China-pub.com
下载

China-pub.com
下载

第4章 防 火 墙

对重要的计算机网络和数据进行保护是Linux系统的一种十分普通的使用方式。TCP/IP协议的强大功能在于它的开放性和普遍性；但是这些优点在恶意的入侵者攻击计算机网络时便成为了极大的隐患。

当一台计算机的主要功能是实现对TCP/IP网络的保护时，这台计算机就称为Internet防火墙，有时也称为城堡主机（bastion host）。在利用Linux实现防火墙时，主要有以下两种技术：一种是代理（proxy）访问，在这种方法下，用户要访问局域网外的资源必须通过某一台特殊的起中介（或代理）作用的主机来实现；另一种是使用 Linux系统内核来作为过滤路由器（filtering router），在这种方法下，Linux系统内核将根据系统管理员定义的规则来简单地转发、修改或抛弃网络数据。

当Linux作为过滤路由器时，维护内部主机是相当容易的；使用代理则使内部主机的配置变得相当复杂。当然这两种技术在一定程度上可以混合在一起，但是代理总是要引入复杂的附加层。

本章假定读者已经正确地配置了一个或多个网络接口，并且已经定义了到达这些接口的路由。代理防火墙可能被设计成使用一个网络接口来连接阻塞的路由器，但是依赖于 IP转发或伪装（masquerading）的防火墙要求至少有两个网络接口。有关网络配置的信息在本书的其他地方可以找到。

因为安全有效的Linux防火墙可以配置在最先进的计算机（可能使用的是基于 80386的机器）上，因此高利用率的环境可以从强大的硬件中获得不少益处。在这种情况下，不应该再使用基于ISA的网卡，同样，也不要再使用任何基于NE2000芯片的网卡。目前，两种非常流行的PCI以太网设计是Tulip芯片（例如DEC设计的21040和21140以太网芯片）和3Com 905系列。一些基于21140的Tulip芯片设计的网卡不是十分昂贵，而且可以实现 100BaseT的快速以太网。这两种设计都是很容易配置的。

动态主机配置协议（Dynamic Host Configuration Protocol，DHCP）在受保护的网络中起着特别重要的作用。由于某些合理的因素，在防火墙的设计中可以不考虑 DHCP，因为扩展成集中管理的益处在于可以动态地降低网络的复杂性。

UNIX的安全问题在极大程度上依赖于正确地设计防火墙。用户应该密切关注 inetd——Internet超级服务器（Internet super server）驻留程序。应该对该进程的配置文件/etc/inetd.conf进行细致的检查，不需要的服务应该被删除。特别应该去除 shell和login服务，除非它们的存在是必需的。更进一步的考虑是去除telnet和ftp服务，尤其是如果要使用ssh，更是如此。所有其他的服务在可能的情况下，都可以考虑删除。

4.1 虚拟IP地址

为内部网络上的某一台主机虚构 IP地址显然是非常必要的——尤其在越来越难以获得Internet IP地址的今天。如果必须这样做的话，那么“人造”的虚拟IP地址可以从表4-1所示的

China-pub.com
下载

地址中选择。

表4-1 私有IP地址范围

网络类型	网 络
A	10.0.0.0
B	172.16.0.0～172.31.0.0
C	192.168.0.0～192.168.255.0

这些地址都是为私有使用而保留的，它们无法在 Internet上路由。

4.2 内核IP转发和伪装规则

在路由式防火墙被配置到 Linux系统上之前，内核应该理解如何通过网络接口进行数据包（代表其他系统）的转发。通过在 Xwindow系统下运行 netcfg命令可以达到上述要求。启动数据包转发的选项如图 4-1所示。

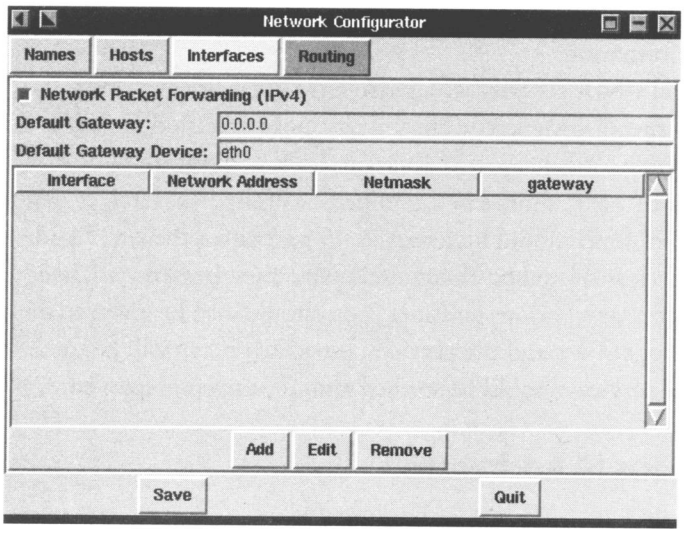


图4-1 网络配置工具

只有在系统重新引导以后，这些选项才会起作用。虽然这看上去有些复杂，但实际操作只是修改了/etc/sysconfig/network文件中的一行。在选项被设置以后，该文件中将可能有如下形式的一个表项：

```
FORWARD_IPV4=true
```

如果表项中取值为false，那么可以手工将其修改为true，这样就可轻松地启动XWindow环境。然而，/etc/sysconfig/network文件仅运行下面的命令，该命令将在系统启动时在内核中启动通过网络接口的转发：

```
echo 1>/proc/sys/net/ipv4/ip_forward
```

要启动数据包转发，用户可以随时运行上述命令。相反，若是将“ 0 ”值写入文件，那么数据包转发将立即被禁止。

在linuxconf工具中也有可用于启动数据包转发的类似选项。

China-pub.com 下载

设置这个选项将把Linux系统转化为一个简单的路由器。然而缺省情况下配置防火墙的通用（也是最安全）方式是禁止所有的数据包转发，只允许明确的路由类型通过。 ipchains管理工具就可用于实现这种修改。例如，为了将缺省的路由行为修改为禁止路由，可以键入如下的命令（这里假定ipchains RPM已经安装在系统中）：

```
ipchains -P forward DENY
```

