第十七章 网络系统

When it comes to networking, there is probably nothing that cannot be done with Linux. Linux is used to build all sorts of networking systems and appliances, including firewalls, routers, name servers, NAS (Network Attached Storage) boxes and on and on.

当谈及到网络系统层面，几乎任何东西都能由 Linux 来实现。Linux 被用来创建各式各样的网络系统和装置， 包括防火墙，路由器，名称服务器，网络连接式存储设备等等。

Just as the subject of networking is vast, so are the number of commands that can be used to configure and control it. We will focus our attention on just a few of the most frequently used ones. The commands chosen for examination include those used to monitor networks and those used to transfer files. In addition, we are going to explore the ssh program that is used to perform remote logins. This chapter will cover:

被用来配置和操作网络系统的命令数目，就如网络系统一样巨大。我们仅仅会关注一些最经常 使用到的命令。我们要研究的命令包括那些被用来监测网络和传输文件的命令。另外，我们 还会探讨用来远端登录的 ssh 程序。这章会介绍：

* ping - Send an ICMP ECHO\_REQUEST to network hosts
* ping - 发送 ICMP ECHO\_REQUEST 软件包到网络主机
* traceroute - Print the route packets trace to a network host
* traceroute - 打印到一台网络主机的路由数据包
* netstat - Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships
* netstat - 打印网络连接，路由表，接口统计数据，伪装连接，和多路广播成员
* ftp - Internet file transfer program
* ftp - 因特网文件传输程序
* wget - Non-interactive network downloader
* wget - 非交互式网络下载器
* ssh - OpenSSH SSH client (remote login program)
* ssh - OpenSSH SSH 客户端（远程登录程序）

We’re going to assume a little background in networking. In this, the Internet age, everyone using a computer needs a basic understanding of networking concepts. To make full use of this chapter we should be familiar with the following terms:

我们假定你已经知道了一点网络系统背景知识。在这个因特网时代，每个计算机用户需要理解基本的网络 系统概念。为了能够充分利用这一章节的内容，我们应该熟悉以下术语：

* IP (Internet Protocol) address
* IP (网络协议)地址
* Host and domain name
* 主机和域名
* URI (Uniform Resource Identifier)
* URI（统一资源标识符）

Please see the “Further Reading” section below for some useful articles regarding these terms.

请查看下面的“拓展阅读”部分，有几篇关于这些术语的有用文章。

Note: Some of the commands we will cover may (depending on your distribution) require the installation of additional packages from your distribution’s repositories, and some may require superuser privileges to execute.

注意：一些将要讲到的命令可能（取决于系统发行版）需要从系统发行版的仓库中安装额外的软件包， 并且一些命令可能需要超级用户权限才能执行。

**检查和监测网络**

Even if you’re not the system administrator, it’s often helpful to examine the performance and operation of a network.

即使你不是一名系统管理员，检查一个网络的性能和运作情况也是经常有帮助的。

**ping**

The most basic network command is ping. The ping command sends a special network packet called an ICMP ECHO\_REQUEST to a specified host. Most network devices receiving this packet will reply to it, allowing the network connection to be verified.

最基本的网络命令是 ping。这个 ping 命令发送一个特殊的网络数据包，叫做 ICMP ECHO\_REQUEST，到 一台指定的主机。大多数接收这个包的网络设备将会回复它，来允许网络连接验证。

Note: It is possible to configure most network devices (including Linux hosts) to ignore these packets. This is usually done for security reasons, to partially obscure a host from a potential attacker. It is also common for firewalls to be configured to block IMCP traffic.

注意：大多数网络设备（包括 Linux 主机）都可以被配置为忽略这些数据包。通常，这样做是出于网络安全 原因，部分地遮蔽一台主机免受一个潜在攻击者地侵袭。配置防火墙来阻塞 IMCP 流量也很普遍。

For example, to see if we can reach linuxcommand.org (one of our favorite sites ;-), we can use use ping like this:

例如，看看我们能否连接到网站 linuxcommand.org（我们最喜欢的网站之一）， 我们可以这样使用 ping 命令：

[me@linuxbox ~]$ ping linuxcommand.org

Once started, ping continues to send packets at a specified interval (default is one second) until it is interrupted:

一旦启动，ping 命令会持续在特定的时间间隔内（默认是一秒）发送数据包，直到它被中断：

[me@linuxbox ~]$ ping linuxcommand.org

PING linuxcommand.org (66.35.250.210) 56(84) bytes of data.

