实验六网络防御

实验介绍

防火墙是一种访问控制技术，位于可信和不可信网络之间，通过设置一系列安全规则对两个网络之间的通信进行控制，检测交换的信息，防止对信息资源的非法存取和访问，以达到保护系统的目的。防火墙有软件防火墙和硬件防火墙，保护的可信网络为企业内部的网络，不可信网络为Internet。

防火墙在做信息包过滤决定时，有一套遵循和组成的规则，这些规则存储在专用的信息包过滤表中，而这些表集成在Linux 内核中。在信息包过滤表中，规则被分组放在我们所谓的链（chain）中。而netfilter/iptables IP 信息包过滤系统是一款功能强大的工具，可用于添加、编辑和移除规则。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。

TCP Wrapper是一个为Unix服务器提供防火墙服务的公共领域计算机程序。利用TCPwrapper，可以灵活有效地实现对服务请求用户的限制。通过配置/etc/hosts.allow和/etc/hosts.deny这两个文件，就可以指定哪些客户端可以使用哪些服务，而哪些客户端不可以使用哪些服务。当服务请求到达服务器时，TCPwrapper按照下列顺序查询这两个文件，直到遇到一个匹配为止：当在/etc/hosts.allow里面有一项与客户端地址匹配，那么就允许该主机获取指定的服务；否则，如果在/etc/hosts.deny里面有一项与客户端地址匹配，就禁止该主机使用指定的服务；如果没有与客户端地址相匹配的语句，则默认可以获取所有服务；如果这两个配置文件均为空，则表示没有任何限制。

实验内容

1. 掌握Linux系统中防火墙的原理，根据相应的网络环境及要求，完成防火墙配置，对系统进行网络数据包的过滤；

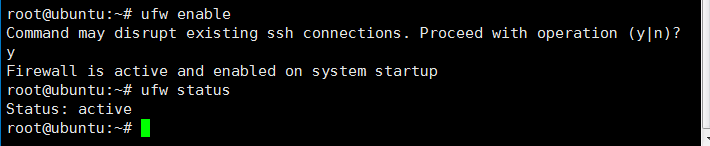
2. 掌握iptables命令的语法，对其进行基本设置。通过增加规则，允许或阻止来自其他主机的连接；

3. 掌握Linux系统中访问控制的方法，学会配置TCP Wrapper实现访问控制。

实验要求

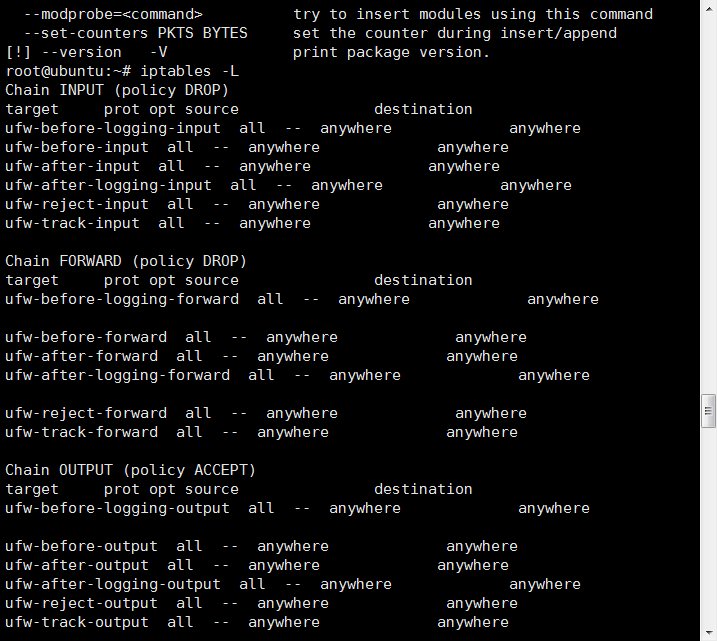
**1.** 配置**iptables**。

1) 开启防火墙。在《实验3：非对称加密》中，我们关闭了Linux虚拟机的防火墙。因此，本次实验需开启防火墙，并重启Linux虚拟机。为方便，以下称Linux虚拟机为服务器，称Windows虚拟机为客户端。



