China-pub.com
下载

第3章　配置TCP/IP网络

3.1　开始之前需要的信息

在设置网络软件之前，需要知道一些有关的网络结构的信息，你的网络“提供者”或“管理者”将提供这些信息。

3.1.1　IP地址

这是每一台机器惟一的地址，由四个十进制字段组合起来。例如：某个数 128.253.153.54，网络管理者会提供你这些信息。若是使用 slip 或plip的联线，可能不需要这些信息，可以略过不读。如果只是使用 loopback 设备（也就是说不连以太网络、slip或plip支持），将不需要IP地址 因为loopback port总是使用 127.0.0.1 为本机地址。

3.1.2. 网络掩码

为了执行效率起见，需要限制网络上的特定区段的主机数，通常网络管理者把他们的网络分成几个子网(subnet)，然后分配每个子网一部份 IP地址，当网络掩码(Network Mask)按位AND你自己的网络上的IP地址时，就得到它所在的子网地址。

这对路由来说很重要。网络管理者在网络被规划时，就已选定了网络掩码，因此他将提供你正确的网络掩码。大部分 C类子网，都使用255.255.255.0作为网络掩码，其他大的网络使用B类网络掩码(255.255.0.0)。当你分配一个地址给一个设备的时候，NET-2/NET-3 程序将自动地选择内定的网络掩码、内定值。假设你的网络还没被划分为子网，NET-2/NET-3 程序会选择以下的设置作为内定信息。

对于网络地址的第一个字节：

```
1～127          255.0.0.0           (Class A)
128～191        255.255.0.0         (Class B)
192--           255.255.255.0       (Class C)
```

如果以上这些都不能运行，试试其他的。如果都还不行，请询问你的网络管理者，不必担心loopback 端口的网络掩码，除非你运行的是 slip/plip 。

3.1.3　网络地址(Network Address)

它是使用你的网络掩码所运算出来的地址。IP地址与网络掩码按位进行 AND运算。例如：

```
你的网络掩码：          255.255.255.0
你的IP地址：            128.253.154.32
你的网络地址：          128.253.154.0
```

3.1.4　广播地址(Broadcast Address)

广播地址是以你的网络地址与掩码做 XOR运算后得到的。以下是一个简单的范例：

China-pub.com
下载


```
你的掩码是：                    255.255.255.0
你的网络地址是：                128.253.154.0       XOR)
则你的广播地址为：              128.253.154.255
```

注意一个由来已久的原因：一些网络以网络地址当作广播地址，如果你有疑问，请与网络管理员联系。

如果能使用sniffer，或其他能提供你追踪网络流量的工具的话，你就可以通过查看其他LAN的流量，来决定网络地址及广播地址，注意，除了往广播地址 ff：ff：ff：ff：ff：ff流去的以太网包之外的每个封包。如果其中有一个你的区域的路由器的 IP源地址，而且 protocol ID 不是 ARP 的话，要检查目的IP地址，因为这个封包可能真的是个从你的路由器广播出去的 RIP，这种情况下目的IP地址将是你的广播地址。再次警告，如果你不能确定，一定要跟网络管理员联系，他们的帮助比你用设置错的机器来连网络还有利。

3.1.5 网关地址(Router/Gateway Address)

这是一个将你的 LAN 连至 Internet 的机器的地址，那是你的电脑连到外面网络的穿堂，有许多的先例可用来分配地址给路由器，你也可以引用，有两种：

1) 路由器使用最低的 address。

2) 路由器使用最高的数字。

可能最常用的是第一个，路由器会拥有一个地址，大部分会跟你的一样，除了最后一个字节是“ .1 ”之外。例如：假如你的地址是 128.253.154.32，那你的路由器可能是128.253.154.1。路由器实际上可以使用任何网络上可用的地址，仍能正常工作，跟地址丝毫没有关系。一个网络上可能有好几个路由器，你可能要向网络管理者询问看看，以得到一个合适的路由器地址。

如果你使用loopback的方式，就不需要路由器地址。如果你使用 PPP，那也不需要知道你的路由器地址，因为 PPP会自动替你决定正确的路由地址。如果你使用 SLIP，你的路由器地址将会是你的SLIP server的地址。