早期的Linux内核使用一个称为ipfwadm的不同于ipchains的工具来管理内核防火墙资源。只是因为一些设计上的不足，该工具在 Linux 2.2内核中被ipchains所取代。Red Hat Linux 6.0自带一个名为ipfwadm-wrapper的脚本，该脚本可以通过如下形式来实现对早期语法的实现：

```
ipfwadm -wrapper -F -p deny
```

如果防火墙控制必须与运行旧版本内核（如Red Hat Linux 5.2或更低版本）的系统相适应，那么就使用ipfwadm -wrapper脚本。但是，值得注意的是ipfwadm-wrapper脚本和ipchains不能混合使用，只能使用其中之一。

下面的命令可以用于清空遗留在内核中的所有路由规则，确认所有的网络传输均返回为缺省的动作（即禁止路由）：

```
ipchains -F
```

而使用ipfwadm来清空路由规则的语法存在一些细微的差别（转发、输入、输出和记帐规则必须分别清空）：

```
ipfwadm-wrapper -F -f
ipfwadm-wrapper -I -f
ipfwadm-wrapper -O -f
ipfwadm-wrapper -A -f
```

这样，各个独立可接受的路由规则可以被加入到内核之中。例如下面的命令使得所有的主机都可以访问主机1.2.3.4上的Web服务器：

```
ipchains -A forward -s 0.0.0.0/0 -p TCP -d 1.2.3.4/32 http -j ACCEPT -b
```

下面按顺序给出上述命令中各选项的含义描述：

-A forward：该选项定义将规则加入到转发链中。在系统中有三个预定义的链：输入链、输出链和转发链。当然还可以创建其他的新链，但是这个操作过程不在本书中讨论。

-s 0.0.0.0/0：该选项用于指示数据包的源。网络地址 0.0.0.0/0是一个专门的命名，它用来表示数据包可以来自任何IP地址；在忽略-s选项时，默认的地址也是0.0.0.0/0。

-p TCP：指明数据包使用 TCP协议，并且网络只转发该协议的数据包。使用用户数据报协议（User Datagram Protocol，UDP）和Internet消息控制协议（Internet Control Message Protocol，ICMP）的数据包将被拒绝。

-d 1.2.3.4/32：-d选项用于指示目的地址。 1.2.3.4/32是一个标准的Internet地址（即它只代表一台主机）。一个完整的网络可以通过指示网络掩码的位数来定义源地址和目的地址。例如，1.2.3.0/24将允许在整个C类地址1.2.3上的Web传输。读者应该注意地址 1.2.3.4并不是真正的C类Internet地址（实际上它是一个A类地址）；在这里将它作为C类地址使用，只是为了使意义更为清晰。需要强调的是，网络 1.2.3和它的任何派生地址在Internet中都是没有使用的。

http：这是来自文件/etc/services的协议名称，还可以使用数字化的端口号。

-j ACCEPT：-j选项所带的参数用来表示对符合前面选项要求的数据包的动作规定。在上例中，数据包将被接受以用于转发。可以使用的规定动作包括： ACCEPT、DENY、MASQ、

China-pub.com
下载

REDIRECT、REJECT和RETURN。DENY和REJECT类似，但是REJECT将产生一个ICMP目的不可达的消息，而DENY则将数据包抛弃且没有任何附加动作。MASQ允许某一台主机被伪装成其他主机。REDIRECT则允许向远程机器发送的数据包可以在本地接收（这对于透明代理是很有用的）。RETURN表示将缺省的链策略用于这条规则中。

-b：-b选项表示规则是双向的，从而允许数据包可以发向或来自任何一台主机。当然还可以通过将规则添加两次的方法来实现 -b选项的行为，只需要在第二次添加时交换源和目的地址即可。-b选项实际上就是插入了两条规则。如果使用 ipchains -L命令来显示链的内容，则可以看到向前和向后的两条规则。

而ipfwadm命令在语法上有所不同：

```
ipfwadm -F -a accept -P tcp -S 0.0.0.0/0 -D 1.2.3.4/32 http -b -o
```

上述命令所使用的选项的含义如下：

-F：该选项指明命令将对转发规则产生影响（与输入、输出和记帐规则相反）。

-S 0.0.0.0/0：-S选项定义数据包源地址。网络地址 0.0.0.0/0是一个专门的命名，用于表示数据包可以来自任何IP地址。

-D 1.2.3.4/32：-D选项指明目的地址。1.2.3.4/32是一个标准的Internet地址（即它只代表一台主机）。一个完整的网络可以通过指示网络掩码的位数来定义源地址和目的地址。例如，1.2.3.0/24将允许在整个C类地址1.2.3上的Web传输。

http：这是来自文件/etc/services的协议名称，还可以使用数字表示的端口号。

-b：-b选项表示传输是双向的，即，也允许反向的规则。

-o：将匹配的数据包日志记录到/var/log/messages中。

与前面介绍的规则相似的列表可以被加入到 Linux内核中，对该列表进行自上而下地分析。当两个矛盾的命令输入以后或一个数据包可以匹配多个规则时，那么最先键入的规则将用于决定对数据包所实施的动作。使用这种形式的命令，可以在受保护的主机中打开一个通道，为需要的服务提供相应的路径，同时又可以阻塞那些潜在的具有威胁的数据包。

如果防火墙的任意一端与 Internet相连，那么在防火墙两侧的主机都应该具备有效的Internet IP地址。当然，存在一种方法，可以在不与 Internet直接相连的一端使用假造的地址，这种方法很有作用，可以保护 Internet上的IP地址资源 —— 因为只需要一个真实的地址。这项技术称为IP伪装，所使用的命令如下：

```
ipchains -P forward DENY
ipchains -F
ipchains -A forward -j MASQ -s 192.168.1.0/24 -d 0.0.0.0/0
```

使用ipfwadm的命令格式如下：

```
ipfwadm-wrapper -F -p deny
ipfwadm-wrapper -F -f
ipfwadm-wrapper -I -f
ipfwadm-wrapper -O -f
ipfwadm-wrapper -A -f
ipfwadm-wrapper -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
```

伪装技术缺省时特别要求关闭路由，这是通过上述命令表中的第一条命令来实现的。而命令列表中最后一次对ipchains（或ipfwadm）命令的使用则用来建立伪装规则。