64 bytes from vhost.sourceforge.net (66.35.250.210): icmp\\_seq=1

ttl=43 time=107 ms

64 bytes from vhost.sourceforge.net (66.35.250.210): icmp\\_seq=2

ttl=43 time=108 ms

64 bytes from vhost.sourceforge.net (66.35.250.210): icmp\\_seq=3

ttl=43 time=106 ms

64 bytes from vhost.sourceforge.net (66.35.250.210): icmp\\_seq=4

ttl=43 time=106 ms

64 bytes from vhost.sourceforge.net (66.35.250.210): icmp\\_seq=5

ttl=43 time=105 ms

...

After it is interrupted (in this case after the sixth packet) by pressing Ctrl-c, ping prints performance statistics. A properly performing network will exhibit zero percent packet loss. A successful “ping” will indicate that the elements of the network (its interface cards, cabling, routing and gateways) are in generally good working order.

按下组合键 Ctrl-c，中断这个命令之后，ping 打印出运行统计信息。一个正常工作的网络会报告 零个数据包丢失。一个成功执行的“ping”命令会意味着网络的各个部件（网卡，电缆，路由，网关） 都处于正常的工作状态。

**traceroute**

The traceroute program (some systems use the similar tracepath program instead) displays a listing of all the “hops” network traffic takes to get from the local system to a specified host. For example, to see the route taken to reach slashdot.org, we would do this:

这个 traceroute 程序（一些系统使用相似的 tracepath 程序来代替）会显示从本地到指定主机 要经过的所有“跳数”的网络流量列表。例如，看一下到达 slashdot.org 网站，需要经过的路由 器，我们将这样做：

[me@linuxbox ~]$ traceroute slashdot.org

The output looks like this:

命令输出看起来像这样：

traceroute to slashdot.org (216.34.181.45), 30 hops max, 40 byte

packets

1 ipcop.localdomain (192.168.1.1) 1.066 ms 1.366 ms 1.720 ms

2 \* \* \*

3 ge-4-13-ur01.rockville.md.bad.comcast.net (68.87.130.9) 14.622

ms 14.885 ms 15.169 ms

4 po-30-ur02.rockville.md.bad.comcast.net (68.87.129.154) 17.634

ms 17.626 ms 17.899 ms

5 po-60-ur03.rockville.md.bad.comcast.net (68.87.129.158) 15.992

ms 15.983 ms 16.256 ms

6 po-30-ar01.howardcounty.md.bad.comcast.net (68.87.136.5) 22.835

...

In the output, we can see that connecting from our test system to slashdot.org requires traversing sixteen routers. For routers that provided identifying information, we see their host names, IP addresses and performance data, which includes three samples of round-trip time from the local system to the router. For routers that do not provide identifying information (because of router configuration, network congestion, firewalls, etc.), we see asterisks as in the line for hop number two.

从输出结果中，我们可以看到连接测试系统到 slashdot.org 网站需要经由16个路由器。对于那些 提供标识信息的路由器，我们能看到它们的主机名，IP 地址和性能数据，这些数据包括三次从本地到 此路由器的往返时间样本。对于那些没有提供标识信息的路由器（由于路由器配置，网络拥塞，防火墙等 方面的原因），我们会看到几个星号，正如行中所示。

**netstat**

The netstat program is used to examine various network settings and statistics. Through the use of its many options, we can look at a variety of features in our network setup. Using the “-ie” option, we can examine the network interfaces in our system:

netstat 程序被用来检查各种各样的网络设置和统计数据。通过此命令的许多选项，我们 可以看看网络设置中的各种特性。使用“-ie”选项，我们能够查看系统中的网络接口：

[me@linuxbox ~]$ netstat -ie

eth0 Link encap:Ethernet HWaddr 00:1d:09:9b:99:67

inet addr:192.168.1.2 Bcast:192.168.1.255 Mask:255.255.255.0

inet6 addr: fe80::21d:9ff:fe9b:9967/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:238488 errors:0 dropped:0 overruns:0 frame:0

TX packets:403217 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:100 RX bytes:153098921 (146.0 MB) TX

bytes:261035246 (248.9 MB) Memory:fdfc0000-fdfe0000

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

...

In the example above, we see that our test system has two network interfaces. The first, called eth0, is the Ethernet interface and the second, called lo, is the loopback interface, a virtual interface that the system uses to “talk to itself.”

在上述实例中，我们看到我们的测试系统有两个网络接口。第一个，叫做 eth0，是 以太网接口，和第二个，叫做 lo，是内部回环网络接口，它是一个虚拟接口，系统用它来 “自言自语”。

When performing causal network diagnostics, the important things to look for are the presence of the word “UP” at the beginning of the fourth line for each interface, indicating that the network interface is enabled, and the presence of a valid IP address in the inet addr field on the second line. For systems using DHCP (Dynamic Host Configuration Protocol), a valid IP address in this field will verify that the DHCP is working.