3.1.6 名字服务器地址(Nameserver Address)

大部分网络上的机器都使用名字服务器，名字服务器负责翻译主机名 (hostname)与IP地址等等，你的网管会告诉你网络上最近的名字服务器，你也可以在自己的机器上运行一个 name server daemon(named)，而你的名字服务器就要设成 127.0.0.1 (也就是loopback的端口地址)。不过，也没必要在自己的机器上运行 named，需要的话可看 named的manual。如果只使用loopback，那就不需要知道名字服务器地址，因为你只跟自己的机器相联。

3.2 用netcfg配置网络

图3-1中所显示的网络配置工具(netcfg)用来简便地操纵各种参数，诸如IP地址、网关地址、网络地址，还有域名服务器及 /etc/hosts等。

可以增加、删除、配置、激活和关闭网络设备，以及取别名等，支持 Ethernet、arcnet、token ring、pocket(ATP)、PPP、SLIP、PLIP以及loopback 设备。对PPP/SLIP/PLIP 的支持在大多数硬件平台上都支持得很好。但是有一些硬件配置将会产生不可预料的结果。当使用网

China-pub.com 下载

络配置工具时，点击 Save将所做修改保存到硬盘上，若不想修改而直接退出的话，直接点击Quit即可。

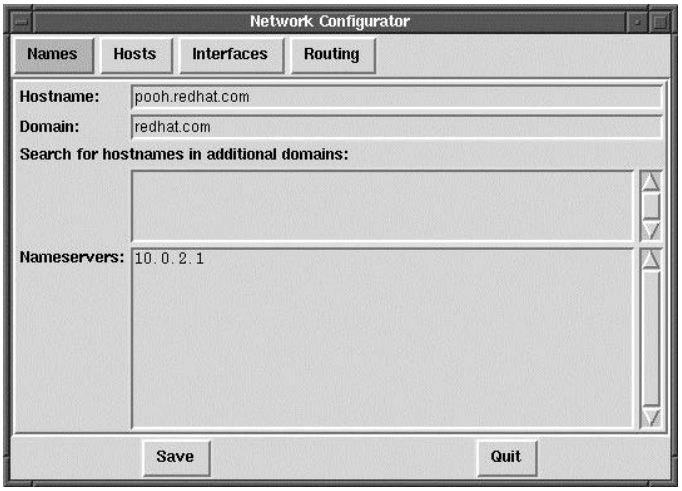

图 3-1

3.2.1 操纵名字

网络配置工具的“名字”选项有两个主要的功用：设置主机名和计算机域名以及设定将使用哪一个域名服务器用于网络上计算机的查找。网络配置工具并不能将一台机器配置为域名服务器。编辑一个域或增添新的信息只需用鼠标左键点击某个域然后键入所需的信息。

3.2.2 操纵主机

可以在HOST管理面板中，增加、编辑和删除 /etc/hosts中的记录 。增加或编辑一条记录的方法是相同的，将会弹出一个编辑对话框，只在其中输入新的信息然后点击 Done就行了。图3-2是一个增加/编辑主机的例子。

