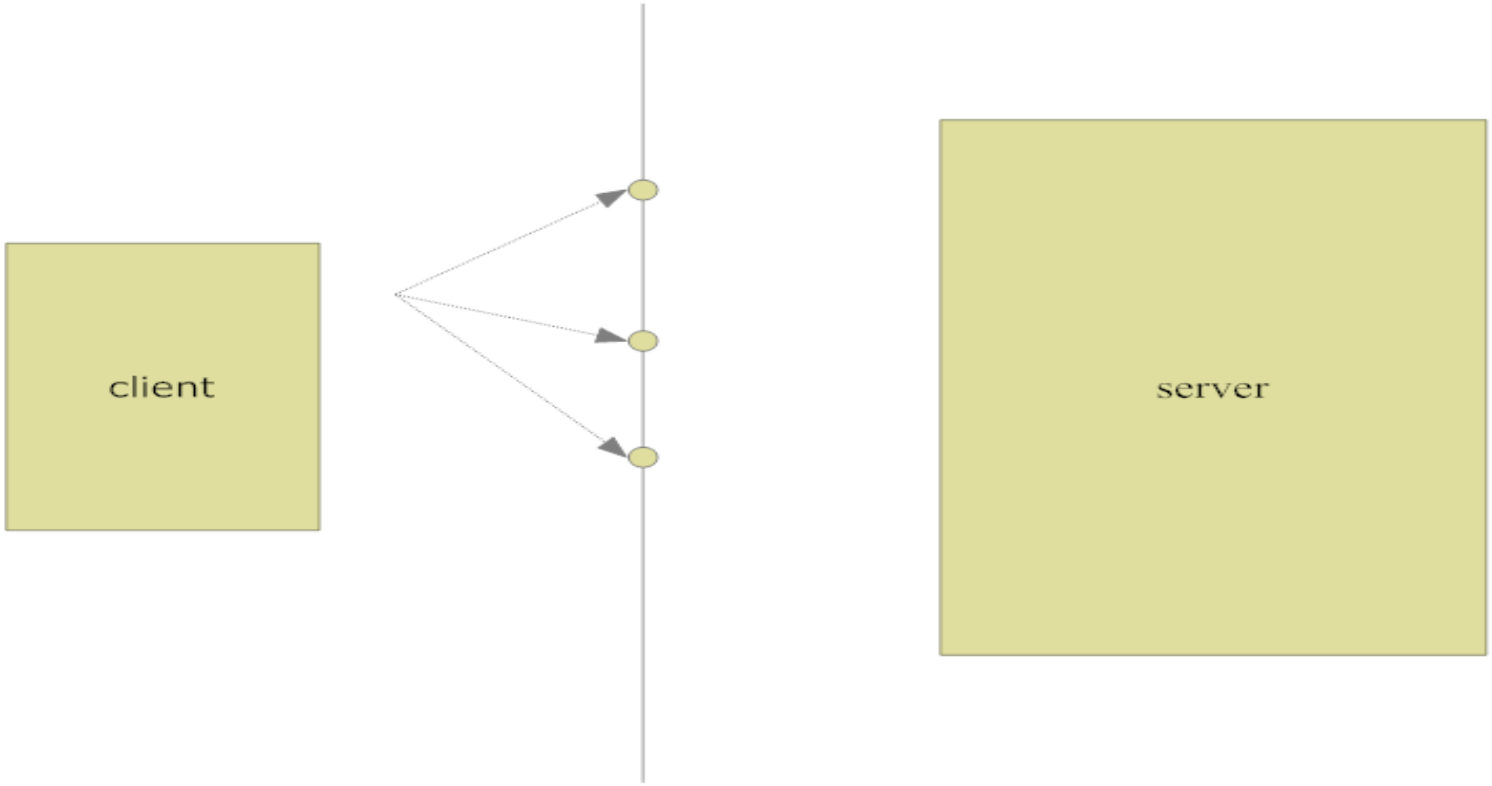
- **http://man.chinaunix.net/network/iptables-tutorial-cn-1.1.19.html**
- **版本： 1.1.19**