当执行日常网络诊断时，要查看的重要信息是每个网络接口第四行开头出现的单词 “UP”，说明这个网络接口已经生效，还要查看第二行中 inet addr 字段出现的有效 IP 地址。对于使用 DHCP（动态主机配置协议）的系统，在 这个字段中的一个有效 IP 地址则证明了 DHCP 工作正常。

Using the “-r” option will display the kernel’s network routing table. This shows how the network is configured to send packets from network to network:

使用这个“-r”选项会显示内核的网络路由表。这展示了系统是如何配置网络之间发送数据包的。

[me@linuxbox ~]$ netstat -r

Kernel IP routing table

Destination Gateway Genmask Flags MSS Window irtt Iface

192.168.1.0 \* 255.255.255.0 U 0 0 0 eth0

default 192.168.1.1 0.0.0.0 UG 0 0 0 eth0

In this simple example, we see a typical routing table for a client machine on a LAN (Local Area Network) behind a firewall/router. The first line of the listing shows the destination 192.168.1.0. IP addresses that end in zero refer to networks rather than individual hosts, so this destination means any host on the LAN. The next field, Gateway, is the name or IP address of the gateway (router) used to go from the current host to the destination network. An asterisk in this field indicates that no gateway is needed.

在这个简单的例子里面，我们看到了，位于防火墙之内的局域网中，一台客户端计算机的典型路由表。 第一行显示了目的地 192.168.1.0。IP 地址以零结尾是指网络，而不是个人主机， 所以这个目的地意味着局域网中的任何一台主机。下一个字段，Gateway， 是网关（路由器）的名字或 IP 地址，用它来连接当前的主机和目的地的网络。 若这个字段显示一个星号，则表明不需要网关。

The last line contains the destination default. This means any traffic destined for a network that is not otherwise listed in the table. In our example, we see that the gateway is defined as a router with the address of 192.168.1.1, which presumably knows what to do with the destination traffic.

最后一行包含目的地 default。指的是发往任何表上没有列出的目的地网络的流量。 在我们的实例中，我们看到网关被定义为地址 192.168.1.1 的路由器，它应该能 知道怎样来处理目的地流量。

The netstat program has many options and we have only looked at a couple. Check out the netstat man page for a complete list.

netstat 程序有许多选项，我们仅仅讨论了几个。查看 netstat 命令的手册，可以 得到所有选项的完整列表。

**网络中传输文件**

What good is a network unless we know how to move files across it? There are many programs that move data over networks. We will cover two of them now and several more in later sections.

网络有什么用处呢？除非我们知道了怎样通过网络来传输文件。有许多程序可以用来在网络中 传送数据。我们先讨论两个命令，随后的章节里再介绍几个命令。

**ftp**

One of the true “classic” programs, ftp gets it name from the protocol it uses, the File Transfer Protocol. FTP is used widely on the Internet for file downloads. Most, if not all, web browsers support it and you often see URIs starting with the protocol ftp://. Before there were web browsers, there was the ftp program. ftp is used to communicate with FTP servers, machines that contain files that can be uploaded and downloaded over a network.

ftp 命令属于真正的“经典”程序之一，它的名字来源于其所使用的协议，就是文件传输协议。 FTP 被广泛地用来从因特网上下载文件。大多数，并不是所有的，网络浏览器都支持 FTP， 你经常可以看到它们的 URI 以协议 ftp://开头。在出现网络浏览器之前，ftp 程序已经存在了。 ftp 程序可用来与 FTP 服务器进行通信，FTP 服务器就是存储文件的计算机，这些文件能够通过 网络下载和上传。

FTP (in its original form) is not secure, because it sends account names and passwords in cleartext. This means that they are not encrypted and anyone sniffing the network can see them. Because of this, almost all FTP done over the Internet is done by anonymous FTP servers. An anonymous server allows anyone to login using the login name “anonymous” and a meaningless password.

FTP（它的原始形式）并不是安全的，因为它会以明码形式发送帐号的姓名和密码。这就意味着 这些数据没有加密，任何嗅探网络的人都能看到。由于此种原因，几乎因特网中所有 FTP 服务器 都是匿名的。一个匿名服务器能允许任何人使用注册名“anonymous”和无意义的密码登录系统。

In the example below, we show a typical session with the ftp program downloading an Ubuntu iso image located in the /pub/cd\_images/Ubuntu-8.04 directory of the anonymous FTP server fileserver:

在下面的例子中，我们将展示一个典型的会话，从匿名 FTP 服务器，其名字是 fileserver， 的/pub/\_images/Ubuntu-8.04的目录下，使用 ftp 程序下载一个 Ubuntu 系统映像文件。

[me@linuxbox ~]$ ftp fileserver

Connected to fileserver.localdomain.