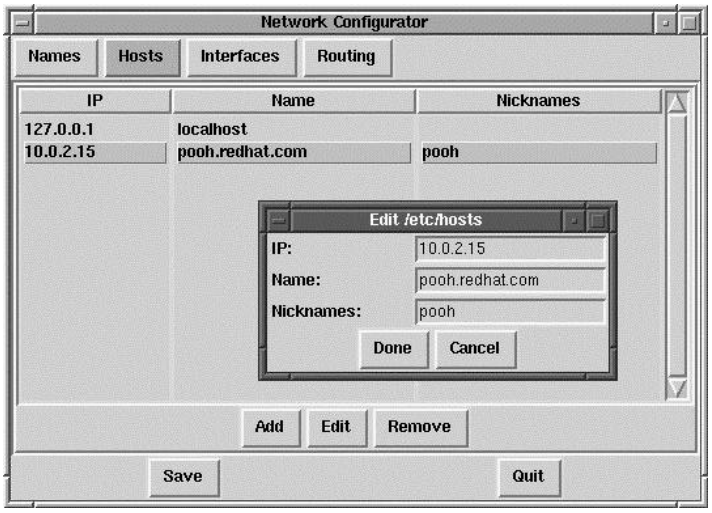

图 3-2

China-pub.com
下载

3.2.3 增加网络接口设备

如果你是在安装了Red Hat Linux之后又装了一块网络设备，或者没有在安装时配置以太网卡，点几下鼠标就可以完成所需的配置。在开始添加一个新接口以前点击主面板上的Interface，将会调出列有若干已配置设备及其选项的列表，见图 3-3。

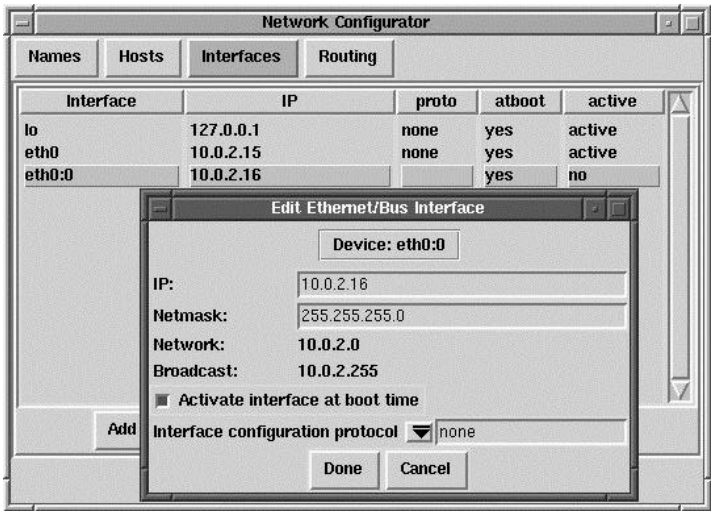

图 3-3

要增加一个设备，则首先点击 Add 按钮，然后选择要增加的接口类型，如图3-4所示。

图 3-4

注意 现在netcfg中有一个clone按钮。用以“克隆”一个已存在的接口设备。通过使用克隆出来的接口设备，就可以方便地为一台膝上电脑配置一个以太网卡用于办公局域网，而另一个用于家庭局域网。

增加一个PPP设备很简单，只需在图 3-5的对话框中填入电话号码，登入用户、口令。如果你的PPP连接使用PAP 认证，就选择Use PAP authentication。在许多情况下，为建立一条PPP连接还需要一些定制信息。选择 Customize 按钮对指定PPP接口进行硬件、通信和网络设置进行定制修改。

图 3-5

China-pub.com 下载

1. SLIP接口

为了配置一个SLIP接口，必须首先提供电话号码、用户和口令。这将为建立 SLIP连接的chat脚本提供初始化的参数。选择 Done以后将会弹出一个标题为 Edit SLIP Interface 的对话框以供进一步定制该SLIP接口的硬件、通信和网络参数。

2. PLIP 接口

为了给系统增加一个 PLIP接口，只需提供IP 地址、远端IP地址和网络掩码。你还可以选择是否在boot激活该接口设备。关于PPP、SLIP和PLIP详细介绍，请参阅有关章节。

3. Ethernet、Arcnet、Token Ring和Pocket Adaptor 接口

如果要为计算机增加 ethernet、arcnet、token ring或pocket 适配器的话，需要提供如下信息：

Device：由netconfig 根据预先配置的设备来决定的。

IP Address: 为该网络设备设定一个IP地址。

Netmask: 为该网络设备设置网络掩码。

网络和广播地址将根据你输入的 IP地址和网络掩码自动算出。

Activate interface at boot time: 如果希望系统在boot时自动配置该设备的话，就选择该项。

Allow any user to (de)activate interface: 如果希望任何用户均能启停该设备的话，就选择该项。

Interface configuration protocol: 如果你所在的网络上运行有 BOOTP 或DHCP 服务器而且想使用它来配置该接口设备，就选择相应的选项，否则就选择 none。