英文：

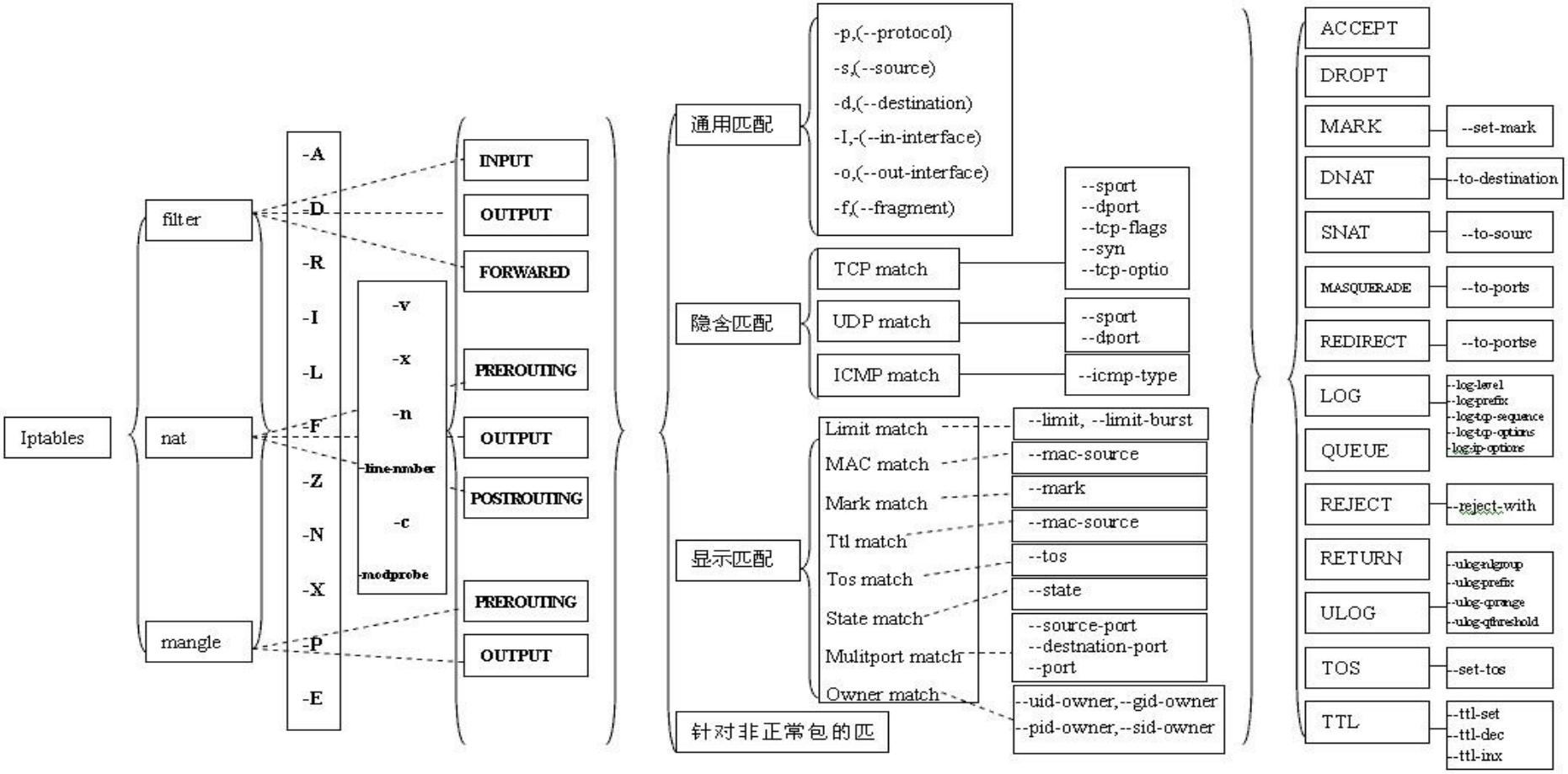
- **http://iptables-tutorial.frozentux.net/iptables-tutorial.html**
- **版本： 1.2.2**

防火墙设置规则

先拒绝所有的服务，再逐一开发对外提供的服务。



Iptables 基本命令



iptables+[-t table]+COMMAND+CHAIN [NO.] +[MATCH] +[-j TARGET]

Iptables 基本命令

- iptables 命令的语法一般如下：
  - Iptables [-flags] [chain] [options[extensions]]　[ACTION]
- iptables 标志项与命令
  - -t table ：制定要操作的表
  - -A ：将一条或多条规则附加到指定链的末尾
  - -D ：从链中删除某个规则
  - -P ：设置链的缺省目标
  - -L ：查看规则设置
  - -F ：清除规则
- iptables 选项：