在上面的例子中，我们假设某一个网络接口有在 192.168.1 C类网络上的地址。这样，在

China-pub.com 下载

该网络上的主机应该将它们缺省的网关配置为伪装服务器（ masquerade server）的网络接口IP地址。

通常需要对通过伪装路由传输的数据包进行重写，以便数据包的源网络地址能够和位于伪装服务器的缺省路由上的网络接口的 IP地址一致（源端口也被修改）。内核将追踪所有的伪装连接。当远程的服务器向伪装服务器返回响应时，内核又用原来的 IP地址和端口重写返回的数据包，并将数据包转发到内部网络中。外部主机将认为防火墙是所有传输的起始位置，但是内部的主机并不了解这种差别。

对大多数协议而言，伪装都可以正常工作，但是并非所有协议都如此。例如，在伪装服务的网络中，FTP（在激活模式下）就将终止工作。当然，对于这些协议而言，伪装服务器可以作为内核模块来使用。下面就是调用所有其他可用伪装协议的命令：

```
depmod -a
modprobe ip_masq_autofw
modprobe ip_masq_cuseeme
modprobe ip_masq_ftp
modprobe ip_masq_irc
modprobe ip_masq_mfw
modprobe ip_masq_portfw
modprobe ip_masq_quake
modprobe ip_masq_raudio
modprobe ip_masq_user
modprobe ip_masq_vdolive
```

显然，只有所需的伪装模块应该被加载，其他的模块是可以被忽略的。加载多余的内核模块将使得内核过大，并且更容易受到攻击。

如果一个网络应用程序在伪装服务器的后面停止了工作，则可能需要在某个端口下创建新的服务。如果这样的服务位于一个任意的端口（并非准确定义的端口）上，那么应用程序与伪装服务一起正常工作的机会是很渺茫的。在本章的后面，我们将讨论使用代理来访问这种网络中的服务器。

伪装规则可以和其他转发规则共同存在，它们都可以应用在真实的 Internet IP地址上。下面的例子实现了所有的外部主机都可以访问在 1.2.3网络上的Web和电子邮件服务，但是其他所有来自1.2.3内部主机的传输都被伪装了：

```
ipchains -P forward DENY
ipchains -F
ipchains -A forward -s 0.0.0.0/0 -p TCP -d 1.2.3.0/24 http -j ACCEPT -b
ipchains -A forward -s 0.0.0.0/0 -p TCP -d 1.2.3.0/24 smtp -j ACCEPT -b
ipchains -A forward -j MASQ -s 1.2.3.0/24 -d 0.0.0.0/0
```

这段命令若使用ipfwadm，则如下所示：

```
ipfwadm -F -p deny
ipfwadm -F -f
ipfwadm -I -f
ipfwadm -O -f
ipfwadm -A -f
ipfwadm -F -a accept -P tcp -S 0.0.0.0/0 -D 1.2.3.0/24 http -b
ipfwadm -F -a accept -P tcp -S 0.0.0.0/0 -D 1.2.3.0/24 smtp -b
ipfwadm -F -a m -S 1.2.3.0/24 -D 0.0.0.0/0
```

这段代码将只打开内部网络服务器上的 HTTP和SMTP端口；其他端口都被阻塞。在这些

China-pub.com
下载

端口上的连接使用的是内部网络中真实的IP地址。任何其他向外的通信都将被伪装。

注意，在上面的代码中，最后三条命令的顺序是很重要的。它们的顺序可以被改写成如下的形式：

```
ipchains -A forward -j MASQ -s 1.2.3.0/24 -d 0.0.0.0/0
ipchains -A forward -s 0.0.0.0/0 -p TCP -d 1.2.3.0/24 http -j ACCEPT -b
ipchains -A forward -s 0.0.0.0/0 -p TCP -d 1.2.3.0/24 smtp -j ACCEPT -b
```

或者是使用更早的语法形式：

```
ipfwadm -F -a m -S 1.2.3.0/24 -D 0.0.0.0/0
ipfwadm -F -a accept -P tcp -S 0.0.0.0/0 -D 1.2.3.0/24 http -b
ipfwadm -F -a accept -P tcp -S 0.0.0.0/0 -D 1.2.3.0/24 smtp -b
```

在上面这种顺序之下，将不再允许 Web或电子邮件的流量通过；这是因为规则是自上而下的，所以所有的连接都将被伪装。

同时，防止IP欺骗（IP spoofing，即主机通过故意错误地使用自己的IP地址以获取通常被否认的访问权限）也是很重要的，可以通过简单地对内核直接配置来启动源地址确认（ Source Address Verification）来实现对IP欺骗的防止。为了建立源地址确认和防止欺骗，将清单 4-1的脚本代码段放置在启动文件中，并且该启动文件在网络设备被启动后执行。

清单4-1 源地址确认脚本

```
# This is the best method: turn on Source Address
# Verification and get spoof protection on all
# current and future interfaces.

if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]; then
  echo -n "Setting up IP spoofing protection..."
   for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
      echo 1 > $f
   done
   echo "done."
else
   echo PROBLEMS SETTING UP IP SPOOFING PROTECTION. BE WORRIED.
   echo "CONTROL-D will exit from this shell and continue system startup."
   echo
   # Start a single user shell on the console
   /sbin/sulogin $CONSOLE
fi
```

作为一条通用的规则，只有作为服务器运行的主机应该得到真实的 Internet IP地址；在客户端应用程序允许的情况下，大多数作为客户端使用的主机都应该被伪装。

本书2.5节有一个伪装与DHCP集成的、完整的简单配置示例。

在IPCHAINS-HOWTO目录下可以找到更多关于 Linux防火墙设计的信息（读者可以从本书所附的光盘中找到该目录，也可以从站点 ftp://tsx-11.mit.edu/pub/linux/docs/HOWTO处找到）。

最后一点应该强调的是，伪装并不是简单的安全特征。伪装技术还可用于多台主机提供服务却只有一个IP地址的情况。例如，在拨号PPP连接中便经常使用伪装技术。

4.3 tcpd

Red Hat Linux 6.0中包含了一个简单的工具： tcpd，以根据主机或网络地址来控制对服务的访问。该工具提供了简单而又强大的可选择应用的访问限制机制。