提示：*sudo ufw enable*

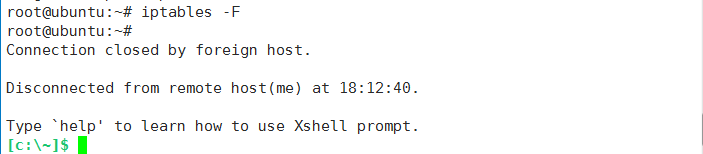
2) 查阅iptables命令手册，查看所有iptables规则。



提示：*sudo iptables -L*

3) 查阅iptables命令手册，清空已有的规则集。与第2)步相比，iptables规则发生了哪些变化？

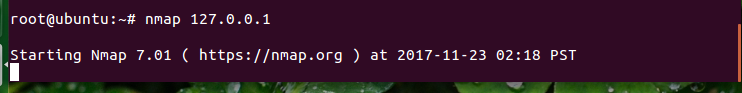
发现断掉了ssh连接



提示：*sudo iptables –F*

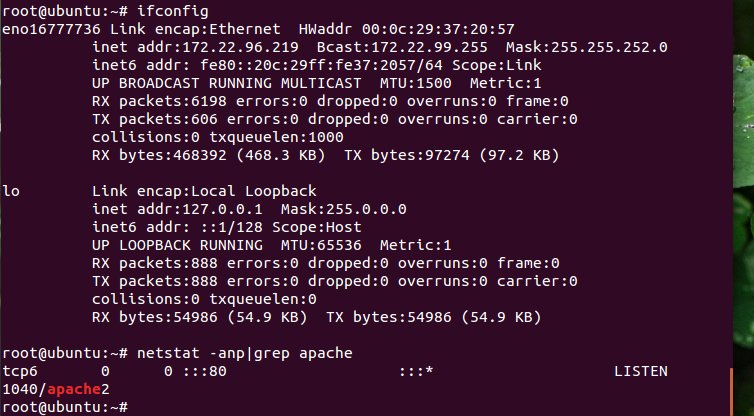
4) 清空规则集后，会导致服务器的本地回环（local loopback）被禁止，各端口不再向客户端开放，服务器被禁止访问外部资源，也不再响应客户端的ICMP请求。在服务器或客户端完成以下操作，分别得到怎样的结果？为什么？

在服务器完成：使用nmap探测本地端口，使用ping命令测试本地回环地址。

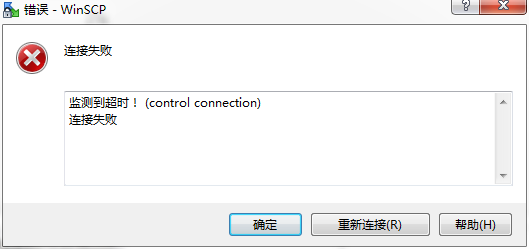




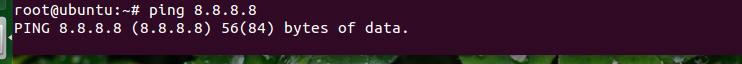
在客户端完成：使用IE浏览器访问Web服务，使用WinSCP访问FTP服务，使用Git Bash访问SSH服务。