在输入了新设备的所有信息以后，点击 Done，该设备将会列在Interfaces 列表中，成为一个非活动设备 (活动列将取值为 no)。为了激活该设备，首先单击选择它，然后选择 Activate 按钮。如果它没有被正常激活，则要选择 Edit按钮对它进行重新配置。

3.3 安装网卡

首先要指出，Linux对网卡的支持往往是只针对芯片，所以对某些不是很著名的网卡，往往需要知道它的芯片型号以配置 Linux。比如笔者的Accton网卡，就不存在Linux的驱动程序，但是由于它和NE 2000兼容，所以把它当做NE 2000就可以在Linux下用了。当你有一块网卡不能用，在找 Linux的驱动程序之前一定搞清楚这个网卡用的什么芯片，跟谁兼容，比如3c509、NE2000、etherexpress等等。这样的型号一般都在网卡上最大的一块芯片上印着，记住芯片的型号。

3.3.1 设置网卡模式

最普遍使用而且最好配的网卡也许就是 NE2000兼容卡了，笔者用它来做例子。注意，实际上很多廉价网卡都是 NE 2000兼容的。对于NE2000卡，先要做的一件事情，是将网卡设定为Jumpless模式，而很多现在的网卡缺省都是 PnP模式，这在Windows 95下确实能减少很多麻烦，但是Linux不支持，所以Linux下必须是Jumpless模式。一般所有网卡都带有驱动盘和 DOS下可执行的一个设定程序，用该程序将网卡设为 Jumpless。当然如果是老型号网卡，本来就不是PnP，这项可以忽略。

接下来必须弄清楚网卡的 IO地址和IRQ。这是两个非常重要的网卡驱动参数。继续用原

China-pub.com
下载

来网卡的设定程序，设定完 Jumpless之后，可以软设定 IO和IRQ。到现在，我们了解到手头的网卡是NE 2000兼容，知道了它是在Jumpless模式下，知道了IO地址和IRQ，便可以开始安装了。

3.3.2 配置网卡

Linux系统与Windows 95系统不同的是，它运行在“ 内核 ”上。所谓内核，就是把系统最核心的部分孤立出来编程，将各种驱动程序，内存控制等部分编在一起。与 Windows 95不同，Linux的内核是公开的，而且经常更新，这样便不需要更新整个系统，用户只需要把最新的内核原程序下载下来编译，就可以得到一个支持更多硬件，更多文件系统，更加安全的系统了。所以需要指出， Linux的驱动程序很少有像Windows 95下那样是“ 安装 ”的。 Linux下的驱动程序大多数都是以C程序形式发布，或者在内核中，或者需要用户自己修改内核的源代码。总之要驱动程序运行，最好重新配置编译内核。

如果Linux已经装好了，到底用的是什么内核呢？一般缺省的 Linux内核是从安装盘上来的。这个内核一般包括大多数硬件的驱动程序，比如 NE 2000卡。所以未必一定要重新编译内核，也许现有的内核就可以驱动。下一步就是把网卡插入计算机，观察 Linux能否发现它。这些可以从Linux的启动画面中看出来，如果发现如下的一行：

```
eth0:NE2000 card found at 0x300 using IRQ 05
```

那就说明你很幸运， Linux发现了NE2000卡。

如果Linux没有发现你的网卡，但是你确认你网卡的型号和参数，比如笔者的网卡是NE2000兼容， IO 0x300， IRQ 05。那么可以修改Linux启动文件专门搜索这个设备。这个文件在/etc/rc.d/rc.modules里。对于NE2000，是这样的：

```
#/sbin/modprobe ne io=0x300 # NE2000 at 0x300
```

将最前面的#号去掉，再启动机器。另外该文件里还可以发现对其他系列网卡如 3C系列的autoprobe。如果到现在启动屏幕上也没有出现 eth0:NE2000 card found at 0x300 using IRQ 05这样的消息，那就必须编译内核。