China-pub.com 下载

tcpd的两个主要配置文件是/etc/hosts.deny和/etc/hosts.allow。唯一可激活tcpd程序的位置在/etc/inetd.conf文件中（不能使用命令行调用tcpd）。

如果一台Linux主机的/etc/hosts.deny文件中包含如下一行：

```
in.telnetd:ALL except 1.2.3.4
```

那么，除了1.2.3.4主机以外，其他主机的telnet访问都将被拒绝；同样相似的表项可以拒绝FTP和Berkeley远程工具：

```
in.ftpd: ALL except 1.2.3.4
in.rshd: ALL except 1.2.3.4
in.rlogind: ALL except 1.2.3.4
```

作为另一种选择，使用如下形式的表项可以拒绝 tcpd控制的所有协议：

```
ALL: ALL EXCEPT 1.2.3.0/255.255.255.0, \
  5.6.7. \
  .allowed.com \
  .permitted.org
```

注意在上面的表项中所使用的地址。可以使用带有网络掩码的网络地址，也可以使用匹配的域名（此时需要有逆向查找）。关于/etc/hosts.deny文件及其语法的其他信息可以通过在shell提示符下键入man 5 hosts_access命令来获得。

完全拒绝对本地服务的访问是设计防火墙时最安全的选择。如果不能这样做，那么可以按照上面的形式来配置tcpd，以拒绝大多数机器的访问。

显然，tcpd工具的通用配置并没有控制在防火墙上的网络接口之间的路由传输，它仅仅保护了本地的服务。在这里介绍tcpd只是为了确保城堡主机的安全。

准确地确定tcpd控制的内容是很简单的。例如，在 /etc/inetd.conf文件中的telnet配置表项如下：

```
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

所有在/etc/inetd.conf文件中的服务，如果它们的配置表项中包含了对 tcpd的引用（如上所示），则它们受 tcpd的控制。注意，有一些服务是不受 inetd控制的，包括 SMTP、httpd、Postgres等等。

4.4 TIS防火墙工具箱

来自受托信息系统（ Trusted Information System，TIS）的防火墙工具箱（ FireWall ToolKit，FWTK）是一个自由软件包，它包含了可用于包括防火墙设计在内的多种网络功能的一套小程序。

当两台主机通过路由式防火墙进行通信时，任意一台主机都没有必要了解连接的机制。虽然主机产生的数据包将被防火墙所检测（并有可能作细微的改动），但是主机应该清楚城堡主机的存在。

另一方面，使用代理防火墙时，两台主机中至少应有一台主机了解防火墙的位置以及用于访问的方法。

FWTK包括了一些小程序，可以实现包括 FTP、Telnet、HTTP（Web）、SMTP、Berkeley rlogin和X在内的一些TCP/IP套件服务的代理。同时，还包括对防火墙设计和 UNIX通用管理很有用的一个通用代理。

China-pub.com
下载

但是，TIS FWTK并没有被包含在Red Hat Linux中；虽然FWTK是开放式的源码，但它的证书协议比GNU通用证书（GNU Public License，GPL）严格，它限制某些类型的商业应用。FWTK可以从站点http://www.fwtk.org处获得，但是用户必须遵守证书才可以下载该产品。

网络协会（近来获得了TIS）出售一个称为Gauntlet的商用防火墙。Gauntlet的功能十分强大，它包含了FWTK的所有代理特征、与Linux内核相同的数据包转发特征以及其他一些高级特征和图形管理界面。从 FWTK移植到Gauntlet并不困难，然而 Gauntlet并不能在 Red Hat Linux上运行，因此有必要改变系统平台。

最新的FWTK版本（在本书编写过程中）为 fwtk2.1.tar.Z。缺省情况下，代理的应用程序和控制文件将被安装在/usr/local/etc目录下（如该目录不存在，则将被创建）。

在Red Hat Linux中构造FWTK时，crypt( )库函数可能会有问题。以根用户方式使用如下的代码来克服这个问题后，就可以开始编译和安装 FWTK了。

```
tar xvzf fwtk2.1.tar.gz
cd fwtk
sed 's/^AUXLIB=.*/AUXLIB= -lcrypt/
  s/^COPT=.*/COPT= -O2 $(DEFINES) -I\/usr\/include\/db1/
  s/^LDFL=.*/LDFL= -s -static/' Makefile.config.linux > Makefile.config
make
make install
```

注意，在上面的命令代码中，除了进行crypt库的设置以外，还启动了工具箱组件的优化并关闭了调试选项。但这并不是工具箱缺省的构造环境。用户可以在文件 Makefile.config.linux中找到更多的构造配置细节。

下面是将安装到/usr/local/etc中的一些文件：

netperm-table：这是工具箱的主配置文件。

ftp-gw：FTP代理。

http-gw：HTTP（Web）代理。如果需要，可以配置为阻塞 Java和JavaScript。此外，根据URL的形式，还可以将Web请求发送到多个主机。

netacl：它是与tcpd类似的工具，有时它更为灵活，因为它可以根据网络连接来源的不同而执行不同的程序。

plug-gw：通用代理。它允许任意的TCP端口上的出入连接。

rlogin-gw：Berkeley rlogin代理。

tn-gw：telnet代理。

x-gw：XWindow应用程序代理。在XWindow开发库和头文件被安装后才可以构造。

每一个利用FWTK的应用程序可以被配置为限制被允许访问代理的一组主机，或者可能被请求的目标机器，或者两者兼有。代理的行为是通过 netperm-table来配置的，它通常是通过/etc/inetd.conf文件控制激活的，下面是该文件中用于配置 telnet代理的表项：

```
telnet    stream    tcp    nowait.1024    root /usr/local/etc/netacl tn-gw
```

很有意思的是上一行中 nowait所带的参数。inetd缺省配置为允许在一个 60s的周期内激活任意一个服务40次。如果超出了这个限制，那么服务将被终止一段时间（ inetd停止派生该服务进程），并且在/var/log/messages中记录该信息。上面的nowait参数1024增加了这个限定值，这个参数对于比较忙的防火墙而言是十分重要的。

China-pub.com 下载

我们继续配置telnet代理的示例，netperm-table文件中的下面两行用来允许使用代理：