220 (vsFTPd 2.0.1)

Name (fileserver:me): anonymous

331 Please specify the password.

Password:

230 Login successful.

Remote system type is UNIX.

Using binary mode to transfer files.

ftp> cd pub/cd\\_images/Ubuntu-8.04

250 Directory successfully changed.

ftp> ls

200 PORT command successful. Consider using PASV.

150 Here comes the directory listing.

-rw-rw-r-- 1 500 500 733079552 Apr 25 03:53 ubuntu-8.04- desktop-i386.iso

226 Directory send OK.

ftp> lcd Desktop

Local directory now /home/me/Desktop

ftp> get ubuntu-8.04-desktop-i386.iso

local: ubuntu-8.04-desktop-i386.iso remote: ubuntu-8.04-desktop-

i386.iso

200 PORT command successful. Consider using PASV.

150 Opening BINARY mode data connection for ubuntu-8.04-desktop-

i386.iso (733079552 bytes).

226 File send OK.

733079552 bytes received in 68.56 secs (10441.5 kB/s)

ftp> bye

Here is an explanation of the commands entered during this session:

这里是对会话期间所输入命令的解释说明：

|  |  |
| --- | --- |
| *Table 17-1:* | |
| Command | Meaning |
| ftp fileserver | Invoke the ftp program and have it connect to FTP server fileserver. |
| anonymous | Login name. After the login prompt, a password prompt will appear. Some servers will accept a blank password, others will require a password in the form of a email address. In that case, try something like “user@example.com”. |
| cd pub/cd\_images/Ubuntu-8.04 | Change to the directory on the remote system containing the desired file. Note that on most anonymous FTP servers, the files for public downloading are found somewhere under the pub directory. |
| ls | List the directory on the remote system. |
| lcd Desktop | Change the directory on the local system to ~/Desktop. In the example, the ftp program was invoked when the working directory was ~. This command changes the working directory to ~/Desktop. |
| get ubuntu-8.04-desktop- i386.iso | Tell the remote system to transfer the file ubuntu-8.04-desktop- i386.iso to the local system. Since the working directory on the local system was changed to ~/Desktop, the file will be downloaded there. |
| bye | Log off the remote server and end the ftp program session. The commands quit and exit may also be used. |

|  |  |
| --- | --- |
| *表17-1:* | |
| 命令 | 意思 |
| ftp fileserver | 唤醒 ftp 程序，让它连接到 FTP 服务器，fileserver。 |
| anonymous | 登录名。输入登录名后，将出现一个密码提示。一些服务器将会接受空密码， 其它一些则会要求一个邮件地址形式的密码。如果是这种情况，试着输入 “user@example.com”。 |
| cd pub/cd\_images/Ubuntu-8.04 | 跳转到远端系统中，要下载文件所在的目录下， 注意在大多数匿名的 FTP 服务器中，支持公共下载的文件都能在目录 pub 下找到 |
| ls | 列出远端系统中的目录。 |
| lcd Desktop | 跳转到本地系统中的 ~/Desktop 目录下。在实例中，ftp 程序在工作目录 ~ 下被唤醒。 这个命令把工作目录改为 ~/Desktop |
| get ubuntu-8.04-desktop-i386.iso | 告诉远端系统传送文件到本地。因为本地系统的工作目录 已经更改到了 ~/Desktop，所以文件会被下载到此目录。 |
| bye | 退出远端服务器，结束 ftp 程序会话。也可以使用命令 quit 和 exit。 |

Typing “help” at the “ftp>” prompt will display a list of the supported commands. Using ftp on a server where sufficient permissions have been granted, it is possible to perform many ordinary file management tasks. It’s clumsy, but it does work.