编译内核之前一定要搞清楚网卡的芯片号。比如笔者有一块 SMC的网卡，但是我无法在Linux的内核配置菜单里找到SMC这样的字样， Linux不同于Windows 95，没有那么长一串厂商牌号。我在这个SMC的卡上找到了digital 21140-AE的字样，于是我知道这个卡用的是 DEC 21140-AE芯片，按照这个方法寻找，便找到了驱动。知道了芯片类型，或者兼容类型 (比如NE2000)就可以开始编译内核了。具体的针对不同系统的内核编译推荐先看看有关内核编译的文章，这里不多作介绍。

进入/usr/src/linux运行：

```
make menuconfig
```

进入菜单配置内核。找到 Network Device Support后，选择EtherNet，再选择相应的芯片号。如果是ISA系线的NE 2000，则选择other ISA cards，选择NE2000/NE1000 ISA support。如果是PCI的网卡，就选择PCI ethernet adapters。

注意 PCI卡未必都能这样驱动，很多都不行，得另外找驱动程序，比如 Intel PCI EtherExpress Pro 100等。

China-pub.com 下载

配置完内核后，运行：

```
make dep;
make;
make zlilo
```

如果内核选项过多，会出现过大的情况而无法安装。需要把一些不必要的驱动程序去掉。

通常这样之后再启动就可以发现网卡了。驱动了网卡，下一步就是设定 TCP/IP协议，这些将在下面的小节内讲到。

根据笔者的经验， Linux下NE2000兼容卡都比较好设置。 3com系列的卡也都有支持。至于其他网卡如DEC21140就麻烦得多。另外，有些卡即使用某种芯片也未必能用这个芯片的驱动程序，这种情况就是有驱动程序也不能用，那就需要上网查了。

3.3.3 有关Intel Etherexpress系列卡的配置

Linux内核中有Etherexpress 16的支持，但没有其他卡的驱动程序。在http://cesdis.gsfc.nasa.gov/linux/drivers/eepro100.html里有关于EtherExpress 100B pro的讨论。在那里可以download到一个c源码的驱动程序，编译进内核就可以了。

如果以上方法都试过，而你的网卡的确还是不能被系统识别，那么你只能上网查驱动程序。建议从http://cesdis.gsfc.nasa.gov/linux/drivers/开始寻找Linux网卡的驱动程序。在那里可以发现的针对网卡的驱动有：

```
DEC DC21*4* Tulip chip based cards
3Com PCI Etherlink PCI and EISA cards
Intel EtherExpress Pro100B Pro100+ and Pro10+ PCI
3c515 ISA Fast Etherlink card
SMC EtherPower II (EPIC/100 83c170 chip) driver
RealTek RTL8129/8139 driver
Lite-On lc82c168 PNIC driver (now merged with the Tulip driver)
Macronix MX98713 and ASIX experimental drivers are now merged with the
Tulipdriver).
VIA Rhine (VT86C100A and 3043) driver (now released).
Winbond w89c840 driver (beta test). Note: this driver was written without
official
documentation.
TI ThunderLAN driver (external link -- Caldera/James Banks).
Hewlett Packard 100VG driver updates (external link -- Yaroslav).
Intel EtherExpress Pro/10 PCI driver (remote link).
3c509/3c529/3c579 ISA/MCA/EISA EtherLinkIII driver update. The driver now
detects
multiple cards when loaded as a module.
AMD LANCE/PCnet driver update. The driver is now usable as a loadable module.
Cirrus/Crystal/IBM CS8900 series driver (remote link).
PCI NE2000 driver (local page)
PCI NE2000 updates (remote link)
Intel Etherexpress Pro 100
DEC 21X4* based board
3Com EtherLink III PCI/EISA (3c590 3c592 3c595, 3c597, 3c900 3c905)
Intel EtherExpress Pro/10 PCI9 With i82596 Chip)
```

China-pub.com
下载


```
TI ThunderLan
PCI NE2000
Packet Engines "Yellowfin" G-NIC
SMC EtherPower II (EPIC/100 83c170 chip)
RealTek RTL8129/8139
```

如果这里你还是没有找到相应的网卡驱动程序，建议你上 Internet Newsgroup查找。因为你绝对不是第一个在Linux下用这种网卡的用户，你的问题也许已经有人回答过。在这里，将找到满意的答案。