```
netacl-tn-gw:        permit-hosts      *        -exec /usr/local/etc/tn-gw
tn-gw:               permit-hosts      *        -passok  -xok
```

注意，只有在inetd被更新后，inetd.conf中新的配置才能发生作用。这可以通过重新引导系统或向inetd进程发HUP信号来实现。相反，当 netperm-table文件的内容被修改后，并不需要更新inetd或其他进程。这是因为代理每次被调用时都要自己更新，它们将扫描 netperm-table文件中的内容；另一方面，inetd只有在启动或接收到HUP信号时才读它的配置文件。

netperm-table文件自身有一些限制，文件中每一行的最大长度根据操作系统的不同而变化，通常是1024个字符。同时，还没有提供续行的方法（如使用“ \ ”来连接下一行）。这些严格限制的规则使得netperm-table可以很容易实现。

一旦inetd被更新后，试图用telnet与城堡主机发生连接时，将出现如下的提示：

```
tn-gw->
```

在代理的提示界面下只有很少的一些选项，最有用的是打开与另一台主机的连接：

```
tn-gw-> connect another.host.com
Trying 1.2.3.4 port 23...
Connected to another.host.com.
login:
```

这种防火墙的配置并不十分安全，因为它允许任何人在任何地方初始化连接。然而对访问的限制是简单的：

```
netacl-tn-gw: permit-hosts 1.2.3.* 192.168.1.* -exec /usr/local/etc/tn-gw
tn-gw: permit-hosts 1.2.3.* 192.168.1.* -passok -xok -dest { 1.2.3.4
*.remotehost.com }
```

将允许访问的主机列出两次（对 netacl一次，对tn-gw一次）看来是多余的，所以以后将netacl从/etc/inetd.conf中删除应该是安全的。上例中的 netacl是有点多余；但是它可以被配置用于基于远程IP地址的多个服务可执行文件的运行，因此还是可以考虑使用。注意，一行被分割（如上面tn-gw行）是不允许的，这里这样用只是为了清晰起见。

注意，虽然可以允许与城堡主机发生 Telnet连接，但我们不建议用户这样做。如果必须与防火墙进行shell连接，那么最好使用加密的协议（如 ssh）和细致的访问控制。

最后，还应该确保代理能够双向连接防火墙；应该允许向外进行通用的连接，而向内的连接访问应该加以限制。

FTP代理的操作和tn-gw在一定程度上相似，尽管通常不需要使用 netcal来配置。下面一行可用于/etc/inetd.conf文件中以配置FTP代理：

```
ftp  stream   tcp        nowait .1024       root       usr/local/etc/ftp-gwftp-gw
```

在netperm-table文件中相应的用于最大访问许可的表项如下：

```
ftp-gw:        permit-hosts*
```

为了更严格一些的访问限制，可以将表项改为：

```
ftp-gw:        permit-hosts        1.2.3.* 192.268.1.* -dest {1.2.3.4 *.remotehost.com}
```

虽然FTP代理的配置相对比较简单，但是在使用时是很让人头痛的。清单 4-2是使用FTP代理的一次FTP对话：

清单4-2 使用FTP代理的对话

```
$ ftp bastion
```

China-pub.com
下载


```
Connected to bastion.
220 bastion FTP proxy (Version V2.1) ready.
Name (bastion:root): ftp@ftp.redhat.com
331-(----GATEWAY CONNECTED TO ftp.redhat.com----)
331-(220 ProFTPD 1.2.0pre1 Server (ProFTPD) [gonzales.redhat.com])
331 Anonymous login ok, send your complete e-mail address as password.
Password:
230-            Red Hat Software -- FTP Site

        For the following sites, use the appropriate server:

        Red Hat Linux Updates    - ftp://updates.redhat.com
        Red Hat Rawhide          - ftp://rawhide.redhat.com

        Red Hat Contrib Net      - ftp://developer.redhat.com
        For Incoming RHCN RPMS   - ftp://in-rhcn.redhat.com

        User Contributed RPMS    - ftp://contrib.redhat.com
        To Upload Incoming RPMS - ftp://incoming.redhat.com
230 Anonymous access granted, restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp ls -l
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
total 0
-rw-r--r--   1 root     root         4050 Nov 18 16:30 MIRRORS.html
drwxr-xr-x   3 root     root         1024 Nov 15 04:03 XBF
drwxr-xr-x   3 root     root         1024 Nov 15 04:03 XFCom
drwxr-xr-x  10 dledford dledford     1024 Dec  8 11:46 aic
drwxr-xr-x  31 root     root         1024 Dec  1 10:06 home
drwxr-xr-x   6 jbj      jbj          1024 Nov 14 04:03 large-fd
-rw-r--r--   1 root     root      2951395 Dec  9 04;07 ls-lR
-rw-r--r--   1 root     root       421475 Dec  9 04:07 ls-lR.gz
lrwxrwxrwx   1 root     root            1 Oct 21 10:53 pub - .
drwxr-xr-x  10 root     root         1024 Nov 14 04:05 redhat
drwxr-xr-x   7 root     root         1024 Nov 14 04:13 sound
drwx--x--x   2 root     root         1024 Nov 14 04:13 sybase
226 Transfer complete.
ftp get MIRRORS.html
local: MIRRORS.html remote: MIRRORS.html
200 PORT command successful.
150 Opening BINARY mode data connection for MIRRORS.html (4050 bytes).
226 Transfer complete.
4050 bytes received in 1.24 secs (3.2 Kbytes/sec)
ftp quit
221 Goodbye.
```

当与代理的连接建立以后，登录名应按照 user@remotehost来定义。然后，代理将初始化与远程服务器的FTP对话。虽然上面的例子中的各个步骤看上去十分直接，但是对于复杂的图形界面的FTP客户软件而言，主机名 /用户名的操作是十分困难的。同样， FTP代理可以通过多个ftp-gw服务器，但这种情况下的语义是很难以理解的。

HTTP代理有许多特征，这使得它的功能特别强大。下面在 /etc/inetd.conf文件中的一行用

China-pub.com 下载

于运行HTTP代理：