  -p ：指定协议
  - -d ：指定目标地址
  - -s ：指定源地址

Iptables 基本命令

- iptables 选项：
  - --dport 端口号：指定目标端口
  - --sport 端口号：指定源端口
  - -i 接口名：指定输入接口
  - -o 接口名：指定输出接口
- iptables 事件项：
  - ACCEPT: 允许数据包通过
  - DROP: 将数据包丢弃掉，这种方式会导致源端误认为数据包丢失，而不断发送新包，直到连接 Timeout 为止；
  - REJECT: 将数据包丢弃，并回送一个 destination unreachable 的 ICMP 数据包给发送端，发送端收到这个数据包后，会立即终止连接动作。

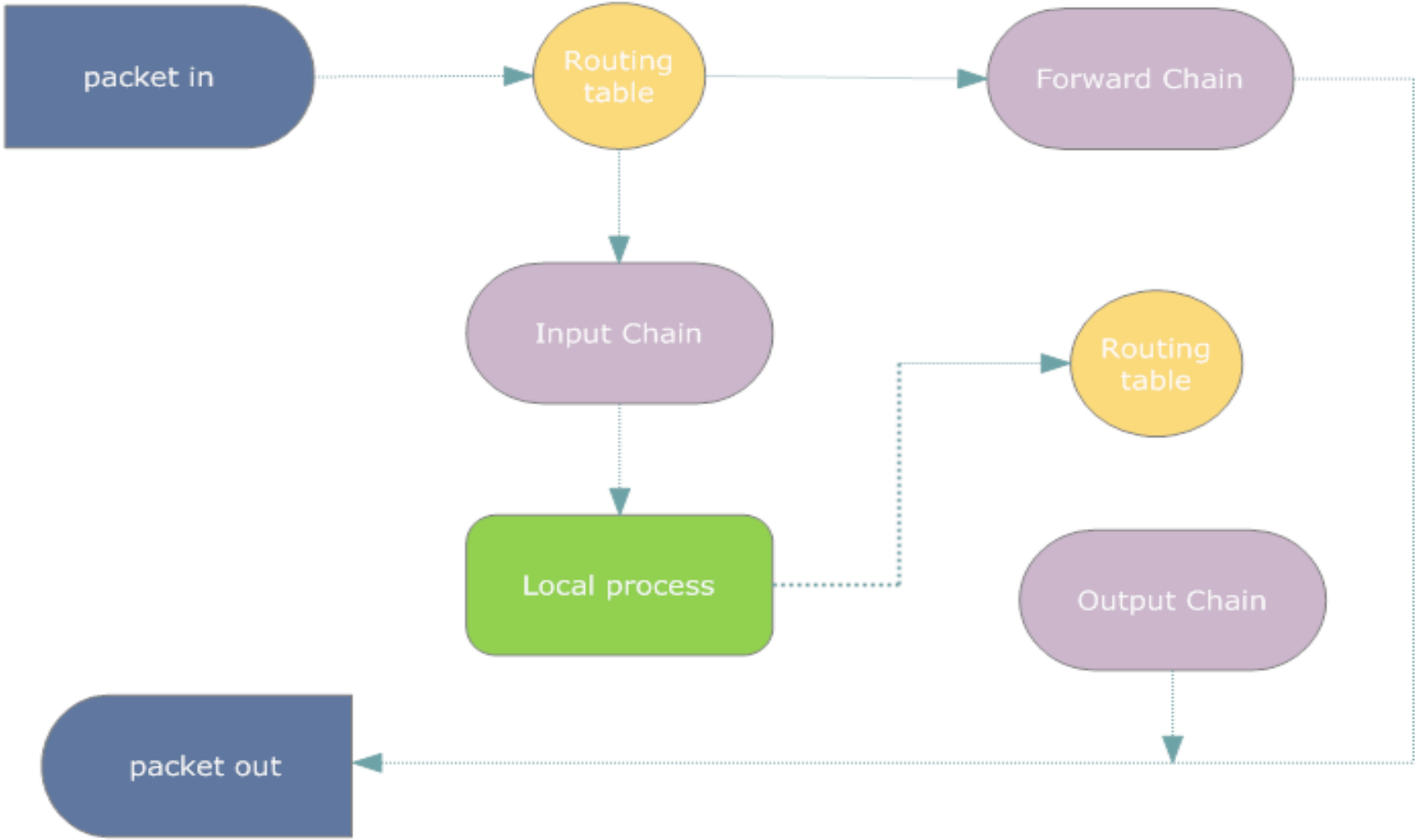
Iptables 中的表

- iptables 可以操纵 4 个表：
  - filter 表
  - nat 表
  - mangle 表
  - raw 表
- 如果不指定，则默认操作 filter 表
- 每个表由若干 “链”（chains）组成
- 每条链由一条或数条 “规则” 组成