3.3.4 网卡配置中的一些疑问

1. 如何实现一个网卡，多个IP地址

在Linux下， 一块网卡拥有多个IP地址是可行的， 这种技术称做IP Aliasing。在老版本内核中， 做IP Alias比较繁。新版的就好做多了。在新版的 Linux下， 如果/proc/net/下有alias_types和aliases两个文件，就表明你的内核支持 IP Aliasing， 否则， 重新配制内核。

假设已有IP地址为192.168.1.1， 要增加192.168.1.2和192.168.1.3，可这样做：

```
# ifconfig eth0:0 192.168.1.2
# route add -host 192.168.1.2 dev eth0:0
# ifconfig eth0:1 192.168.1.3
# route add -host 192.168.1.3 dev eth0:1
```

2. 如何安装双网卡

在 slackware 2.0.0以后，网卡缺省的是用 module 方式编译进内核的，而在这种方式下，检测到第一块网卡后就不再进行下一步的检测，所以应该在 /etc/rc.d/rc.modules 里加上/sbin/modprobe ne io=0x300，0x330，这里假设用两块 NE 2000网卡，IO地址分别是 0x300，0x330，以下方法是可行的：

```
#/etc/conf.module
alias eth0 ne
options ne io=0x300,0x330
```

请读者参考 /usr/src/linux/Documentations/modules.txt。

3.4 路由、网关和IPChains

3.4.1 路由和网关的概念

目前的局域网大多基于 Ethernet(以太网)，这是基于IEEE 802.3标准的一种局域网标准。以太网使用“广播”方式收发信息，具体实现就像这样：网络内部每台计算机拥有一个惟一的地址，每一台计算机发送信息时，将接收者的地址包括在数据包头部 (目标地址)。广播方式决定了网络上所有的计算机都能接收到这个数据包，如果一台计算机发现目标地址和本机地址相同，就接受并处理这个数据包，其他的计算机发现目标地址不是本机地址，就把它忽略掉。

但是，这种纯粹的广播方式只适合于小型的局域网，对于 Internet而言，如果也采取上面提到的简单的方式，就意味着 Internet上每一台计算机发出的每一条信息，都要“跑遍”整个Internet，这显然是不可能的。因此，我们必须采取“路由”技术来进行数据包的转发，这相

China-pub.com 下载

当于在数据包“广播”的过程中加入了一些“分捡”措施。例如图 3-6所示的网络连接，有两个星型连接的子网，计算机Ⅰ发送数据给计算机Ⅲ时，路由器 A检测到数据包目标地址在本子网内，就不向外部网络转发数据包。但是，当Ⅰ向计算机Ⅴ发送数据时， A检测到目标地址在本子网外，就将该数据包向外部转发，这个数据包经过一系列类似的转发以后，到达路由器 B，B发现目标地址在本子网内，就接受并向子网内转发这个数据包。最后，计算机Ⅴ收到这个数据包。

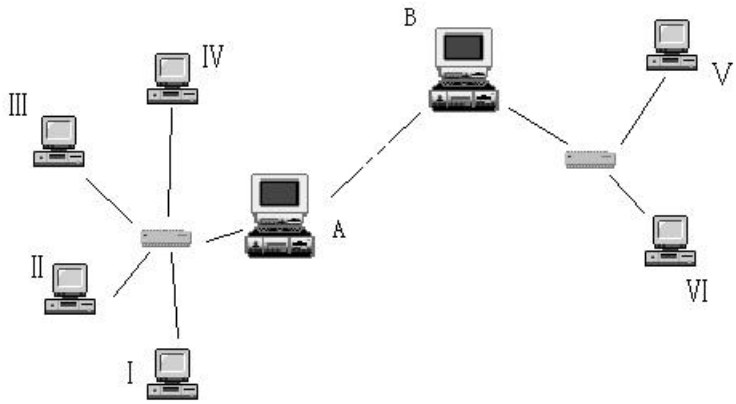

图 3-6

另一个相似的概念是“网关”，在OSI的术语中，网关(gateway)指的是第七层(应用层)的概念，路由器(router)是第三层(网络层)的概念，但是在TCP/IP术语中，gateway和router是一个意思，在一般的Internet文档中，将IP层路由器称为网关。

