如何随心订制Linux的透明防火墙 PDF转换可能丢失图片或格式，建议阅读原文
https://www.100test.com/kao_ti2020/145/2021_2022__E5_A6_82_E4_BD_95_E9_9A_8F_E5_c103_145311.htm 一般而言，防火墙的两个网络接口应分属两个不同的网络，根据系统管理员定义的访问规则在两个接口之间转发数据包，或者拒绝、丢弃数据包。实际上，防火墙不单单是访问控制的功能，而且还充当了路由器的角色。当然，这并非有什么不妥当的地方，但是当你企图把你配置好的linux防火墙放入运行网络，来保护现有系统安全的时候，你不得不重新考虑和更改你的网络架构。另外一个可能的麻烦是，当防火墙发生意外时,如果没有防火墙的硬件备份的话,那么你将面临巨大的心理压力,因为防火墙的故障,整个网络瘫痪了。假如你把防火墙配置成透明模式(可称为伪网桥)，就无需更改网络架构，即使是防火墙不能工作了，要做的仅仅是拔出网线，把网线直接插在路由器的内部接口就可以让网络正常工作，然后你就有时间慢慢恢复发生故障的防火墙。 好了，既然透明防火墙有那么多方便，我们赶快动手来配置吧！准备一台pc机，两块网卡(建议用3com网卡),网线若干， redhat linux 9安装盘一套。打开机箱，把两块网卡插入计算机的pci插槽，用网线把计算机分别与网关和交换机相连(如前页图“ 正常状态 ”那样)；盖上计算机的盖子，插上电源，开机。在光驱里放上Linux 9安装光盘，由光盘引导计算机，从而安装Linux 系统。选择定制安装，不要保守，多花一点时间体验一下图形界面的安装乐趣，取消防火墙(no firewall)，在安装快结束时选择以文本方式登录系统，完成安装。 透明防火墙功能配置： 1、 设置网络地址

修改文件 /etc/sysconfig/network-scripts/ifcfg-eth0 和 /etc/sysconfig/network-scripts/ifcfg-eth1，使其具有相同的ip地址，相同的子网掩码。 用vi 来编辑如下，保存文件，运行命令 service network restart 使修改生效。 DEVICE=eth0 BOOTPROTO=none BROADCAST=192.168.1.255 IPADDR=192.168.1.254 NETMASK=255.255.255.0 NETWORK=192.168.1.0 ONBOOT=yes USERCTL=no PEERDNS=no TYPE=Ethernet DEVICE=eth1 BOOTPROTO=none BROADCAST=192.168.1.255 IPADDR=192.168.1.254 NETMASK=255.255.255.0 NETWORK=192.168.1.0 ONBOOT=yes USERCTL=no PEERDNS=no TYPE=Ethernet 这里需要注意两个地方，第一个是要区分清楚那一个网卡是eth0,那一个是 eth1.这个问题十分关键，如果搞混了就会导致防火墙不能连通网络。至于怎样区分eth0和 eth1，我将在文章的末尾作简单的描述。在这里假定与路由器相连的网卡是eth0. 2、设置默认路由 在文件 /etc/sysconfig/network-scripts/ifcfg-eth0 中加入一行 gateway= 192.168.1.1 保存后运行命令 service network restart ，修改生效。找一个开放ICMP协议的公网IP，用命令 ping 202.108.36.196 (www.163.com 的主机)检测跟外网的连通状况，如果正常，表明Linux防火墙主机跟外网配置正确。再用命令ping 192.168.1.18 检测防火墙主机与内网主机的连通状况，如果正常则进行下一步操作。 3、 启用网络转发和proxy_arp 这是透明防火墙的核心部分，我把它们写进文件/etc/rc.d/rc.local。用vi /etc/rc.d/rc.local 插入如下内容。在做这一步的时候，我曾 #Ip forward /sbin/sysctl -w net.ipv4.conf.all.forwarding=1 #Enable