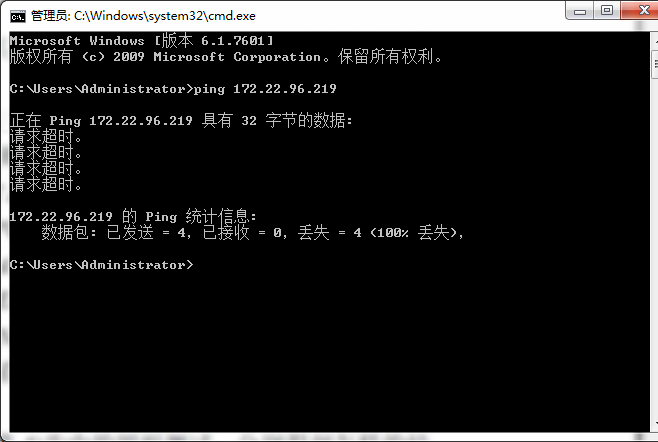




在服务器完成：使用ping命令测试其他外部IP地址，如Google提供的免费DNS服务器地址8.8.8.8。



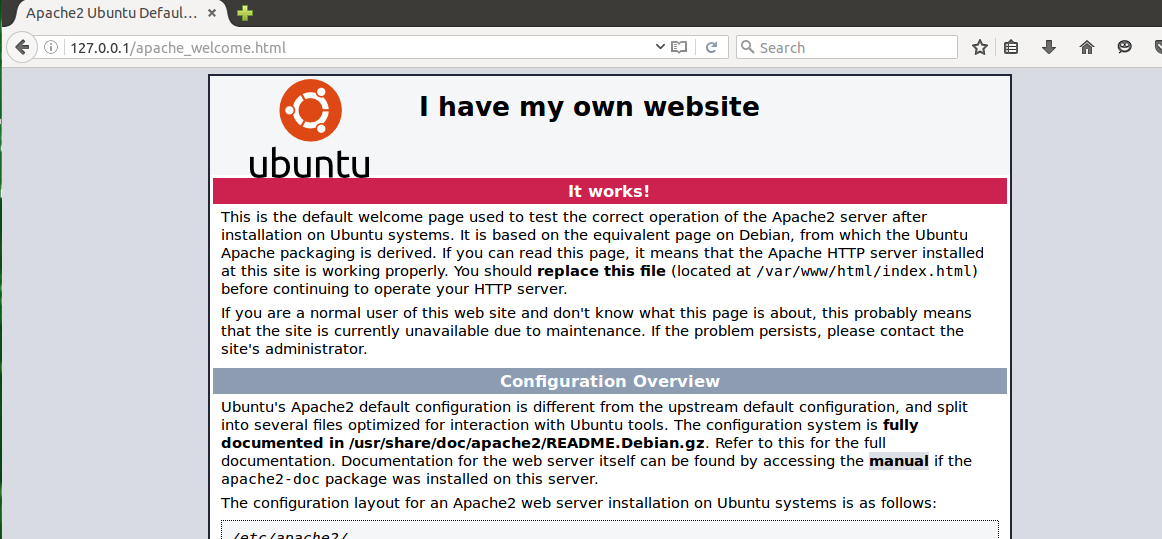
在客户端完成：使用ping命令测试服务器的IP地址。



5) 配置iptables。下面，通过配置iptables规则，逐步恢复上述被禁用的网络连接。

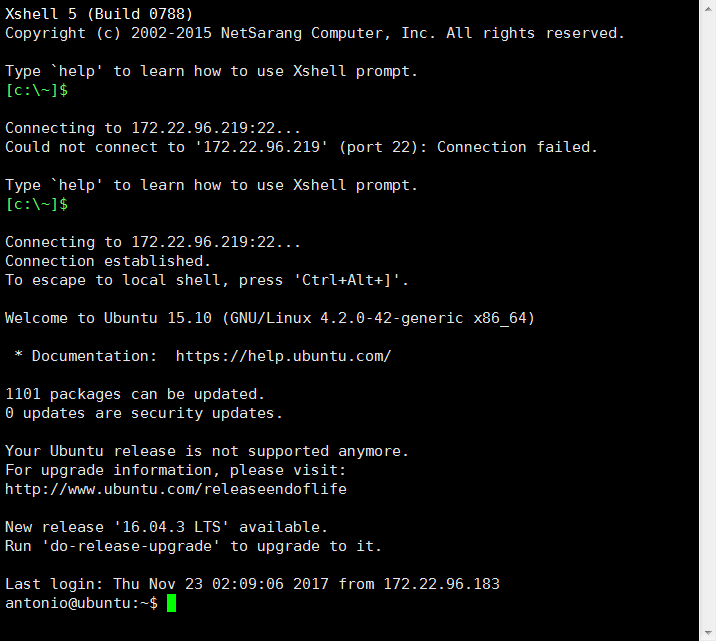
开放本地回环网卡：*sudo iptables* –*I INPUT 1* –*i lo* –*j ACCEPT*