```
http stream tcp nowait.1024 root /ust/local/etc/http-gw http-gw
```

允许访问的netperm-table文件表项如下：

```
http-gw: permit-hosts*
```

Web浏览器必须作一定的配置以使用代理。图 4-2显示了在Netscape Communicator中配置使用代理的相关对话框。

在图4-3所示的代理配置对话框中，插入代理的主机名或 IP地址。http-gw还可以作为FTP和Gopher代理（不要在此试图使用 ftp-gw）。

代理可以使用下面的配置来限制它发向浏览器的内容：

```
http-gw: permit-hosts * -nojava -nojavascript -noactivex
```

HTTP代理根据UPL形式的不同，也可以直接指向不同的服务器：

```
http-gw:      forward   /Home*   -protocol http      -tohost www.tis.com
http-gw:      forward   /pub/*   -protocol ftp       -tohost ftp.uu.net
```

可能plug-gw是最灵活的代理，它允许重定向到任意的 TCP端口，并且在防火墙上有很大的使用潜力。例如：

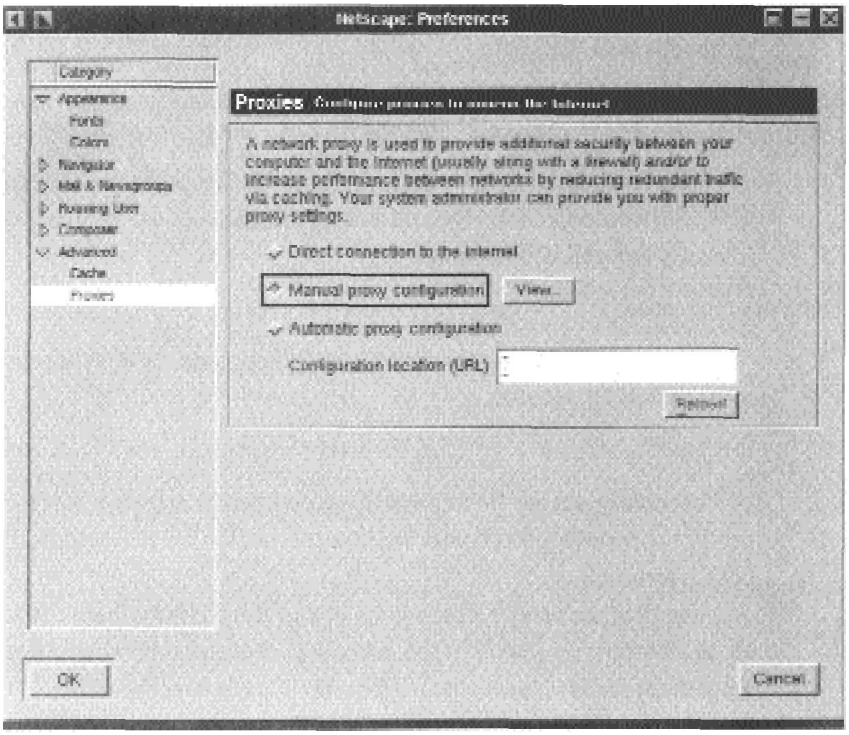


图4-2 Netscape代理优先

```
pop-3 stream tcp nowait.1024 root /usr/local/etc/plug-gw plug-gw pop-3
```

上面取自/etc/inetd.conf文件中的一行直接将POP邮件端口指定为由plug-gw提供服务。