在 “ftp>” 提示符下，输入 “help”，会显示所支持命令的列表。使用 ftp 登录到一台 授予了用户足够权限的服务器中，则可以执行很多普通的文件管理任务。虽然很笨拙， 但它真能工作。

**lftp - 更好的 ftp**

ftp is not the only command line FTP client. In fact, there are many. One of better (and more popular) ones is lftp by Alexander Lukyanov. It works much like the traditional ftp program, but has many additional convenience features including multiple protocol support (including HTTP), automatic re-try on failed downloads, background processes, tab completion of path names, and many more.

ftp 并不是唯一的命令行形式的 FTP 客户端。实际上，还有很多。其中比较好（也更流行的）是 lftp 程序， 由 Alexander Lukyanov 编写完成。虽然 lftp 工作起来与传统的 ftp 程序很相似，但是它带有额外的便捷特性，包括 多协议支持（包括 HTTP），若下载失败会自动地重新下载，后台处理，用 tab 按键来补全路径名，还有很多。

**wget**

Another popular command line program for file downloading is wget. It is useful for downloading content from both web and FTP sites. Single files, multiple files, and even entire sites can be downloaded. To download the first page of linuxcommand.org we could do this:

另一个流行的用来下载文件的命令行程序是 wget。若想从网络和 FTP 网站两者上都能下载数据，wget 是很有用处的。 不只能下载单个文件，多个文件，甚至整个网站都能下载。下载 linuxcommand.org 网站的首页， 我们可以这样做：

[me@linuxbox ~]$ wget http://linuxcommand.org/index.php

--11:02:51-- http://linuxcommand.org/index.php

=> `index.php'

Resolving linuxcommand.org... 66.35.250.210

Connecting to linuxcommand.org|66.35.250.210|:80... connected.

HTTP request sent, awaiting response... 200 OK

Length: unspecified [text/html]

[ < => ] 3,120 --.--K/s

11:02:51 (161.75 MB/s) - 'index.php' saved [3120]

The program’s many options allow wget to recursively download, download files in the background (allowing you to log off but continue downloading), and complete the download of a partially downloaded file. These features are well documented in its better-than-average man page.

这个程序的许多选项允许 wget 递归地下载，在后台下载文件（你退出后仍在下载），能完成未下载 全的文件。这些特性在命令手册，better-than-average 一节中有详尽地说明。

**与远程主机安全通信**

For many years, Unix-like operating systems have had the ability to be administered remotely via a network. In the early days, before the general adoption of the Internet, there were a couple of popular programs used to log in to remote hosts. These were the rlogin and telnet programs. These programs, however, suffer from the same fatal flaw that the ftp program does; they transmit all their communications (including login names and passwords) in cleartext. This makes them wholly inappropriate for use in the Internet age.

通过网络来远程操控类 Unix 的操作系统已经有很多年了。早些年，在因特网普遍推广之前，有 一些受欢迎的程序被用来登录远程主机。它们是 rlogin 和 telnet 程序。然而这些程序，拥有和 ftp 程序 一样的致命缺点；它们以明码形式来传输所有的交流信息（包括登录命令和密码）。这使它们完全不 适合使用在因特网时代。

**ssh**

To address this problem, a new protocol called SSH (Secure Shell) was developed. SSH solves the two basic problems of secure communication with a remote host. First, it authenticates that the remote host is who it says it is (thus preventing so-called “man in the middle” attacks), and second, it encrypts all of the communications between the local and remote hosts.

为了解决这个问题，开发了一款新的协议，叫做 SSH（Secure Shell）。 SSH 解决了这两个基本的和远端主机安全交流的问题。首先，它要认证远端主机是否为它 所知道的那台主机（这样就阻止了所谓的“中间人”的攻击），其次，它加密了本地与远程主机之间 所有的通讯信息。

SSH consists of two parts. An SSH server runs on the remote host, listening for incoming connections on port twenty-two, while an SSH client is used on the local system to communicate with the remote server.

SSH 由两部分组成。SSH 服务器运行在远端主机上运行，在端口号22上监听将要到来的连接，而 SSH 客户端用在本地系统中，用来和远端服务器通信。

Most Linux distributions ship an implementation of SSH called OpenSSH from the BSD project. Some distributions include both the client and the server packages by default (for example, Red Hat), while others (such as Ubuntu) only supply the client. To enable a system to receive remote connections, it must have the OpenSSH-server package installed, configured and running, and (if the system is either running or is behind a firewall) it must allow incoming network connections on TCP port 22.

大多数 Linux 发行版自带一个提供 SSH 功能的软件包，叫做 OpenSSH，来自于 BSD 项目。一些发行版 默认包含客户端和服务器端两个软件包（例如，Red Hat）,而另一些（比方说 Ubuntu）则只是提供客户端服务。为了能让系统接受远端的连接，它必须 安装 OpenSSH-server 软件包，配置，运行它，并且（如果系统正在运行，或者是在防火墙之后） 它必须允许在 TCP 端口号上接收网络连接。

Tip: If you don’t have a remote system to connect to but want to try these examples, make sure the OpenSSH-server package is installed on your system and use localhost as the name of the remote host. That way, your machine will create network connections with itself.