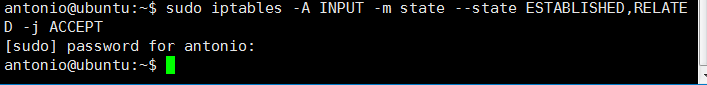


开放网络端口：*sudo iptables* –*A INPUT* –*p tcp --dport* 端口号–*j ACCEPT*





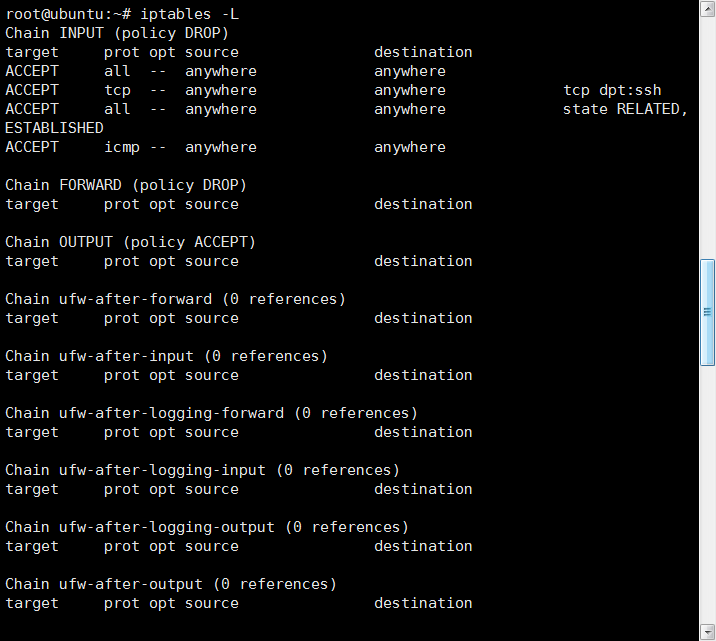
允许服务器访问外部：*sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT*



开放ICMP协议：*iptables -A INPUT -p icmp -j ACCEPT*



6) 配置iptables后，使用*iptables –L*命令查看iptables规则。与第3)步相比，iptables规则发生了哪些变化？



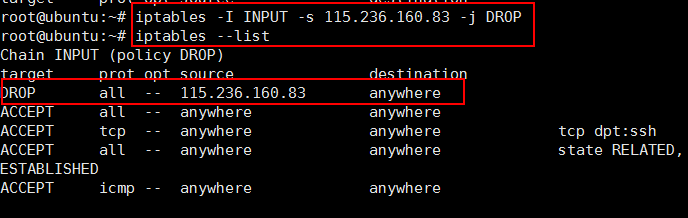
7) 配置iptables后，在服务器或客户端完成第4)步中的四组测试，分别得到怎样的结果？为什么？

都可以通过，

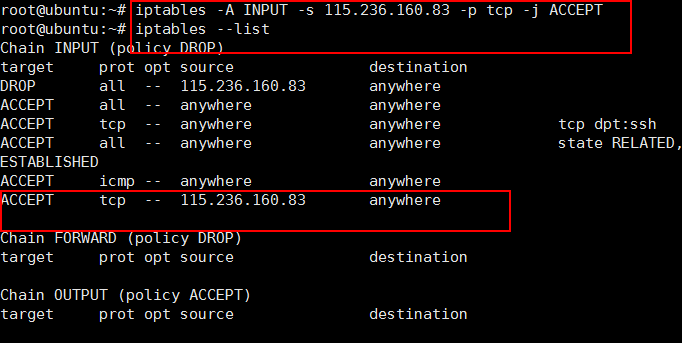
会重新打开服务器的本地回环（local loopback），再次开放22端口号。