链

- 系统缺省的表为“ filter” ，该表中包含了 3 个链：
  - INPUT ：网络上其他主机发给本机的数据包；
  - FORWARD ：由本机转发的数据包；
  - OUTPUT ：本机发送出去的数据包；

Filter 完整结构图



iptables 规则匹配方式

iptables 的规则匹配方式遵循“ first match” 原则：数据包在的特征在第一条规则被匹配，那么该数据包的存活就完全由第一条规则决定，如果被丢弃，那么数据包马上就被丢弃，而不管下面的规则是什么。数据包被接受同样如此

列出防火墙当前的规则：

iptables -t filter [表名] –L [链名]

```
[root@localhost ~]# iptables -t filter -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp  --  anywhere             anywhere            udp dpt:domain
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:domain
ACCEPT     udp  --  anywhere             anywhere            udp dpt:bootps
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:bootps
RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             192.168.122.0/24    state RELATED,ESTAB
LISHED
ACCEPT     all  --  192.168.122.0/24     anywhere
ACCEPT     all  --  anywhere             anywhere
REJECT     all  --  anywhere             anywhere            reject-with icmp-po
rt-unreachable
REJECT     all  --  anywhere             anywhere            reject-with icmp-po
rt-unreachable
RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

清除 table 中的规则

iptables –t filter -F

```
[root@localhost ~]# iptables -t filter -F
[root@localhost ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (0 references)
target     prot opt source               destination
[root@localhost ~]#
```

设定默认规则

iptables -t filter[表名] –P INPUT[链名] DROP

```
[root@localhost ~]# iptables -t filter -P INPUT DROP
[root@localhost ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (0 references)
target     prot opt source               destination
[root@localhost ~]#
```

添加规则—最简单的方式

```
[root@localhost ~]# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

--- 127.0.0.1 ping statistics ---
19 packets transmitted, 0 received, 100% packet loss, time 18022ms

[root@localhost ~]#
```

Iptables –t filter [表名] –A INPUT[链名] –p icmp[协议名] –j ACCEPT[动作]