3.4.2 使用IPChains实现数据包过滤和转发

在Linux kernel 2.2.0中，用ipchains取代ipfwadm，实现IP包的过滤和转发，这也就成为了一个防火墙。下面举出IPChains的具体配置实例。

下面的设置启动了一个基本的 IP转发系统，禁止IP欺骗，广播包，提供本局域网内部的你使用IP伪装连入Internet，这是企业、实验室等单位利用公用 IP上网的通常解决方案。本例中的内部网段使用IP地址192.168.2.0，eth1为内部网段网卡接口， eth0为外部网段网卡接口。

1. 文件的配置

这部分非常简单，安装完系统后，在 /etc/rc.d/目录下创建一个描述文件，命名为ipchains.rules。执行：

```
chmod u+x ipchains.rules
```

确保它为可执行文件。

然后在文件/etc/rc.d/rc.local中添加一行：

```
/etc/rc.d/ipchains.rules
```

以确保每次机器重新启动后即运行所设定的各项防火墙规则。

2. ipchains.rules文件的内容

下面是需要设置的相关内容：

```
#!/bin/sh
echo "Starting ipchains firewall rules..."
```

China-pub.com
下载


```
# refresh all firewall rules
/sbin/ipchains -F forward
/sbin/ipchains -F input
/sbin/ipchains -F output

# setup default firewall rules (deny)                    缺省防火墙设置)
/sbin/ipchains -P forward DENY
/sbin/ipchains -P input ACCEPT
/sbin/ipchains -P output ACCEPT
external_interface=a.b.c.d

# setup Loopback interface
/sbin/ipchains -A input -j ACCEPT -i lo
/sbin/ipchains -A output  -j ACCEPT -i lo

# disabling IP spoofing (                                禁止IP欺骗)
/sbin/ipchains -A input -j DENY -i eth0 -s 192.168.0.0/16
/sbin/ipchains -A input -j DENY -i eth0 -d 192.168.0.0/16
/sbin/ipchains -A output -j DENY -i eth0 -s 192.168.2.0/16
/sbin/ipchains -A output -j DENY -i eth0 -d 192.168.2.0/16
/sbin/ipchains -A input -j DENY -i eth0 -s $external_interface/32
/sbin/ipchains -A output -j DENY -i eth0 -d $external_interface/32
#refuse packets claiming to be to or from the loopback interface
/sbin/ipchains -A input -j DENY -i eth0 -s 127.0.0.0/8
/sbin/ipchains -A input -j DENY -i eth0 -d 127.0.0.0/8
/sbin/ipchains -A output -j DENY -i eth0 -s 127.0.0.0/8
/sbin/ipchains -A output -j DENY -i eth0 -d 127.0.0.0/8
#refuse broadcast address source packets (                禁止广播包)
/sbin/ipchains -A input -j DENY -i eth0 -s 255.255.255.255
/sbin/ipchains -A input -j DENY -i eth0 -d 0.0.0.0
#refuse multicast/anycast/broadcast address
/sbin/ipchains -A input -j DENY -i eth0 -s 240.0.0.0/3

#forwarding all internal traffic (                        转发内部包)
/sbin/ipchains -A forward -j ACCEPT -i eth1 -s 192.168.2.0/24 -d
192.168.2.0/24

#setup IP Masquerading rules (                            设置IP伪装规则)
echo "1" > /proc/sys/net/ipv4/ip_forward

# add modules for ftp, cuseeme, irc, real audio, etc...
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_quake
/sbin/modprobe ip_masq_irc
/sbin/modprobe ip_masq_user
/sbin/modprobe ip_masq_raudio

#starting IP masquerading (                               设置IP伪装)
/sbin/ipchains -A forward -j MASQ -i eth0 -s 192.168.2.0/24
```

对于最后一部分，如果仅仅希望部分进行IP伪装，也可以个别设置，如：

```
/sbin/ipchains -A forward -j MASQ -i eth0 -s 192.168.2.3/32
/sbin/ipchains -A forward -j MASQ -i eth0 -s 192.168.2.5/32
```