8) 练习。查阅iptables命令手册，写出以下需求的iptables规则（不需要给出测试）：

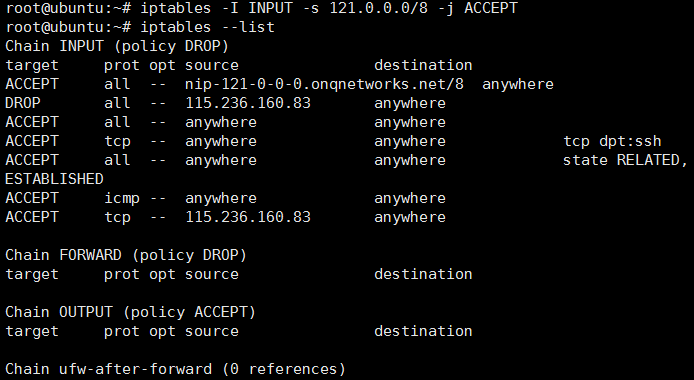
阻塞192.168.47.222这一IP的所有连接。



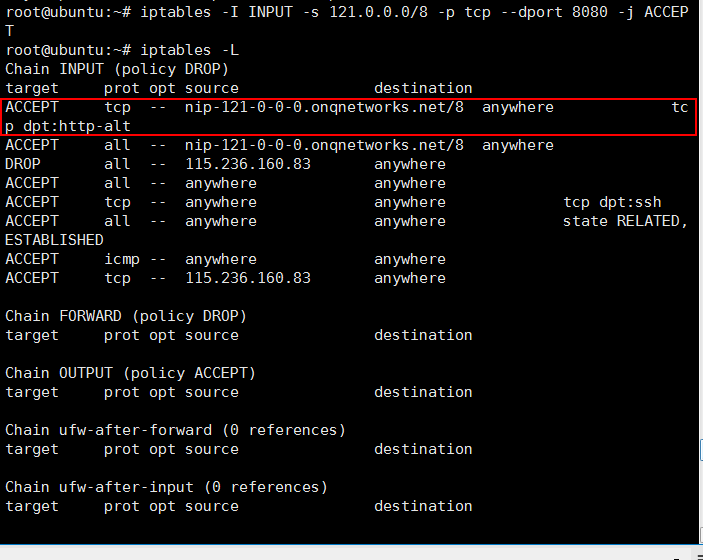
取消上条规则。



允许来自192.168.48.0这一网段的所有连接。



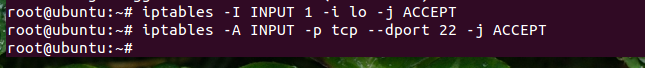
允许来自192.168.49.0这一网段的针对8080端口的所有连接。

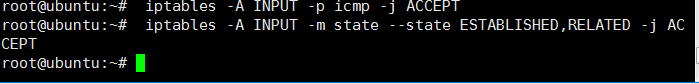


2. 保存**iptables**规则。

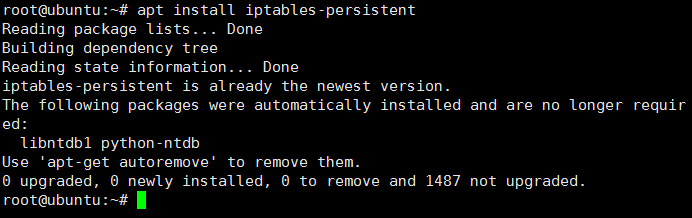
1) 重启服务器，查看iptables规则。第5)步中所做的iptables配置仍然生效吗？为什么？