```
[root@localhost ~]# iptables -t filter -A INPUT -p icmp -j ACCEPT
[root@localhost ~]# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=2.71 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.148 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.105 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.557 ms
^[^A64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.113 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.074 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.109 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.316 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.115 ms
```

删除规则

Iptables –t filter [表名] –D INPUT[链名] –p icmp[协议名] –j ACCEPT[动作]

限制连接

Iptables –t filter –A INPUT –p icmp –s 192.168.0.193 –DROP

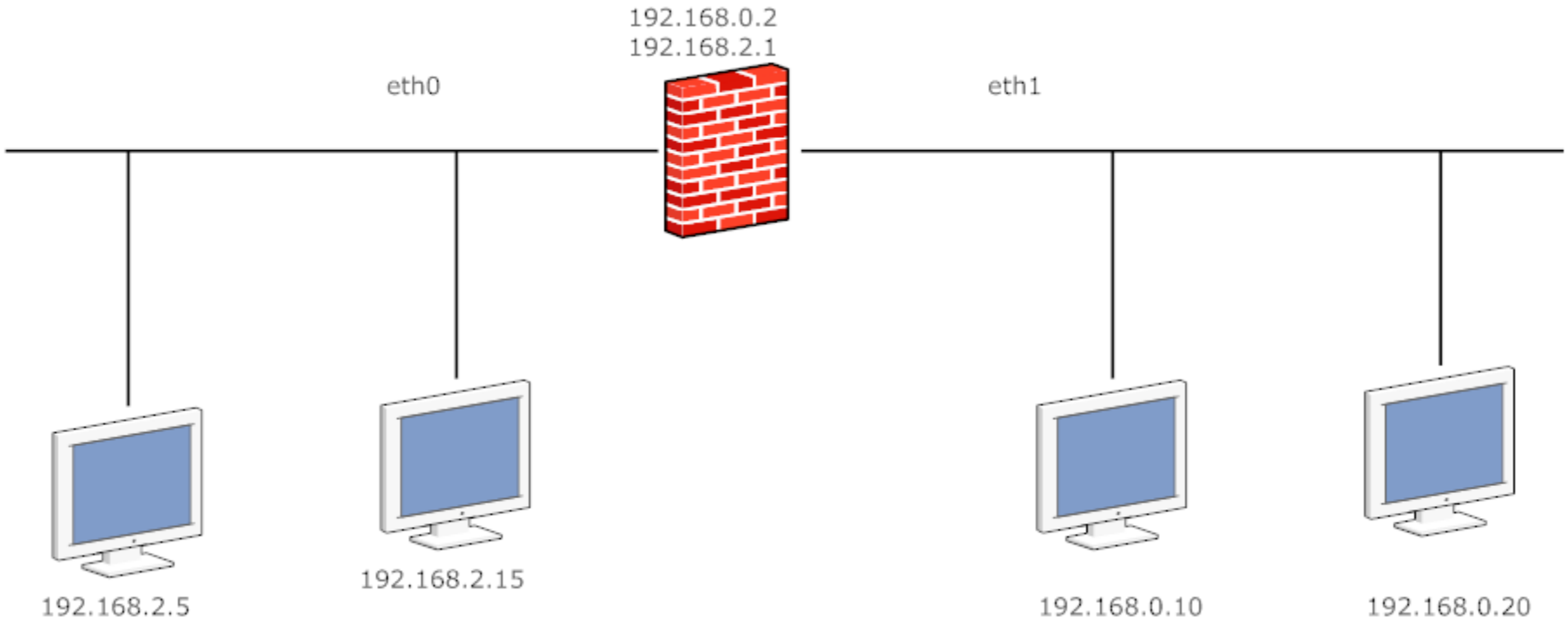
-s 后接的地址可以为某台主机的 IP 地址、某一个网段的网络地址，也可以是某个网站的域名

-d 同上

--dport: 限制目的端口

--sport: 限制源端口

iptables 设置实例



题目

1、在防火墙设置： iptables –A INPUT –p icmp –j DROP ， 192.168.2.15 与 192.168.0.20 哪一台主机可以 ping 到防护墙；

2、在防护墙设置： iptables –A INPUT –i eth0 –p icmp –d 192.168.0.2 –j DROP, 192.168.2.15 与 192.168.0.20 哪一台主机可以 ping 到防护墙；

3、 防护墙有 web service, 在防护墙设置： iptables –A INPUT –i eth1 –dport 80 –s192.168.0.0/24 –j DROP, 图中四台主机哪一台主机可以访问 web service ；

4、 192.168.2.5 有 web service, 在防护墙上设置： iptables -A INPUT –i eth1 –p tcp –d 192.168.2.5 --dport 80 –j REJECT ， 192.168.0.20 与 192.168.2.15 哪一台主机可以访问 web service ；

5、 192.168.2.5 与 192.168.0.10 都有有 web service, 在防护墙上设置： iptables -A FORWARD –i eth0 –o eth1 –p tcp --dport 80 –j REJECT ， 192.168.0.20 与 192.168.2.15 可以访问哪些

防火墙的管理方法

当我们通过 iptables 添加完后规则后，规则就被存储在各个不同的链表中，但是机器重启后，这些规则通通都会消失。

解决方法：

1、将规则存储到规则文件中，然后将防火墙设置为自动启动

```
# service iptables save
# setup 或 chkconfig iptables on
```

2、通过将规则的设置命令编写为 shell 文件的形式；

编写 shell 脚本

在 linux 分区上编写，如 cd /home/

vi iptest.sh

#!/bin/sh

XXXXXXX

chmod +x iptest.sh

./iptest.sh

在启动时自动加载规则

在 /etc/rc.local 中加入需要执行脚本的绝对路径

/XXX/iptest.sh

本次课要求（一）

1. 修改本机 ip 为 10.3.1. （ X ＋ 100 ），其中 X 为 WinXP IP 的最后一位
2. 编写脚本 iptest.sh ，功能如下：
    1. iptest.sh 清空所有存在的规则；
    2. 只开放 22 号端口（ ssh 、 tcp ）；
    3. 禁止发送 icmp 包；
    4. INPUT 默认规则为 DROP ；
    5. OUTPUT 默认规则为 ACCEPT ；
    6. FORWARD 默认规则为 DROP ；
3. iptest.sh 能随系统启动。

本次课要求（二）

1. 编写脚本 iptest2.sh ，功能如下：
    1. 不清空所有存在的规则的情况下；
    2. 只允许 icmp 通过；
    3. 禁止 22 号服务。