```
plug-gw: port pop-3 * -plug-to mail.remotehost.com -port 110
```

而这个配置指示则将在本地 POP端口连接初始化时自动建立与远程 POP端口的连接。

注意，在上面的配置指示中服务名（ pop-3）和端口号（ 110）将被相互交换地使用（服

China-pub.com
下载

务名在/etc/services文件中定义)。唯一必须使用服务名的地方就是在 /etc/inetd.conf文件中对服务的定义。上面的最后一个选项 -port 110也是可选的，像plug-gw一样，缺省的情况下，该服务初始化时将会被重定向到同号的端口。

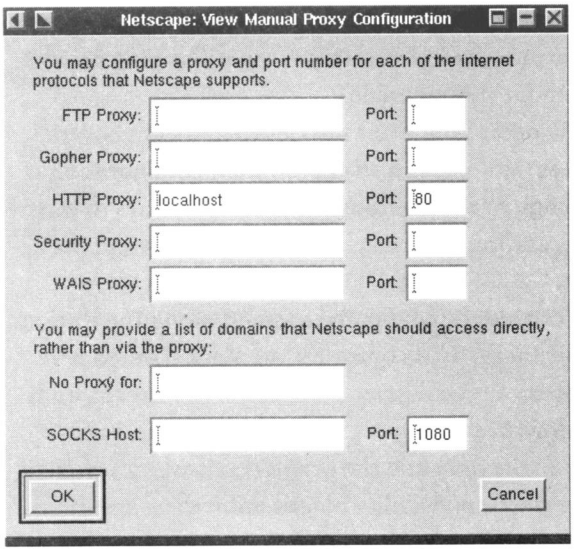


图4-3 Netscape代理配置

还要注意，服务名和端口号都必须作为参数传递给 /etc/inetd.conf文件中的plug-gw；否则plug代理将无法工作。

还有其他一些重要的代理（Berkeley rlogin、x-gw和smap）在本书中没有介绍，但是它们的配置或者与前面介绍的代理相似，或者十分复杂以至超出了本书的讨论范围。

所有的各种代理还可以被配置为在 inetd之外单独运行的应用程序，关于这一点本书不再讨论。

代理防火墙（与路由式防火墙相比）优缺点并存。它最大的优点在于提高了过滤数据和源地址/目标地址的能力；而最大的缺点在于只有了解了代理的结构才能够处理数据在防火墙保护下的数据放置。

然而，不要幻想可以将代理防火墙组件与路由过滤共存。事实上，对于伪装网络（大多数机器不能在Internet中被访问）中访问网络服务而言，代理是一种理想的方法。一种较好的防火墙是结合了上面介绍的全部技术同时又注意保护与网络利益相关的安全考虑。

4.5 SSH

SSH协议及实现 SSH的配套程序给 TCP/IP网络中主机提供了安全加密的网络通信通道。SSH主要是面向UNIX系统的，但是也可用于MS-Windows、VMS和OS/2的版本。

因为SSH使用了强壮的加密技术，又由于法律的限制，所以各国的发布很复杂。例如，起始于美国的 SSH在法国、俄罗斯、伊拉克、巴基斯坦以及其他国家是完全被禁止使用的，也就是说，它不能以任何形式离开起始国。不要过于热心使用这个应用程序；在用户安装之前还应该理解并遵守本国有关加密的法律法规。

SSH在Internet上的主页为http://www.ssh.fi（公司的基地在芬兰）。SSH的拷贝在此可以找

China-pub.com 下载

到，用户可以下载也可以购买。

注意，最新的SSH版本（2.0或更高）只对非商业使用来说是免费的；如果用户为了商业利益而使用它，则需要支付许可证的费用。

对于SSH的不同版本，存在着一些混乱的情况。最近对 SSH进行了一次较大规模的重写，在本书编写时的SSH最新版本为2.0.11。但是许多SSH1的安装仍然在使用中（虽然在老版本中存在许多漏洞和安全隐患）。更严重的问题是 SSH1的客户不能在SSH2服务下工作，因此SSH的客户和服务的主版本号必须一致。如果用户必须使用 SSH1，那么使用1.2.26版本，并且在希望使用kerberos时应用kerberos安全补丁。

SSH中的一个加密技术是公用密匙（public key cryptography）技术（在SSL和PGP中也使用）。理解SSH的机制并不十分重要，因此只需对它进行简单的介绍。在大多数人考虑习惯的密码时，通常会考虑一个用于解锁和破译的单独的关键字（或称密匙）。公用密匙并不只依赖于一个密匙，而是一对密匙 —— 一个是公用密匙，另一个是私有密匙。当使用其中一个对信息加密时，只能用另一个来对信息解密。通常公用密匙对外人是自由发布的，而私有密匙则是保密的。当两个用户相互交换信息时，他们应首先交换公用密匙；同时用公用和私有密匙对信息加密后发出。这项技术使得不用交换解锁密码就实现了安全的解密；而使用单一口令的设计是无法实现这一点的。 SSH对于这一套密匙是非常小心的，一段主机 /服务器密匙（host/server key）将每隔一小时就被改变并重建一次；这是为了防止当服务器设备被捕获后，密匙不被获得。在SSH中还可以使用其他一些加密技术，包括 IDEA、Triple DES和RSA等，但是这些技术对用户都过于透明。

SSH还可以在安全连接之上透明地转发来自远程图形 XWindow客户的通信（这将由xauth和环境变量DISPLAY自动配置）。在远程服务器上运行 XWindow应用程序但在本地显示器上显示，这通常并不需要作更多的配置，对于 XWindow的用户而言是很方便的。

从SSH发布站点并不能够获得SSH的RPM格式；只有用tar和gzip压缩打包格式的C语言源代码可用。如果用户已经获得了 SSH的1.2.26版本，那么在Red Hat Linux 6.0下可以作为超级用户运行如清单4-3所示的命令代码来编译和安装SSH。

清单4-3 构造SSH

```
# The below call to sed could also be written as:
# sed '/^#.*UTMPX/s/^/\/\/' config.h  config.h.new
#
tar xvzf ssh-1.2.26.tar.gz
cd ssh-1.2.26
LDFLAGS="-s" CFLAGS="-O2" ./configure
sed '556d' config.h > config.h.new
mv -f config.h.new config.h
make
make install
```

上述过程将把大多数内容都安装到 /usr/local/bin目录下，但是如果系统中已经安装了SSH1，则sshd（服务进程）将安装到/usr/local/sbin目录下。此时， sshd驻留程序应该被重新启动。ssh和/或sshd的路径可能没有加入到shell的搜索路径中；如果出现这样的问题，可以将路径加入到文件/etc/profile中。如果用户希望在系统启动时开始 SSH服务，那么应该将命令加到系统的rc文件中（至少是/etc/rc.d/rc.local文件中）。

China-pub.com
下载

安装SSH软件到另一台作为客户的主机中。如果机器只是 SSH客户机，那么没有必要运行sshd，只需要将ssh和scp二进制文件从服务器拷贝到客户机即可。作为超级用户，键入如下的命令：

```
ssh remotehost.com
```

remotehost.com是一个运行SSH服务的完整的标准主机名。 SSH将提示输入远程机器的超级用户口令，然后将打开一个UNIX shell。

SSH使用与Berkeley远程工具相似的命令行参数，并且包含 rsh和rlogin的功能。例如可以使用如下命令作为一个没有启动连接的用户登录：

```
ssh -l luser remotehost.com
```

这样在一个远程系统中用任何有效登录 ID取代了luser。SSH将提示输入远程的口令，然后打开一个shell。除了打开一个交互式的连接外，还可以使用 SSH在远程主机上运行单独的命令。例如，键入如下的命令：

```
ssh -l luser remotehost.com "df;ps aux"
```

这时，远程主机将提交关于磁盘空闲空间的报告，并产生进程列表（只有对 ps命令使用BSD语义的系统例如Linux才能正确运行），然后关闭连接。

数据还可以通过管道发向或来自 ssh，即使有口令提示时，也是如此。考虑下面两条命令：

```
cd /home/httpd/html; tar cf - * | \
  ssh remotehost.com "cd /home/httpd/html; tar xvf -"

ssh remotehost.com "cd /usr/local/bin; tar cf - *" | \
  (cd /usr/local/bin; tar xvf -)
```

这两条命令将在本地和远程主机上启动 tar进程使得数据交换通过 ssh连接实现。第一条命令将本地主机/home/httpd/html目录下的内容拷贝到远程主机的同一位置；第二条命令则将远程主机/usr/local/bin目录下的内容拷贝到本地。

上面的例子可能是用于在两个 UNIX系统之间完整地移动整个目录层次结构的最方便的方法。用户应该多研究多实践这些方法。

当然，还可以采用如下的方法更简单地使用 ssh管道技术：

```
ssh remotehost.com "cat /etc/passwd"| grep luser
```

这条命令将给出用户 luser在远程主机的口令文件中的表项。

采用与Berkeley rcp相似的语义来使用scp命令可以在两台主机间通过加密连接实现文件的传输。但是，在文件传输前，应该知道远程文件的绝对路径名。例如：