proxy-arp /sbin/sysctl -w net.ipv4.conf.eth0.proxy_arp=1 /sbin/sysctl -w net.ipv4.conf.eth1.proxy_arp=1 经花费较多的时间，因为我做参考的那本书里的这一步没有参数“ w ”,后来单独运行 sysctl net.ipv4.conf.eth0.proxy_arp=1 才发现red hat Linux 9 没有参数“ -w ”不能运行。 4、指定路由 由于两块网卡(eth0,eth1)使用同样的ip，如果不专门指定转发路径，一定会导致路由混乱，从而使防火墙以内的计算机没法访问Internet 。还是用命令 vi 修改文件 /etc/rc.d/rc.local ，插入如下几行。保存文件，重新启动计算机/ #Define route /sbin/ip route del 192.168.1.0/24 dev eth0 /sbin/ip route add 192.168.1.1 dev eth0 /sbin/ip route add 192.168.1.0/24 dev eth1 Linux防火墙，如果不出意外，就可以从192.168.1.18 这台主机访问Internet，当然内网的任何机器都是可以访问 Internet 的。在这里对定义的路由(Define route)作些说明： /sbin/ip route del 192.168.1.0/24 dev eth0 表明所有到子网192.168.1.0/24的数据包都不从网卡eth0转发而从 eth1转发，即命令 /sbin/ip route add 192.168.1.0/24 dev eth1； /sbin/ip route add 192.168.1.1 dev eth0 表明所有到192.168.1.1的数据包都由eth0转发，这其实可以理解为两个网卡数据转发的分工到192.168.1.1 的数据包由eth0负责，其余的由eth1负责。到这一步，恭喜你！已经成功了一大半，如果安装Linux的时候，选择的防火墙规则为中等级别，那么这个防火墙已经配置成功了。相信大家跟我一样，且肯就此罢休。 定制防火墙策略 都是2.4.20的内核版本，当然要用netfilter/iptables。由于安装Linux系统的时候，选择了“ 无防火墙 ”这个选项，那么在/etc/sysconfig 下将没有iptables这个文件存在。还是让我们随心所欲的来定制防火墙访问策略吧。

在目录 /etc/rc.d 下创建脚本文件 myfirewall.sh,用命令 touch /etc/rc.d/myfirewall.sh并给文件执行权限 chmod 711 myfirewall。然后用 vi 编辑这个文件。我写的这个 vi /etc/rc.d/myfirewall.sh #!/bin/bash #Define string IPT=/sbin/iptables #Refresh rules $IPT -F FORWARD $IPT -F INPUT $IPT -F OUTPUT #Default policy $IPT -P INPUT DROP $IPT -P FORWARD DROP $IPT -P OUTPUT ACCEPT #Enable loopback $IPT -A INPUT -i lo -p all -j ACCEPT #Enable icmp $IPT -A INPUT -p icmp j ACCEPT #Interface forward $IPT -A FORWARD -s 192.168.1.0/24 -j ACCEPT $IPT -A FORWARD -d 192.168.1.0/24 -j ACCEPT #Enable ssh $IPT -A INPUT -p tcp --dport 22 -j ACCEPT #Add other access rule //可根据实际情况添加或减少规则 $IPT -A INPUT -p tcp --dport 20 -j ACCEPT $IPT -A INPUT -p tcp --dport 21 -j ACCEPT $IPT -A INPUT -p tcp --dport 80 -j ACCEPT $IPT -A INPUT -p tcp --dport 53 -j ACCEPT $$IPT -A INPUT -p udp --dport 53 -j ACCEPT $IPT -A INPUT -p tcp --dport 23 -j ACCEPT $IPT -A INPUT -p tcp --dport 110 -j ACCEPT $IPT -A INPUT -p tcp --dport 25 -j ACCEPT $IPT -A INPUT -p tcp --dport 443 -j ACCEPT 规则只开放了较少的允许访问的策略(可以ping ,收发邮件，浏览网页，ssh，https,telnet，ftp，其它的访问则全部丢弃)。$ IPT A OUTPUT ACCEPT 没有设置成DROP的原因是由于大部分网络服务所使用的协议是tcp协议，众所周知，tcp协议是面向连接的，如果设置 $IPT A OUTPUT DROP, 那么任何协议为tcp的连接就要写两条了。况且防火墙对外的访问总是允许的，因此这样做是为了简化规则。 修改完成后保存，然后在当前目录运行命令