不能生效，没有保存规则。

2) 重新配置防火墙。清空已有规则集，按照第5)步要求，重新配置iptables。



3) 安装iptables-persistent。

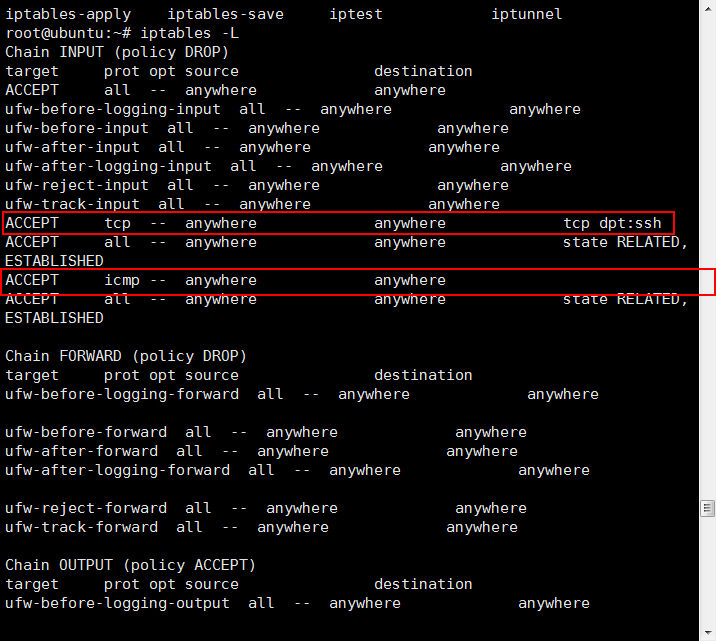


4) 编辑完iptables后，将规则保存到指定文件/etc/iptables/rules.v4



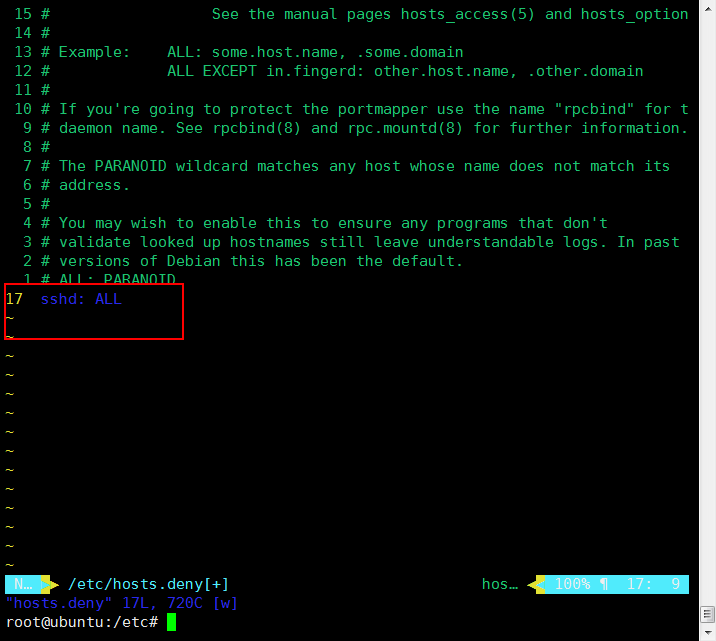
提示：*iptables-save > /etc/iptables/rules.v4*

5) 重启服务器，查看iptables规则。所做的iptables配置是否被保存？



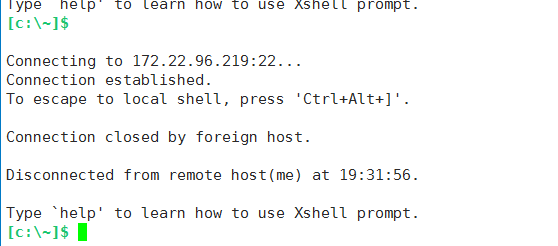
**3. TCP Wrapper**。

1) 配置hosts.deny。编辑/etc/hosts.deny，在文件最后加入一行：



*sshd: ALL*

在客户端上使用Git Bash访问SSH服务。你得到怎样的结果？为什么？

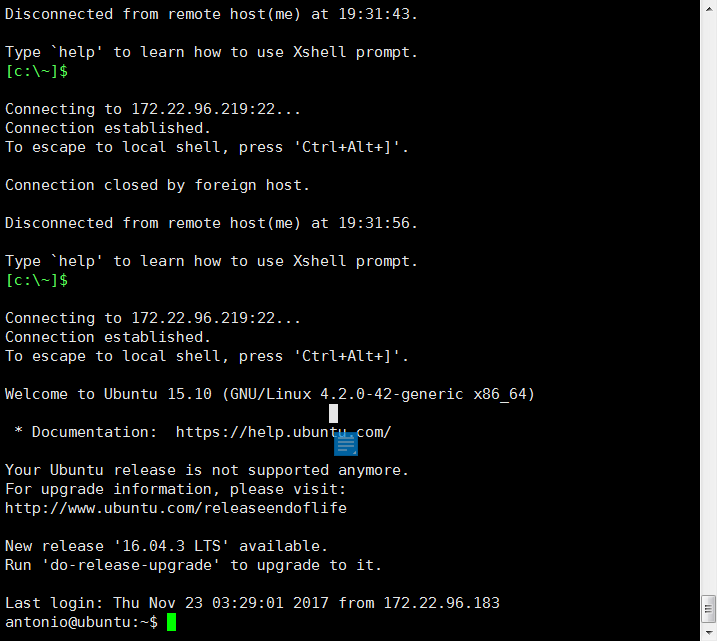


连接不上，因为拒绝连接

2) 配置hosts.allow。编辑/etc/hosts. allow，在文件最后加入一行：

*sshd:* 客户端*IP*地址

在客户端上使用Git Bash访问SSH服务。你得到怎样的结果？为什么？



在hosts.allow 里加入了本机ip地址，故而又可以连接了