小贴示：如果你没有远端系统去连接，但还想试试这些实例，则确认安装了 OpenSSH-server 软件包 ，则可使用 localhost 作为远端主机的名字。这种情况下，计算机会和它自己创建网络连接。

The SSH client program used to connect to remote SSH servers is called, appropriately enough, ssh. To connect to a remote host named remote-sys, we would use the ssh client program like so:

用来与远端 SSH 服务器相连接的 SSH 客户端程序，顺理成章，叫做 ssh。连接到远端名为 remote-sys 的主机，我们可以这样使用 ssh 客户端程序：

[me@linuxbox ~]$ ssh remote-sys

The authenticity of host 'remote-sys (192.168.1.4)' can't be

established.

RSA key fingerprint is

41:ed:7a:df:23:19:bf:3c:a5:17:bc:61:b3:7f:d9:bb.

Are you sure you want to continue connecting (yes/no)?

The first time the connection is attempted, a message is displayed indicating that the authenticity of the remote host cannot be established. This is because the client program has never seen this remote host before. To accept the credentials of the remote host, enter “yes” when prompted. Once the connection is established, the user is prompted for his/her password:

第一次尝试连接，提示信息表明远端主机的真实性不能确立。这是因为客户端程序以前从没有 看到过这个远端主机。为了接受远端主机的身份验证凭据，输入“yes”。一旦建立了连接，会提示 用户输入他或她的密码：

Warning: Permanently added 'remote-sys,192.168.1.4' (RSA) to the list

of known hosts.

me@remote-sys's password:

After the password is successfully entered, we receive the shell prompt from the remote system:

成功地输入密码之后，我们会接收到远端系统的 shell 提示符：

Last login: Sat Aug 30 13:00:48 2008

[me@remote-sys ~]$

The remote shell session continues until the user enters the exit command at the remote shell prompt, thereby closing the remote connection. At this point, the local shell session resumes and the local shell prompt reappears.

远端 shell 会话一直存在，直到用户输入 exit 命令后，则关闭了远程连接。这时候，本地的 shell 会话 恢复，本地 shell 提示符重新出现。

It is also possible to connect to remote systems using a different user name. For example, if the local user “me” had an account named “bob” on a remote system, user me could log in to the account bob on the remote system as follows:

也有可能使用不同的用户名连接到远程系统。例如，如果本地用户“me”，在远端系统中有一个帐号名 “bob”，则用户 me 能够用 bob 帐号登录到远端系统，如下所示：

[me@linuxbox ~]$ ssh bob@remote-sys

bob@remote-sys's password:

Last login: Sat Aug 30 13:03:21 2008

[bob@remote-sys ~]$

As stated before, ssh verifies the authenticity of the remote host. If the remote host does not successfully authenticate, the following message appears:

正如之前所讲到的，ssh 验证远端主机的真实性。如果远端主机不能成功地通过验证，则会提示以下信息：

[me@linuxbox ~]$ ssh remote-sys

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@

WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!

@

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!

Someone could be eavesdropping on you right now (man-in-the-middle

attack)!

...

This message is caused by one of two possible situations. First, an attacker may be attempting a “man-in-the-middle” attack. This is rare, since everybody knows that ssh alerts the user to this. The more likely culprit is that the remote system has been changed somehow; for example, its operating system or SSH server has been reinstalled. In the interests of security and safety however, the first possibility should not be dismissed out of hand. Always check with the administrator of the remote system when this message occurs.

有两种可能的情形会提示这些信息。第一，某个攻击者企图制造“中间人”袭击。这很少见， 因为每个人都知道 ssh 会针对这种状况发出警告。最有可能的罪魁祸首是远端系统已经改变了； 例如，它的操作系统或者是 SSH 服务器重新安装了。然而，为了安全起见，第一个可能性不应该 被轻易否定。当这条消息出现时，总要与远端系统的管理员查对一下。

After it has been determined that the message is due to a benign cause, it is safe to correct the problem on the client side. This is done by using a text editor (vim perhaps) to remove the obsolete key from the ~/.ssh/known\_hosts file. In the example message above, we see this:

当确定了这条消息归结为一个良性的原因之后，那么在客户端更正问题就很安全了。 使用文本编辑器（可能是 vim）从文件~/.ssh/known\_hosts 中删除废弃的钥匙， 就解决了问题。在上面的例子里，我们看到这样一句话：

Offending key in /home/me/.ssh/known\_hosts:1

This means that line one of the known\_hosts file contains the offending key. Delete this line from the file, and the ssh program will be able to accept new authentication credentials from the remote system.