./myfirewall.sh,在上述脚本没有书写错误的情况下，规则生效，但它仅仅在内存里，用命令 service iptables save 将自动生成文件 /etc/sysconfig/iptables，前面设定的访问策略就被保存到硬盘，系统重启时，系统将自动地从文件 /etc/sysconfig/iptables 获得定制的访问策略。 到这里，一个透明的linux 防火墙就架设好了。更改计算机的BIOS设置，使它可以在没有键盘的情况下启动系统。启用ftp，以便可以在需要时可以向防火墙主机拷贝文件。把键盘和显示器拿掉，剩下的操作只是摁一下电源开关。 防火墙的管理 可能有时候我们需要更改防火墙的某些规则，或者做些别的管理，既然我们是系统管理员，再插上键盘和接上显示器坐在防火墙面前可能会被人耻笑，因此这些管理工作当然通过网络来进行。Ssh和webmin是我的偏好，ssh的协议端口是22，webmin的默认协议端口是10000。其中ssh是 linux系统的默认服务，只要安装客户端就可以(windows下的程序securecrt 是个不错的选择，据说ssh连接速度没有vnc 快)对防火墙进行所有的管理(和直接操作防火墙主机一样)；webmin是基于web的图形界面管理方式，非常的方便和直观，尽管它不能象ssh那样对系统进行完全的管理，但是对于我们的工作需求还是可以满足，建议在防火墙系统安装webmin服务器程序。Ssh与webmin两者结合使用，可以帮助我们较快较深入地掌握Linux。 Ssh客户端安装较为简单，而webmin不需要安装客户端。这里介绍webmin 服务器的安装：把webmin-1.110. tar.gz 下载到另外一台windows的硬盘里，然后用ftp把它复制到防火墙主机的ftp目录(如果你是linux高手，并不需要如此，只须以ssh方式登录防火墙，用get/wget指令取得该文件)，解开文

件webmin-1.110.tar.gz tar zxvf webmin- 1.110.gz.tzr cd webmin-1.110 安装webmin ./setup.sh ,一路回车，创建一个webmin管理账户，安装完毕；在任何一台运行浏览器的地址栏输入防火墙的ip加上端口号10000就可以管理防火墙(192.168.1.254:10000)。以这种方式管理linux 网络的防火墙十分直观，并且选项十分详尽，就算不懂iptable语法的人也能容易的配置防火墙的访问规则。这里有一个技巧，假如你更改了某条访问规则导致网络不能向外访问，不要慌，到防火墙跟前重启一下系统即可。万一更改规则发生不测并且规则已经写入硬盘，那么请你直接删除文件 /etc/sysconfig/iptables, 然后再运行脚本 sh /etc/rc.d/myfirewall 再次重写文件 /etc/sysconfig/iptables service iptables save 。有的系统管理员倾向于直接编辑 /etc/sysconfig/iptables 文件，但是这需要更多的耐心和勇气。如果你是新手，建议你跟我一样，先写脚本，再生成 iptables。 特别关注： 最好把除路由器而外的整个网络放在防火墙的保护之中。如果象那样有同一网段的主机放在防火墙的前面，将导致严重的网络故障。实践表明，这台windows主机的ip地址丢失了(网络属性的ip值还在，但用命令 ipconfig /all 则是 0.0.0.0)，重启windows后提示 ip地址冲突，更换同一网段内的任何一个未用的ip地址还是提示冲突。搞的我的两台邮件服务器和两台web服务器停火，我还以为是中了邪门的病毒，直到后来我把tcp/ip协议卸载再安装才解决问题。经分析，是防火墙的路由导致这样的故障。强烈建议把所有的主机放在防火墙的保护之下，以减少网络的复杂程度。另外，我们应该养成这样一种习惯在系统正常的情况下，如果更改了配置，请一定要用笔记录所作的更改，以便在改出

问题时我们能够快速准确的恢复，这种习惯更可运用到所有的IT管理工作,它是我的不传之密。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com