```
scp /local/file.txt remotehost.com:/long/paths/make/this/hard.txt
```

如在主机名前加上用户名和“ @ ”符号，那么文件还可以作为另一个用户的属主来传输，例如：

```
scp /local/file.txt luser@remotehost.com:/long/paths/make/this/hard.txt
```

当然，文件传输也可以在另一方向上进行：

```
scp luser @ remotehost.com:/long/paths/make/this/hard.txt /local/file.txt
```

最后还要再对scp简单介绍如下。如果不涉及本地系统中的文件，那么传输将不会进行。如果拷贝的目标是一个目录的话，那么可以定义多个拷贝源文件（同 cp命令一样）。此外，远

China-pub.com 下载

程文件中不能使用通配符，因为远程系统没有可用于解释通配符的 shell（换言之，命令scp remote:/root/* /root将不会工作）。如果，在scp命令中使用了-p选项，那么在拷贝时将保留文件的修改时间、访问时间和权限（但是若远程系统采用的是 DOS FAT文件系统，则将失败）。-r选项将实现递归地拷贝整个目录结构，但是将丢失符号链接 —— 建议使用前面的tar/ssh例子来实现高可靠性的拷贝。目前，安全的FTP工具将在SSH以后的版本中实现，从而可以在ssh连接之上结合使用ssh和scp来获取整个目录结构，然而这样的工具并不包含在 UNIX SSH 1.2.26发布中。还要注意的是，与telnet和rlogin不同的是，ssh和scp允许超级用户在远程登录使用。

/etc/sshd_config和/etc/ssh_config文件分别控制SSH服务器和客户的行为，下面是值得关注的一些选项。

在/etc/sshd_config文件中：

AllowHosts *.permitted.org 1.2.3.* 1.?.2.3：本选项允许使用ssh连接的机器（可以使用通配符）。

AllowGroups groupname1 groupname2 ：本选项规定 ssh连接只能来自所定义的组（/etc/group文件规定）中的用户。

DenyHosts *.goaway-spammer.com 5.6.7.* 8.?.9.10：本选项不允许使用 ssh连接的机器（可以使用通配符）。

DenyGroups groupname1 groupname2 ：本选项规定 ssh连接不能来自所定义的组（/etc/group文件规定）中的用户。

在/etc/ssh_config文件中：

Compression yes：本选项将使得在SSH客户和服务器之间交换的数据被压缩。这可能会减慢在快速网络中传输的速度。只应在每个字节都记数的有限带宽环境中使用。通过在 ssh命令中使用-C选项也可以实现压缩。

CompressionLevel 6：压缩的级别可以被设置为从1到9（与gzip工具类似）。级别1是最快的，但是数据并没有得到很好的压缩；级别 9则非常慢，但是数据几乎被完全压缩。缺省的压缩级别为6。

ssh命令中的-g和-L选项对于防火墙设计特别有用。这些选项允许在安全的连接上将 TCP端口转发到远程主机。当远程的 shell对话激活时，下面的命令将在 2300端口上实现加密的telnet连接（与TIS FWTK中的plug-gw十分相似的作用）：

```
ssh -L 2300:remotehost.com:23 -g remotehost.com
```

当用户远程连接到本地主机的2300端口时，连接将被加密再发送到远程，然后转发到命令行中定义的目的端口（23）。

SSH的最后一个问题是如何使得在使用时不要求提供口令。在 Berkeley远程工具中，正确地配置.rhosts文件可以实现无口令的连接。 SSH也可以使用.rhosts文件，但是在缺省情况下是不使用的。支持无口令登录的一种安全方法是利用公用密匙密码术；它的设置很简单。

首先在本地主机作为超级用户运行如下的命令来产生一对密匙；接受所有的缺省设置，但是不要键入口令，只要按回车键就可以了（这不会造成安全漏洞，关于这点将在后面讨论）。

```
ssh-keygen
```

使用下面的命令之一将公用密匙拷贝到远程主机。使用这个语法实现拷贝：

China-pub.com
下载


```
scp ~/.ssh/idetity.pub remotehost.com:/root/.ssh/authorized_keys
```

还可以用如下的语法将密匙附加在远程的 authorized_key文件上：

```
ssh remotehost.com "cat>>~/.ssh/authorized_keys" < ~/.ssh/identity.pub
```

下面的ssh登录就不再需要口令了（scp也会打开）：

```
ssh remotehost.com
```

除非绝对需要，否则最好不要使用无口令的连接 —— 因为无口令连接巧妙地克服了一个重要的UNIX安全屏障。通过 crontabs激活的脚本可能是一个很好的借口。在大多数情况下，最好只让具有极少特权的用户这样做；如果可能，不应该使超级用户可口令登录（即让超级用户远离这种自动操作）。

注意，在这个例子中，如果本地主机被外来攻击者破坏或摧毁，那么远程的主机也很可能遭受同样的命运。也就是说，便利的代价是冒险。

4.6 结束语

防火墙管理员如果认为自己的防火墙是万无一失的，那是很不明智的。任何一个有决心的攻击者都不会把防火墙的安全特征视为不可逾越的障碍，而是看成一点小小的麻烦。只有真正的小心谨慎才可以有效地对敌对攻击进行防御。

管理员应该特别注意以下列出的每一个系统组成部分：

- 在/etc.inetd.conf文件中每一个服务表项的功能，了解服务被关闭的原因。
- ps aux命令报告的所有运行的进程（可能是由欺骗内核伪造的）。
- 允许的路由源地址和目的地址，包括文档。
- 所有通过防火墙的未加密通信通道，以及它们不构成冒险的原因。
- 针对平台提出的Red Hat Errata软件包的安装情况。

此外，C编译器和Linux内核源码不允许驻留在防火墙上（事实上，不允许任何开发工具和无关的软件包安装在防火墙上）。

如果是在一个人人遵纪守法的社会中，这样令人讨厌的安全预防是完全没有必要的：无意识的破坏会得到纠正；而恶意的行为是不可思议的。但是不幸的是我们生活的社会不是这样，因此下面这句话是很明智的说法：好篱笆才会有好邻居。