这意味着文件 known\_hosts 里面某一行包含攻击型的钥匙。从文件中删除这一行，则 ssh 程序 就能够从远端系统接受新的身份验证凭据。

Besides opening a shell session on a remote system, ssh also allows us to execute a single command on a remote system. For example, to execute the free command on a remote host named remote-sys and have the results displayed on the local system:

除了能够在远端系统中打开一个 shell 会话，ssh 程序也允许我们在远端系统中执行单个命令。 例如，在名为 remote-sys 的远端主机上，执行 free 命令，并把输出结果显示到本地系统 shell 会话中。

[me@linuxbox ~]$ ssh remote-sys free

me@twin4's password:

total used free shared buffers cached

Mem: 775536 507184 268352 0 110068 154596

-/+ buffers/cache: 242520 533016

Swap: 0 1572856 0 110068 154596

[me@linuxbox ~]$

It’s possible to use this technique in more interesting ways, such as this example in which we perform an ls on the remote system and redirect the output to a file on the local system:

有可能以更有趣的方式来利用这项技术，比方说下面的例子，我们在远端系统中执行 ls 命令， 并把命令输出重定向到本地系统中的一个文件里面。

[me@linuxbox ~]$ ssh remote-sys 'ls \\*' > dirlist.txt

me@twin4's password:

[me@linuxbox ~]$

Notice the use of the single quotes in the command above. This is done because we do not want the pathname expansion performed on the local machine; rather, we want it to be performed on the remote system. Likewise, if we had wanted the output redirected to a file on the remote machine, we could have placed the redirection operator and the filename within the single quotes:

注意，上面的例子中使用了单引号。这样做是因为我们不想路径名展开操作在本地执行 ；而希望 它在远端系统中被执行。同样地，如果我们想要把输出结果重定向到远端主机的文件中，我们可以 把重定向操作符和文件名都放到单引号里面。

[me@linuxbox ~]$ ssh remote-sys 'ls \* > dirlist.txt'

Tunneling With SSH

**SSH 通道**

Part of what happens when you establish a connection with a remote host via SSH is that an encrypted tunnel is created between the local and remote systems. Normally, this tunnel is used to allow commands typed at the local system to be transmitted safely to the remote system, and for the results to be transmitted safely back. In addition to this basic function, the SSH protocol allows most types of network traffic to be sent through the encrypted tunnel, creating a sort of VPN (Virtual Private Network) between the local and remote systems.

当你通过 SSH 协议与远端主机建立连接的时候，其中发生的事就是在本地与远端系统之间 创建了一条加密通道。通常，这条通道被用来把在本地系统中输入的命令安全地传输到远端系统， 同样地，再把执行结果安全地发送回来。除了这个基本功能之外，SSH 协议允许大多数 网络流量类型通过这条加密通道来被传送，在本地与远端系统之间创建某种 VPN（虚拟专用网络）。

Perhaps the most common use of this feature is to allow X Window system traffic to be transmitted. On a system running an X server (that is, a machine displaying a GUI), it is possible to launch and run an X client program (a graphical application) on a remote system and have its display appear on the local system. It’s easy to do, here’s an example: let’s say we are sitting at a Linux system called linuxbox which is running an X server, and we want to run the xload program on a remote system named remote-sys and see the program’s graphical output on our local system. We could do this:

可能这个特性的最普遍使用是允许传递 X 窗口系统流量。在运行着 X 服务器（也就是， 能显示 GUI 的机器）的系统中，有可能在远端启动和运行一个 X 客户端程序（一个图形化应用程序）， 而应用程序的显示结果出现在本地。这很容易完成，这里有个例子：假设我们正坐在一台装有 Linux 系统， 叫做 linuxbox 的机器之前，且系统中运行着 X 服务器，现在我们想要在名为 remote-sys 的远端系统中 运行 xload 程序，但是要在我们的本地系统中看到这个程序的图形化输出。我们可以这样做：

[me@linuxbox ~]$ ssh -X remote-sys

me@remote-sys's password:

Last login: Mon Sep 08 13:23:11 2008

[me@remote-sys ~]$ xload

After the xload command is executed on the remote system, its window appears on the local system. On some systems, you may need to use the “-Y” option rather than the “-X” option to do this.

这个 xload 命令在远端执行之后，它的窗口就会出现在本地。在某些系统中，你可能需要 使用 “－Y” 选项，而不是 “－X” 选项来完成这个操作。

**scp 和 sftp**

The OpenSSH package also includes two programs that can make use of an SSH encrypted tunnel to copy files across the network. The first, scp (secure copy) is used much like the familiar cp program to copy files. The most notable difference is that the source or destination pathnames may be preceded with the name of a remote host, followed by a colon character. For example, if we wanted to copy a document named document.txt from our home directory on the remote system, remote-sys, to the current working directory on our local system, we could do this:

这个 OpenSSH 软件包也包含两个程序，它们可以利用 SSH 加密通道在网络间复制文件。 第一个，scp（安全复制）被用来复制文件，与熟悉的 cp 程序非常相似。最显著的区别就是 源或者目标路径名要以远端主机的名字，后跟一个冒号字符开头。例如，如果我们想要 从远端系统，remote-sys，的家目录下复制文档 document.txt，到我们本地系统的当前工作目录下， 可以这样操作：

[me@linuxbox ~]$ scp remote-sys:document.txt .

me@remote-sys's password:

document.txt

100% 5581 5.5KB/s 00:00

[me@linuxbox ~]$

As with ssh, you may apply a user name to the beginning of the remote host’s name if the desired remote host account name does not match that of the local system:

和 ssh 命令一样，如果你所期望的远端主机帐户与你本地系统中的不一致， 则可以把用户名添加到远端主机名的开头。

[me@linuxbox ~]$ scp bob@remote-sys:document.txt .

The second SSH file copying program is sftp which, as its name implies, is a secure replacement for the ftp program. sftp works much like the original ftp program that we used earlier; however, instead of transmitting everything in cleartext, it uses an SSH encrypted tunnel. sftp has an important advantage over conventional ftp in that it does not require an FTP server to be running on the remote host. It only requires the SSH server. This means that any remote machine that can connect with the SSH client can also be used as a FTP-like server. Here is a sample session:

第二个 SSH 文件复制命令是 sftp，正如其名字所示，它是 ftp 程序的安全替代品。sftp 工作起来与我们 之前使用的 ftp 程序很相似；然而，它不用明码形式来传递数据，它使用加密的 SSH 通道。sftp 有一个 重要特性强于传统的 ftp 命令，就是 sftp 不需要远端系统中运行 FTP 服务器。它仅仅要求 SSH 服务器。 这意味着任何一台能用 SSH 客户端连接的远端机器，也可当作类似于 FTP 的服务器来使用。 这里是一个样本会话：

[me@linuxbox ~]$ sftp remote-sys

Connecting to remote-sys...

me@remote-sys's password:

sftp> ls

ubuntu-8.04-desktop-i386.iso

sftp> lcd Desktop

sftp> get ubuntu-8.04-desktop-i386.iso

Fetching /home/me/ubuntu-8.04-desktop-i386.iso to ubuntu-8.04-

desktop-i386.iso

/home/me/ubuntu-8.04-desktop-i386.iso 100% 699MB 7.4MB/s 01:35

sftp> bye

Tip: The SFTP protocol is supported by many of the graphical file managers found in Linux distributions. Using either Nautilus (GNOME) or Konqueror (KDE), we can enter a URI beginning with sftp:// into the location bar and operate on files stored on a remote system running an SSH server.

小贴示：这个 SFTP 协议被许多 Linux 发行版中的图形化文件管理器支持。使用 Nautilus (GNOME), 或者是 Konqueror (KDE)，我们都能在位置栏中输入以 sftp:// 开头的 URI， 来操作存储在运行着 SSH 服务器的远端系统中的文件。

An SSH Client For Windows?

**Windows 中的 SSH 客户端**

Let’s say you are sitting at a Windows machine but you need to log in to your Linux server and get some real work done, what do you do? Get an SSH client program for your Windows box, of course! There are a number of these. The most popular one is probably PuTTY by Simon Tatham and his team. The PuTTY program displays a terminal window and allow a Windows user to open an SSH (or telnet) session on a remote host. The program also provides analogs for the scp and sftp programs.

比方说你正坐在一台 Windows 机器前面，但是你需要登录到你的 Linux 服务器中，去完成 一些实际的工作，那该怎么办呢？当然是得到一个 Windows 平台下的 SSH 客户端！有很多这样 的工具。最流行的可能就是由 Simon Tatham 和他的团队开发的 PuTTY 了。这个 PuTTY 程序 能够显示一个终端窗口，而且允许 Windows 用户在远端主机中打开一个 SSH（或者 telnet）会话。 这个程序也提供了 scp 和 sftp 程序的类似物。

PuTTY is available at <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

PuTTY 可在链接 <http://www.chiark.greenend.org.uk/~sgtatham/putty/> 处得到。